# Algorithm and Hardware Design for High Volume Rate 3-D Medical Ultrasound Imaging

by

Jian Zhou

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2019 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Chair
Antonia Papandreou-Suppappola
Umit Ogras
Thomas F. Wenisch

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Ultrasound B-mode imaging is an increasingly significant medical imaging modality for clinical applications. Compared to other imaging modalities like computed tomography (CT) or magnetic resonance imaging (MRI), ultrasound imaging has the advantage of being safe, inexpensive, and portable. While two dimensional (2-D) ultrasound imaging is very popular, three dimensional (3-D) ultrasound imaging provides distinct advantages over its 2-D counterpart by providing volumetric imaging, which leads to more accurate analysis of tumor and cysts. However, the amount of received data at the front-end of 3-D system is extremely large, making it impractical for power-constrained portable systems.

In this thesis, algorithm and hardware design techniques to support a hand-held 3-D ultrasound imaging system are proposed. Synthetic aperture sequential beamforming (SASB) is chosen since its computations can be split into two stages, where the output generated of Stage 1 is significantly smaller in size compared to the input. This characteristic enables Stage 1 to be done in the front end while Stage 2 can be sent out to be processed elsewhere.

The contributions of this thesis are as follows. First, 2-D SASB is extended to 3-D. Techniques to increase the volume rate of 3-D SASB through a new multi-line firing scheme and use of linear chirp as the excitation waveform, are presented. A new sparse array design that not only reduces the number of active transducers but also avoids the imaging degradation caused by grating lobes, is proposed. A combination of these techniques increases the volume rate of 3-D SASB by 4× without introducing extra computations at the front end.

Next, algorithmic techniques to further reduce the Stage 1 computations in the front end are presented. These include reducing the number of distinct apodization coefficients and operating with narrow-bit-width fixed-point data. A 3-D die stacked

architecture is designed for the front end. This highly parallel architecture enables the signals received by 961 active transducers to be digitalized, routed by a network-on-chip, and processed in parallel. The processed data are accumulated through a bus-based structure. This architecture is synthesized using TSMC *28 nm* technology node and the estimated power consumption of the front end is less than 2 W.

Finally, the Stage 2 computations are mapped onto a reconfigurable multi-core architecture, TRANSFORMER, which supports different types of on-chip memory banks and run-time reconfigurable connections between general processing elements and memory banks. The matched filtering step and the beamforming step in Stage 2 are mapped onto TRANSFORMER with different memory configurations. Gem5 simulations show that the private cache mode generates shorter execution time and higher computation efficiency compared to other cache modes. The overall execution time for Stage 2 is 14.73 $ms$. The average power consumption and the average Giga-operations-per-second/Watt in 14 $nm$ technology node are 0.14 $W$ and 103.84, respectively.

DEDICATION

*To the sunshine in Arizona*

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor Dr. Chaitali Chakrabarti for her devotion on teaching me how to research, write, and speak. It was her great patience and encouragement that gave me confidence to challenge myself during my PhD journey. I would not complete this work without her mindful guidance.

I'm extremely grateful to Dr. Antonia Papandreou-Suppappola who introduced me to the academic world and provided me with warm encouragement in both research and life.

I would like to thank my committee members: Dr. Thomas F. Wenisch and Dr. Umit Ogras. The insightful comments that they provided make this work more complete and comprehensive. I would also like to thank the University of Michigan team, Dr. Brian Fowlkes, Dr. Oliver Kripfgans, Dr. Ronald Dreslinski, Dr. Trevor Mudge, and Dr. Hun-seok Kim, for their insightful technical advice during the portable ultrasound imaging project and the software-defined hardware project.

I am also grateful to the National Science Foundation CCF-1406810, the DARPA Software-defined Hardware Project, and the School of Electrical, Computer and Energy Engineering at ASU for funding my graduate study.

I want to thank my colleagues in the two projects, Dr. Siyuan Wei, Dr. Ming Yang, Dr. Richard Sampson, Dr. Rungroj Jintamethasawat, Brendan West, Sumit Mondal, Subhankar Pal, Yan Xiong, Srinidhi Renganathan, and Anuraag Soorishetty, for their support and help on my PhD research work. It was our smooth cooperation that produces all the rich outcomes.

I also want to thank all my labmates in the low power system lab, Dr. Hsing-min Chen, Dr. Manqing Mao, Shunyao Wu, Jiang Xiang, Jingtao Li, Srimayee Kanagala,

Xing Chen, and Liangliang Chang, for bringing joy and happiness to my dull daily life.

Finally, I am grateful to my parents. I can not accompany them these years but I'm always proud of them for the great things they are working on.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

# Chapter 1

# INTRODUCTION

Ultrasound imaging is an increasingly important medical imaging modality for clinical applications. It is non-invasive and inexpensive compared to computing tomography (CT) and magnetic resonance imaging (MRI). It is also more appropriate for portable applications, since transmitting ultrasound modulated wave and receiving echoed signals require little power [2, 3].

Modern ultrasound systems are able to make detailed observation of blood movements in vessels and tissues, monitor subtle changes in tissue texture, and detect even very small cysts [4]. Although most B-mode imaging systems in clinics nowadays are two dimensional (2-D) [5,6], three dimensional (3-D) ultrasound imaging provides distinct advantages of providing volumetric imaging of cysts and tumors and accurate measurements of volumetric flow.

Clinical studies have shown that the portability of ultrasound imaging devices helps early diagnosis. However, owing to the constraints in the computational ability and power budget, today's portable systems generate 2-D images with low resolution and also low frame rate. Designing a portable system for 3-D system is significantly more challenging. For example, a 3-D ultrasound imaging system which employs 90×90 transducer elements to generate a image with 30×30×5195 voxels needs 2700× more computations compared to its 2-D counterpart which uses 90 transducer elements and generates a image of size 30×5195.

The most computationally complex unit in an ultrasound system is the beamforming unit. Beamforming is a spatial filtering process where the locations and the amplitude of the scatterers are reconstructed from the wavefronts received

by the transducer elements. This process can be done in either time domain or frequency domain. Typical beamforming algorithms used in ultrasound are delay-and-sum beamforming and $f$-k domain beamforming.

In this work, we present low-cost beamforming algorithms suitable for portable 3-D imaging. We focus on a beamforming algorithm that generates imaging volumes with low complexity and range-independent resolutions. This algorithm is referred to as synthetic aperture sequential beamforming in the literature [7, 8]. It is very hardware-friendly since it drastically reduces the front-end data and allows for the processing to be split into two stages. We design a low-power architecture for the first stage of this algorithm and plan to implement the second stage on a prototype multi-core architecture.

## 1.1 Problem Definition

In 3-D imaging, the amount of receive data that has to be processed by the front-end is extremely large, making it impractical to implement high volume-rate imaging in power-constrained portable systems. One way to reduce the amount of front-end data that must be stored or transmitted is to perform beamforming in the transducer probe [9]. However, the computational requirement of beamforming is very large. Separable beamforming [10, 11] has been shown to reduce the computational requirement by 19×.

Designing a low-power architecture to support high volume rate imaging is still challenging. Recently, researchers have proposed use of compressive sensing to sample the received signal with a sub-Nyquist sampling rate. This approach requires transforming the radio frequency (RF) signal to a domain where the signal can be sparsely represented [12–14], or implementing beamforming in the frequency domain and recovering the beamformed signal using convex optimization [15]. These methods

reduce the size of the receive data but have higher computational complexity at the front-end.

Synthetic aperture sequential beamforming (SASB) is a promising beamforming technique [7,8] that drastically reduces the front-end data size while achieving range-independent resolution. SASB divides the beamforming into two stages: fixed transmit and receive beamforming in the first stage, followed by dynamic transmit and receive beamforming in the second stage. Since front-end receive data is drastically compressed in the first stage, the intermediate data can be easily transferred to another computational unit for computations in the second stage. Recently, researchers have proposed several extensions of 2-D SASB in [7, 8, 16, 17]. These methods focus on increasing the imaging quality and reducing the computational complexity in the second stage.

The first beamforming stage of SASB still has high computational complexity. Since this is supposed to be computed in the transducer head, the high complexity will result in large power consumption and limit the volume rate in power constrained devices. Simply reducing the number of receive elements results in grating lobes and degrades the imaging quality. Furthermore, as the number of scanlines is the same as the number of firing events, the volume rate of SASB is low, which is not desirable.

In this thesis, the goal is to (i) develop a high volume rate 3-D SASB algorithm that is hardware-friendly, and (ii) demonstrate its superior algorithmic and hardware performance.

## 1.2 Contributions

### *1.2.1 High Volume Rate 3-D SASB*

In this work, we first present a high volume rate 3-D SASB algorithm that is suitable for kidney and obstetric B-mode imaging. We first extend 2-D SASB to 3-D SASB. To reduce computational requirements of 3-D SASB, we propose reducing the number of active receive elements by increasing the element spacing from $\lambda$ to $2\lambda$, where $\lambda$ denotes wavelength. Increased spacing reduces the number of computations in the front-end by $4\times$ without introducing severe degradation in image quality. Details of this work are presented in [18].

To increase the volume rate, we propose to employ a multiple-transmit multiple-receive (MTMR) firing scheme and use linear chirps instead of sinusoids as the excitation waveform. Linear chirp excitation reduces the interference between simultaneously firing subapertures, especially in deep regions. To support four simultaneous transmit and receive operations, we design four chirps that operate over two overlapped frequency bands, with two chirps in each frequency band having opposite chirp rates. We show that a 25% - 50% frequency overlap results in good image quality. With four subapertures firing simultaneously, the volume rate of 3-D SASB is increased by $4\times$ without increasing front-end computations. However, the MTMR firing scheme results in grating lobes in the shallow region if the element spacing between active receive elements is $2\lambda$. To mitigate the grating lobes, we design a sparse array based on a bin-based random array. We optimize the active receive element locations to reduce sidelobe levels.

We present cyst images generated by Field II [19, 20] simulations to demonstrate the effectiveness of our imaging method in both shallow and deep regions. Simulation results show that the proposed method incurs small degradation in image quality

compared to 3-D SASB using a single transmit and single receive firing scheme. Details of MTMR work are presented in [21, 22].

### *1.2.2 Accelerator Design for Stage 1 of 3-D SASB*

We further investigate techniques to reduce the number of computations in Stage 1 of 3-D SASB. Since multiple receive data are scaled by the same apodization coefficient, we propose a Sum-before-Multiply scheme that first sums up the relevant data, and then multiplies it with the apodization coefficient. We also reduce the number of distinct apodization coefficients by clustering the coefficients that have close values, so that the number of multiplications are further reduced.

We present imaging quality results generated by Field-II using different number of coefficients to demonstrate that 16 distinct coefficient values is sufficient for good imaging quality. We also investigate narrow-bit-width fixed-point computations in both data path and ADC. Simulation results show that the lowest bit-widths that achieve good imaging quality are 12 bits for the data path and 8 bits for ADC.

To support the proposed Sum-before-Multiply scheme, we propose a highly parallelized 3-D die stacked architecture. In this architecture, signals received by 961 active receive elements are digitized by 961 ADCs and routed to 961 digital processing channels through a network-on-chip (NoC) [23–26]. The routed digital signals are delayed and interpolated by 961 processing channels in parallel. The interpolated samples are summed up through a bus-based structure that traverses through all 961 processing channels. We synthesize the proposed digital processing channel using Taiwan Semiconductor Manufacturing Company (TSMC)'s *28 nm* technology node and find that the power consumption of the 961 channels is ~1 W and the area is 2.66 $mm^2$. We also estimate the area and power consumption of NoC through *BookSim* [27]. The simulation results show that with 2 virtual channels, the maximum latency is

308 $ns$, resulting in no change in the volume rate. The power of this NoC is 66 $mW$ and the area is 0.51 $mm^2$.

### *1.2.3 Mapping Stage 2 of 3-D SASB onto a Multi-core Architecture*

In 3-D SASB, the data generated by the front-end architecture can be transferred to a separate computing unit for Stage 2 computations. In Stage 2, the data are first passed through a matched filter. Then, a dynamic focus beamforming is performed over the filtered signals to generate each imaging voxel.

We map Stage 2 computations onto a reconfigurable multi-core architecture, TRANSFORMER [1], that has been designed at the University of Michigan. TRANSFORMER consists of a number of general processing elements connected to a two-level memory hierarchy through a crossbar. The on-chip memory can be configured as either scratchpad or cache, based on the requirements of the algorithm. Furthermore, each memory can operate in the private mode, shared mode, and hybrid mode. In hybrid mode, half of the in-tile memory is configured as scratchpad mode and the other half is configured as shared cache mode. We divide Stage 2 into the convolution step and the beamforming step. We implement the convolution step using both shared cache mode and hybrid mode. We compare their performance with respect to execution time, computation efficiency (GOPS/Watt) and computing element utilization for different memory modes. We also describe implementing the convolution using FFT in systolic array mode in this architecture. Gem5 simulation results show that the best configuration for convolution step is 4 tiles with 16 GPEs in each tile. The execution time is 1.7 $ms$ and the power consumption is 0.29 W.

We implement dynamic beamforming using the shared cache mode. In the shared cache mode, the GPE directly accesses the constants, the data samples, and the apodization coefficients from the main memory. The scanlines are distributed to all

GPEs. Each GPE first decides whether the current input contributes to an output in its scanline. If so, the GPE multiplies the input sample with the apodization coefficient, and then adds the product to the relevant partial sum on the scanline. The partial sums are updated by data generated from subsequent firing events to generate the final output in a scanline. Gem5 simulation results show that the best configuration for beamforming step is 2 tiles with 16 GPEs in each tile. The execution time is 13.03 $ms$ and the power consumption is 0.12 W. The execution time of the overall system is 14.73 $ms$, the average power consumption is 0.14 $W$, and the average Giga-operations-per-second/Watt (GOPS/W) is 103.84.

## 1.3 Dissertation Organization

This report is organized as follows.

In Chapter 2, the time-domain and frequency-domain beamforming techniques are introduced along with different firing schemes. Also, existing hardware architectures corresponding to the different firing schemes are presented.

Chapter 3 describes a high volume rate 3-D ultrasound imaging technique. It is based on a new multi-line based firing scheme that uses chirp-based excitation. It also includes an optimization based sparse array design to remove the imaging artifacts without increasing the computational complexity. Field-II simulations are presented to verify the effectiveness of the proposed methods.

In Chapter 4, front-end architecture for implementing Stage 1 of SASB is presented. A technique for reducing the number of apodization coefficients is discussed along with imaging quality evaluation using Field-II simulations. Synthesis results generated using TSMC 28 nm technology node are also presented.

In Chapter 5, mapping Stage 2 computation onto a reconfigurable parallel architecture, TRANSFORMER, is presented. Simulations of the implementation are

presented along with performance comparisons with respect to execution time and computation efficiency for different mapping schemes.

Chapter 6 concludes the dissertation.

Chapter 2

# B-MODE IMAGING TECHNIQUES

## 2.1 System Overview of B-mode Imaging

In a B-mode imaging system, an array of transducer elements transmits high frequency waveforms into the region of interest and receives the waves echoed by the scatterers. The transducer elements are typically implemented by capacitive micromachined ultrasonic transducers (CMUTs). In the transmit mode, the waveforms are converted to analog form by D/A converter, amplified, and fed to the transducer elements. In the receive mode, the received signals undergo time gain compensation to reduce the dynamic range and are then digitalized by analog-to-digital converters (ADC). The digitized signals are beamformed to form each

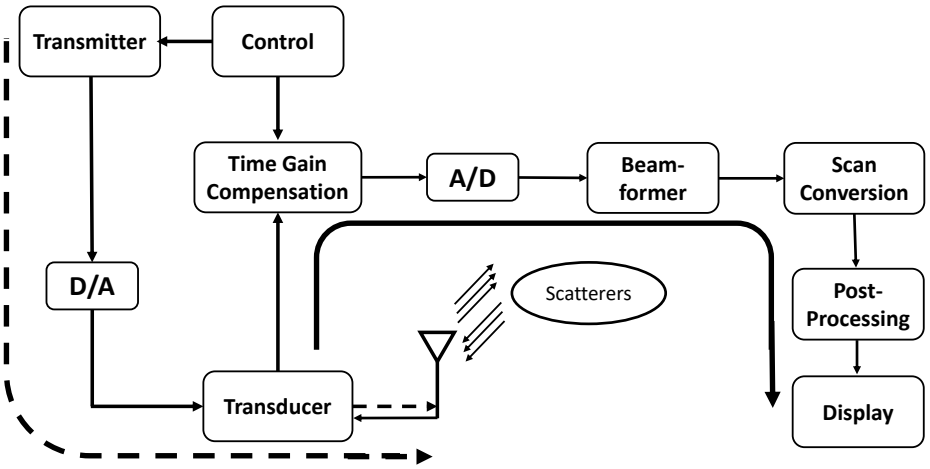



Figure 2.1: Main Components in the B-mode Imaging Systems.

scanline. After scan conversion and post-processing techniques, the B-mode image is reconstructed. Figure 2.1 shows the main components in a B-mode imaging system. The solid line shows the data path for received signals while the dashed line shows the data path for transmitted signals.

Beamforming is the algorithm that processes the received signals to reconstruct the final image. This can be done in either time domain or frequency domain. In time domain, the received signals are delayed in time and then summed up to form the imaging pixel. In frequency domain processing, the received signals (in time domain) are first transformed to frequency domain followed by application of appropriate phase shifts. The phase shifted frequency signals are summed up and then transformed back to the time domain to form the final image. Time-domain beamforming has high requirement on the ADC accuracy while frequency-domain beamforming has higher computational complexity.

Most medical ultrasound systems adopt delay-and-sum (DAS) beamforming, which is suited for reconstructing near-region scatterers. It has low computational complexity compared to other algorithms, such as frequency-domain beamforming or minimum variance distortionless response (MVDR) beamforming. In DAS beamforming, the digital signals are delayed by predetermined amounts, multiplied with apodization coefficients, and then summed up to reconstruct the image pixels. DAS has been applied to different ultrasound imaging firing schemes, such as synthetic aperture ultrasound (SAU) imaging, plane-wave imaging, phased array imaging, and synthetic aperture sequential beamforming (SASB).

In this chapter, we first introduce three imaging techniques that are widely used in ultrasound B-mode imaging applications, namely, plane-wave imaging, SAU, and SASB. Next, we briefly describe existing time-domain and frequency-domain

beamforming algorithms. Finally, we present a brief overview of hardware architecture for beamforming.

## 2.2 Delay-and-sum Beamforming

Delay-and-sum (DAS) beamforming is a typical time-domain beamforming algorithm that is widely used in B-mode ultrasound imaging. Compared to frequency-domain beamforming, it is easy to implement and it has low computational complexity. The digitized signals are delayed appropriately, where the delay value depends on the imaging technique. In fixed-focus beamforming, such as in the first stage of SASB, the delay value depends on the distance between the transducer element and the focusing point. In dynamic-focus beamforming, such as in SAU, plane-wave imaging, and the second stage of SASB, the delay depends on the distance between the transducer element and the imaging pixel.

The delay operation can be implemented with analog or digital circuits. In digital implementation, the received signals have to be oversampled to achieve high accuracy. Typically, the sampling rate required by DAS beamforming is 4-10 times of Nyquist sampling rate. This imposes high requirement on ADC sampling frequency. The digitized signals are usually interpolated before the delay operations, which helps increase the accuracy of DAS beamforming. Next, we introduce several firing schemes used in B-mode imaging.

### *2.2.1 Conventional B-mode Imaging Firing Scheme*

In conventional B-mode imaging, the linear array transmits a pulse focused on a specific point in the scanline using all the transducer elements. The focusing of the transmitted wave is achieved by applying appropriate delay values to the transducer elements, where the delay values depend on the locations of the transducer elements

and the location of the focusing point. This process is called fixed focusing. After the pulse is transmitted, all transducer elements start to receive the signals echoed by different scatterers in the field of view. The received signals are processed to reconstruct the imaging pixels along the scanline through beamforming. This process is repeated for all the scanlines yo image the whole region,.

However, using a single focusing point in each scanline results in low imaging quality in the regions away from the focusing point. To increase the imaging quality, the scanline is divided into different focal zones and a central point in each focal zone is chosen as the focusing point. The firing process is repeated for different focusing points along a scanline.

The frame rate in conventional B-mode imaging is a function of the number of scanlines and the number of focal zones in each scanline. Since the number of scanlines is usually large to achieve high lateral resolutions, the frame rate of conventional B-mode imaging is low [28].

### *2.2.2 Plane-wave Ultrasound Imaging*

In plane-wave ultrasound imaging, all the transducer elements in the linear array transmit waveforms simultaneously, forming a plane wave. Once a complete waveform is transmitted, all the transducer elements start to receive the signals. After the signal echoed by the furthest scatterer in the imaging region is received, the transducer elements transmit again (corresponding to the next firing event). Compared to conventional ultrasound imaging, plane-wave imaging generates one image frame using only one transmit and receive event, thus achieving a high frame rate. The high frame rate makes this scheme attractive for flow estimation applications.

The image is beamformed using dynamic receive beamforming, where the received signals are delayed depending on the distance between the image pixel and the

receive transducer element, and then summed up to reconstruct each image pixel. For different scanlines, since the relative distance between the pixels and the receive elements around the scanline are the same, one set of delay values can be shared by all scanlines.

Unfortunately, plane-wave imaging achieves low imaging quality compared to other imaging modalities. The imaging quality can be improved through compounding, where the plane-wave is transmitted and received multiple times, once for each angle. The plane wave is transmitted with a specific angle by delaying the time that each transducer element starts to transmit. The final image is generated by summing up the images generated in all firing events. As information acquired from more firing events are used to generate the final image, the quality is improved. This improvement comes at the cost of lower frame rate since more firing events are used to generate one frame.

### *2.2.3 Synthetic Aperture Ultrasound Imaging*

Synthetic aperture ultrasound (SAU) imaging is an ultrasound beamforming modality that achieves high imaging quality. Compared to conventional ultrasound imaging firing scheme and plane-wave imaging, SAU achieves higher lateral resolution. The use of dynamic focusing in both transmit and receive ends makes the frame rate higher than conventional ultrasound imaging firing scheme [29].

In 2-D SAU imaging, a linear array of transducers is used to transmit and receive the ultrasound signals. Different from the conventional ultrasound imaging firing scheme where all the transducer elements focus in each focal area one by one, SAU imaging transmits with only one transducer element and receives with all transducer elements. In each firing event, one transducer element transmits the waveform, and all the transducer elements receive the echoed signals. After the signals echoed from

scatterers at the maximum imaging depth is received, the next transducer element transmits. The signals received by each transducer element are delayed appropriately and summed up to form each image pixel. In this way, a low-resolution image is generated after each firing event. The final high-resolution is generated by summing up all the low-resolution images [28]. Since SAU transmits with one transducer element, the signal amplitude is limited, which results in low imaging quality in the deeper region. Extensions of SAU have been proposed that transmit divergent waves using a group of transducer elements which acts like a virtual source. Compared to when one transducer element is used, the virtual source based method uses more elements to fire, thus it generates higher SNR, which enables deeper penetration [30].

Dynamic focusing used in SAU generates higher lateral resolution, but also has higher computational complexity. Each image pixel is generated by delaying and summing up the signals received by all transducer elements. Since the delay value depends on both the distance between the imaging focal point and the receive transducer element as well as the distance between the imaging focal point and the transmit transducer element or the virtual source, it requires large number of delay values to generate one high-resolution image. This requirement is challenging for real-time implementations. Current methods to avoid the large storage requirement for the delay values include calculating the values on-the-fly and replacing the values with piece-wise approximation [31].

### *2.2.4 Synthetic Aperture Sequential Beamforming Ultrasound Imaging*

SASB is an advanced beamforming technique proposed as an extension of SAU [29]. Compared to SAU, SASB achieves range-independent resolution and also requires fewer computations. It is a two-stage beamforming process where after the first stage the volume of data gets reduced significantly. This allows for the first stage

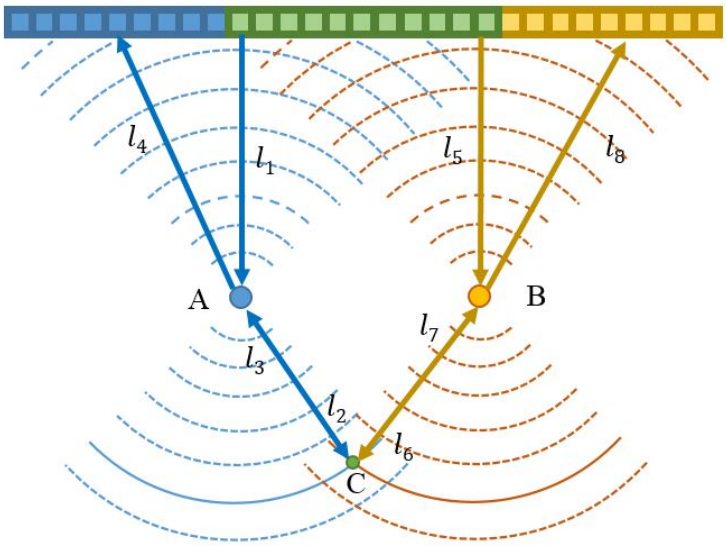



Figure 2.2: SASB Example of Delay Calculation Paths for Two Subapertures.

to be computed by the front end and front-end data to be transferred to a separate computing unit for processing the second stage.

SASB beamforming consists of a fixed transmit and receive beamforming that results in a single array of data, and a dynamic receive beamforming [7]. In the first stage, transducer elements within one subaperture are focused to a fixed point, referred to as the virtual element (VE). The process is repeated for each subaperture one by one, covering the entire imaging area of interest [8].

In the second stage, a general dynamic receive beamformer processes the first stage outputs to form a high-resolution image. Since the first stage is based on fixed focusing, the second stage can be regarded as the process that sums up spherical wave fronts emitted by the different VEs. Figure 2.2 shows how the imaging focal point C is obtained by summing the wavefronts corresponding to two VEs A and B. Here $l_1$ and $l_4$ represent the transmit and receive paths in the first beamforming stage, and $l_2$ and $l_3$ represent the paths for dynamic transmit and receive in the second beamforming

stage. The total beam path $l$ is given by:

$$l = l_1 + l_4 \pm (l_2 + l_3) \tag{2.1}$$

where the choice of ± depends on the location of the image pixel C relative to the VE. For instance, if the image pixel is nearer to the transducer than the VE, the path length $(l_2 + l_3)$ is subtracted. For a fixed-size subaperture, if the VE location is in the shallow region, the VE contributes to many more imaging points, thereby increasing the lateral resolution and also the complexity. Along the axial direction, the number of wavefronts contributing to an imaging point increases with depth, which helps maintain good lateral resolution, allowing SASB to achieve range-independent resolution [7]. However, as the number of transmit and receive events is same as the number of scanlines, SASB has low frame rate compared to both SAU and plane-wave imaging.

Recently, researchers have proposed several extensions of 2-D SASB in [7, 8]. F-k domain beamforming has been proposed to replace dynamic receive beamforming in the second stage. While this method increases processing speed in the second stage without losing axial or lateral resolution [16], it does not reduce the first-stage computation requirement. Another method [17] replaces the second beamforming stage with a spatial matched filter. This method helps remove the grating lobe caused by the low f-number aperture, thus resulting in higher contrast-to-noise ratio.

## 2.3 Frequency-domain Beamforming

Frequency domain beamforming is widely used in sonar and radar applications, where the center frequency of the waveform is much larger than in ultrasound. In these applications, the distance between the target and the antenna (or sensors) is much larger than the size of sensor or antenna array. Consequently, the incoming waves

that arrive at different sensors are regarded as parallel. The beamforming operation then only has to steer the received signals at a specific angle. Thus frequency domain beamforming involves phase shifting the received signal, where the phase shift is a function of the beamforming angle and the location of the transducer element. This process achieves higher accuracy compared to DAS beamforming and does not require oversampling.

However, in ultrasound beamforming, the region-of-interest is close to the source and thus the scattered signals are propagated as wavefronts instead of plane waves. So, dynamic focusing is used instead of steering based beamforming. In dynamic focusing, the delay applied on the received signals vary with the locations of the imaging pixel, thus it is a time-varying process. For a time-varying process, delaying with time dependent terms (in time domain) is equivalent to convolving the received signal with the phase shift function in the frequency domain. Such a process has high computational complexity, making frequency-domain beamforming unsuitable for hand-held devices.

Researchers recently proposed a frequency-domain beamforming for phased array imaging [32]. In this method, the phase shift is approximated by setting most coefficients to zero and only keeping a few with large amplitudes. In this way, the number of computations in the convolution operation is largely reduced. The required sampling rate is also dropped to Nyquist rate. To further reduce the sampling rate, the authors also combine this algorithm with compressive sensing where the image is reconstructed from the sub-Nyquist sampled signals using an $l_1$-norm based optimization. This method achieves imaging quality comparable to oversampled DAS algorithm while requiring much lower ADC rate.

### 2.3.1 $f$-$k$ Domain Beamforming

Frequency-wavenumber domain beamforming, or$f$-$k$ domain beamforming, is another Fourier-based beamforming algorithm proposed for plane-wave imaging [33, 34]. As the plane wave reaches a scatterer, the scatterer becomes a secondary source that emits spherical waves towards the transducer elements and generates diffraction hyperbolas in the back-scattered signals. In $f$-$k$ domain beamforming, the wavefronts of the back-scattered signals are assumed to be transmitted by the scatterers. The algorithm is based on seismic migration, where the wavefronts are used to recover the location and the amplitudes of the scatterers. The B-mode image is reconstructed from the locations and the amplitudes of the scatterers.

In $f$-$k$ domain beamforming, a 2-D Fourier Transform is performed on the signals received by all transducer elements. Then Stolt's migration is applied to transform the frequency-domain signals to wavenumber domain through non-linear interpolation. Finally, 2-D inverse Fourier Transform is performed to transform the wavenumber-domain signals back to the spatial domain, which is the reconstructed B-mode image [33].

Existing works also combine $f$-$k$ domain beamforming with the second beamforming stage of SASB [35]. In the second stage of SASB, the signals are regarded as transmitted and received by all VEs, similar to plane-wave imaging. In this implementation, the images are divided into two segments: one segment is between the VEs and the transducers elements, and the other is away from VEs. Each segment is reconstructed through the $f$-$k$ domain beamforming independently. FFT accelerators are used to reduce the computation time of the second beamforming stage.

## 2.4 Existing Hardware Architectures for 3-D Ultrasound Imaging

In this work, we propose a 3-D ultrasound imaging system based on SASB. Compared to 2-D imaging, 3-D imaging has much larger input data volume and requires significantly higher number of computations. To support large number of active receive elements, a large number of ADC units have to reside in the transducer head. Furthermore, as the existing cable bandwidth is not enough to transfer data to a separate computing unit, data has to be processed within the transducer head. Thus software-only beamforming techniques cannot be applied for 3-D imaging.

One solution to reduce the data volume size is to use analog-digital hybrid beamforming. Before digitalization, the signals received by a group of transducers are applied with fixed delays and then summed to generate only one channel in the analog domain [36–39]. This method reduces the number of required digital processing channels. But this comes at the cost of inaccurate beamforming computation which leads to degradation in imaging quality.

Another solution is to process beamforming within the transducer head using a hardware accelerator. The low-power 3-D beamforming accelerator, *Sonic Millip3De*, generates high-quality 3-D imaging volume for SAU systems. In each firing, the number of active receive elements is 1024, and the received data are processed by 1024 processing channels in parallel [9]. However, the 3-D die stack architecture and the large size of LPDDR2 make this architecture impractical. Other architectures for 3-D imaging include a table-free beamforming technique proposed in [40], which bypasses the need for storing large number of constants. Another 3-D beamformer, *Ekho*, does not require any external storage [41]. This is achieved by processing the incoming data at the sampling rate using bandpass processing and applying an approximate delay computation. By using a phased array, the system explores the region of interest

using 10K active transducer elements. Even in 28*nm* technology node, the power dissipation is 30.3 W, which is large for a portable device. Another FPGA-based architecture for portable 3-D imaging was proposed to support plane-wave imaging and spatial compounding [42]. This architecture requires a high-bandwidth cable (20 Gbps) to transfer the data from the transducer head to the FPGA board.

## 2.5 Summary

In this chapter, we introduced several beamforming algorithms and firing schemes to reconstruct ultrasound images. Firing scheme used for conventional B-mode imaging generates low imaging quality and also has low volume rate. Compared to the conventional scheme, plane-wave imaging generates images with a small number of firing events. It has a very high frame rate but low imaging quality. SAU generates high imaging quality but the large number of computations and large size of intermediate data make it impractical for hand-held implementations. SASB is another firing scheme that is able to largely compress the intermediate data but has low frame rate.

Of the beamforming algorithms, DAS beamforming has low computational complexity and is suitable for dynamic receive focusing. However, DAS beamforming requires high ADC rate. To bypass high ADC rate requirement, frequency domain beamforming is proposed. When combined with the compressive sensing techniques, the ADC rate requirements can be reduced to sub-Nyquist rate. $f$-$k$ domain beamforming is another beamforming technique, where seismic migration is used to recover the location and amplitude of the scatterers by regarding them as secondary sources. With the help of FFT accelerators, $f$-$k$ domain beamforming is able to largely reduce the computation time of plane-wave imaging and the second stage of SASB.

We also presented existing architectures for 3-D imaging. Compared to 2-D imaging, 3-D imaging has large data volume at the front end which makes software only beamforming impractical. One solution is to embed a mixed-signal beamformer at the front end, so that the data volumes are reduced. Another solution is to incorporate a hardware accelerator within the transducer head. Existing architectures focus on reducing the data to be processed in the front end and reducing the size of on-chip memory to store delay values.

# Chapter 3

# HIGH VOLUME RATE 3-D SYNTHETIC APERTURE SEQUENTIAL BEAMFORMING

## 3.1 Motivation

SASB divides the beamforming into two stages, where the received signals are beamformed using a fixed receive focusing in the first stage followed by a dynamic receive focusing in the second stage. The significant advantage of SASB is that the fixed receive focusing in the first stage compresses the received data drastically, making it possible to transfer the data to a separate computational unit, such as a CPU or GPU, for further processing. Such a scheme benefits 3-D ultrasound imaging even more, since the data volume at the front end is very large and the cost of transferring the data out of the chip can be significant.

Since 3-D imaging generates a volumetric view which helps improve the diagnosis accuracy, we propose a 3-D extension of SASB, where a 2-D transducer array is used instead of a 1-D linear array. The subaperture is of size $S \times S$, where $S$ can be 16, 32, or 64. Although the intermediate data size is the same as 2-D SASB, the increase in the number of active receive elements results in large number of computations at the front end. So in this work, we propose schemes to reduce the computational complexity at the front end.

The frame rate of 2-D SASB is low since the subapertures fire and receive one by one and the number of subapertures is same as the number of scanlines. In 3-D SASB, shift of one results in very low volume rate (which is the equivalence of "frame rate" in 3-D). For instance, for a transducer array of 90×90 elements and subaperture of

32×32 elements, the number of subapertures is 59×59 for shift of one. If the imaging depth is 100 mm and ultrasound velocity is 1540m/s, the round-trip time is 130 $\mu s$, resulting in a frame rate of 2.2 frames per second, which is too small! If the shift is larger, the volume rate is higher but the number of subapertures (which is equivalent to the number of scanlines) is smaller, resulting in degradation in imaging quality.

In the rest of this chapter, we first describe the 3-D extension of SASB and present two methods that reduce its computational complexity. Then, we describe the multiple transmit and multiple receive (MTMR) firing scheme that increases the volume rate by 4×. To reduce the interference between different firings, we propose to use linear chirps as the excitation waveform. To remove the grating lobes caused by MTMR firing scheme, we propose a non-uniformly distributed sparse array as the active receive subaperture without increasing the computational complexity.

## 3.2 3-D Synthetic Aperture Sequential Beamforming

Consider a 3-D SASB system, with $S_x \times S_y$ receive elements per subaperture. Let the VE be located at depth $d_p$ below the center of the subaperture. The first stage beamforming corresponding to subaperture $(n_x, n_y)$ is as follows:

$$F_1(n_x, n_y; t) = \sum_{s_x=1}^{S_x} \sum_{s_y=1}^{S_y} a_1(s_x, s_y) \cdot r(n_x, n_y, s_x, s_y; t - \tau_1(s_x, s_y)) \tag{3.1}$$

where $(s_x, s_y)$ is the index of the transducer element within the subaperture $(n_x, n_y)$, $a_1(s_x, s_y)$ represents the 2-D apodization coefficient for $(s_x, s_y)$, $r(n_x, n_y, s_x, s_y; t)$ represents the data received by $(s_x, s_y)$, and $\tau_1(s_x, s_y)$ represents the delay for $(s_x, s_y)$. The values of $\tau_1(s_x, s_y)$ are fixed for a fixed VE location. After the first stage beamforming, the data received in $S_x \times S_y$ channels are compressed into one scanline of length $M_z$. The size of storage required for intermediate data reduces significantly and it is now possible to ship the data to a processing unit located elsewhere.

In the first stage, all beams are steered to the VE. Delay for fixed focusing transmit and receive is given by:

$$\tau_1(s_x, s_y) = \frac{1}{c}(d_p + \sqrt{d_p^2 + (x_s - x_0)^2 + (y_s - y_0)^2}) \tag{3.2}$$

where $(x_s, y_s)$ represents the coordinates of the transducer element $(s_x, s_y)$, and $(x_0, y_0)$ represents the center of the subaperture. In this expression, the first term represents the transmit path ($l_1$ in Figure 2.2) and the second term represents the receive path ($l_4$ in Figure 2.2). Since the delay of the first stage does not depend on the location of subaperture in the transducer array, the same set of delay values can be stored and shared among all subapertures.

The second stage beamforming can be expressed as:

$$F_2(m_x, m_y, m_z; t) = \sum_{n_x=m_x-(N_x(m_z)-1)/2}^{m_x+(N_x(m_z)-1)/2} \sum_{n_y=m_y-(N_y(m_z)-1)/2}^{m_y+(N_y(m_z)-1)/2} \tag{3.3}$$
$$a_2(n_x - m_x, n_y - m_y; m_z) \cdot F_1(n_x, n_y, t - \tau_2(n_x, n_y, m_x, m_y, m_z))$$

where $F_1$ is the partial beamformed results of the first stage, $(m_x, m_y, m_z)$ represents the index of the imaging focal point, $N_x(m_z)$ and $N_y(m_z)$ represent the number of VEs contributing at depth $m_z$ in $x$ direction and $y$ direction, respectively, and $a_2$ is the 2-D apodization window of size $N_x(m_z) \times N_y(m_z)$. Note that the size of the apodization window changes with the depth since the number of virtual elements contributing to an imaging focal point changes.

For the second stage beamforming, delay is given by:

$$\tau_2(n_x, n_y, m_x, m_y, m_z) = \frac{2 \times \sqrt{(z_m - d_p)^2 + (x_m - x_n)^2 + (y_m - y_n)^2}}{c} \tag{3.4}$$

where $(x_m, y_m, z_m)$ represents the coordinates of the imaging focal point $(m_x, m_y, m_z)$, and $(x_n, y_n, d_p)$ represents the coordinates of VE for subaperture $(n_x, n_y)$. The delays in the second stage depend on the distance between an imaging focal point and VE,

shown as paths $l_2$ and $l_3$ in Figure 2.2. Once the focal depth is fixed, the number of imaging points that are affected by each VE is fixed. Since the relative distance between each imaging point and VE remains the same, the delay values for the second stage are the same across different subapertures, and only one set of delay values need to be stored. For a system with 30×30 subapertures and 32×32 transducer elements within one subaperture, if the VE is located at 31.1 mm, the number of required delay values is 32×32=1024 for the first stage, which can be further reduced to 136 by using symmetry (8-way symmetry, plus the diagonal). If the maximum imaging depth is 100 mm, then the numbers of delay values for the second stage is 1.4M ($\sum_{m_z=1}^{M_z} N_x(m_z) \times N_y(m_z)$), which can be reduced to 219K by using symmetry. So storing the pre-calculated delay values in the LUT is more efficient than calculating delay values in real time.

### *3.2.1 Complexity Analysis*

For a subaperture of size $S_x \times S_y$, if the number of points in a scanline is $M_z$, the total number of computations in the first stage is given by:

$$N_{1st} = M_x \times M_y \times S_x \times S_y \times M_z \tag{3.5}$$

where $M_x$ and $M_y$ are the number of scanlines (same as the number of subapertures) in lateral and elevational dimensions.

In the second stage, where dynamic receive is applied, the beamforming process computes the weighted sum of all the wavefronts that contribute to the imaging focal point. Thus the computation complexity is determined by the number of VEs that contributes to a point, and so the number of computations is smaller for the points closer to the VE, but increases quadratically as the points move further away. The

total number of computations for the second stage is:

$$N_{2nd} = M_x \times M_y \times \sum_{m_z=1}^{M_z} N_x(m_z) \times N_y(m_z) \tag{3.6}$$

For a system with 30×30 subapertures, 32×32 transducer elements within one subaperture, and 100 mm of imaging depth ($M_z$ = 5195 for 40 MHz ADC rate), the number of multiply-accumulate operations for the first stage is 4.79 billion and for the second stage is 3.27 billion.

### *3.2.2 Receive Elements Reduction in the First Stage*

Our baseline configuration consists of 32×32 transducer elements for transmit as well as receive. Since the number of active receive elements directly affects the complexity of the first stage, we reduce the number from 32×32 to 16×16 by increasing the spacing to $2\lambda$, where one of every four elements is used. while keeping aperture size the same. With this reduction, the number of multiply-accumulate operations in the first stage is reduced from 4.79 billion to 1.20 billion. Reducing it any further degrades the performance significantly and is not considered here.

### *3.2.3 Separable Beamforming in the Second Stage*

In order to reduce the complexity of second stage beamforming that involves dynamic receive beamforming, we utilize separable beamforming that was derived in [10]. The key idea is to decompose 2-D array beamforming into two stages of 1-D array beamforming.

First we use the outputs from the first stage SASB to beamform along $x$ direction. Then the partial beamformed result is used to beamform along $y$ direction. The delay applied is the decomposition of 3-D SASB second stage delay $\tau_2(n_x - m_x, n_y - m_y, m_z) \cong \tau_{2,x}(n_x - m_x, m_z) + \tau_{2,y}(n_y - m_y, m_z)$. The decomposition is chosen based on the root

mean square error (RMSE) minimization as discussed in [11]. The two-step separable beamforming process can be implemented as follow:

$$F_{2,x}(m_x, n_y, m_z; t) = \sum_{n_x=m_x-(N_x(m_z)-1)/2}^{m_x+(N_x(m_z)-1)/2} \tilde{A}(n_x - m_x; m_z) \cdot F_1(n_x, n_y, t - \tau_{2,x}(n_x - m_x, m_z)) \tag{3.7}$$

$$F_{2,y}(m_x, m_y, m_z; t) = \sum_{n_y=m_y-(N_y(m_z)-1)/2}^{m_y+(N_y(m_z)-1)/2} \tilde{A}(n_y - m_y; m_z) \cdot F_{2.1}(m_x, n_y, t - \tau_{2,y}(n_y - m_y, m_z)) \tag{3.8}$$

$F_{2,x}(m_x, n_y, m_z; t)$ represents the partial beamformed results in x direction, and $F_{2,y}(m_x, m_y, m_z; t)$ represents the final beamforming output. The apodization coefficient $\tilde{A}(x; m_z)$ represents the value with index $x$ in a 1D window (such as Hamming window) whose length depends on the number of VEs contributing to depth $m_z$.

For the original SASB second stage beamforming, the total number of delay-and-sum operations at depth $m_z$ is $(M_x M_y N_x(m_z) N_y(m_z))$. Separable beamforming reduces the number of computation to $(M_x N_x(m_z) N_y(m_z) + M_x M_y N_y(m_z))$. The reduction in each depth is $(N_x(m_z) M_y / (N_x(m_z) + M_y))$. The LUT size at depth $m_z$ is $N_x(m_z) + N_y(m_z)$ due to separable beamforming, resulting in a $(N_x(m_z) \times N_y(m_z) / (N_x(m_z) + N_y(m_z)))$ saving in the LUT size. For a system with 32×32 transmit elements and 16×16 receive elememts, the number of computations in the second stage reduces from 3.27 billion to 339 million, and the LUT size reduces from 1.43 million to 149 thousand. Thus separable beamforming achieves a 9.6× reduction in both number of computations and LUT size. The proposed method with the system configuration listed in Table 3.1 reduces the complexity by 14× compared to the separable SAU systems in [10] that has comparable image quality.

Table 3.1: System Configuration

| **Parameters** | **Values** |
|---|---|
| Pitch ($\mu$m) | 335.0 |
| Array size (element) | 90×90 |
| Subaperture Number | 30×30 |
| Subaperture Size (element) | 32×32 |
| Transmit Size (element) | 32×32 |
| Receive Size (element) | 16×16 |
| Center frequency (MHz) | 4 |
| A/D sampling rate (MHz) | 40 |
| 6 dB transducer bandwidth (MHz) | 3.6 |
| F-number | 3 |
| VE location depth (mm) | 31.1 |

### *3.2.4 Simulation Results*

To verify the quality of images generated by the proposed 3-D SASB method, we present images of cysts using Field-II platform [19,20]. We study the effect of number of receive elements and the depth on the imaging quality. The system configuration is given in Table 3.1.

To compare the cyst images, we introduce the metric, contrast-to-noise ratio (CNR) to quantify the performance. CNR is defined as:

$$CNR = \frac{|\mu_{cyst} - \mu_{bgnd}|}{\sqrt{\sigma_{cyst}^2 + \sigma_{bgnd}^2}} \tag{3.9}$$

where $\mu_{cyst}$ and $\mu_{bgnd}$ correspond to the average brightness of the cyst volume and the background speckles, while $\sigma_{cyst}$ and $\sigma_{bgnd}$ represent the standard deviation. For

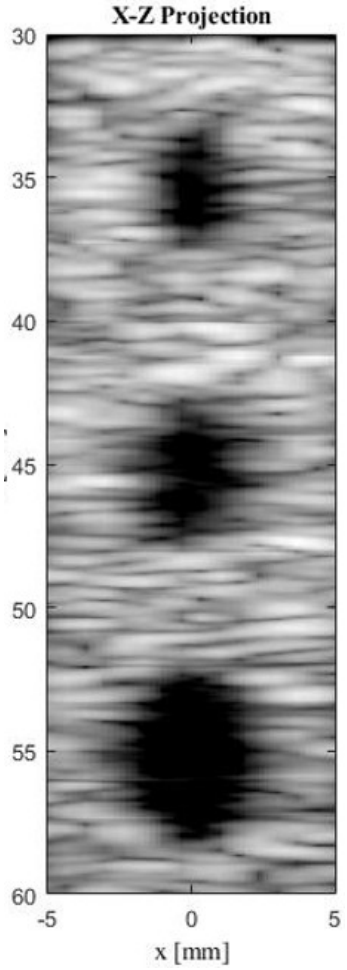
the two cyst images in Figure 3.1, we use 32×32 elements to transmit and 16×16 and 32×32 elements to receive. We see that there is minor difference in CNR when the number of receive elements is dropped to 16×16, and since the 16×16 case has significantly lower complexity, we conclude that 32×32 for transmit and 16×16 for receive is the best choice. Figure 3.2 compares the cyst performance at different depths. At shallower depths (cyst centers at 35 mm), the CNR is 3.43. At deeper depths (cyst centers at 75mm), the CNR is 2.70. Comparing with the simulations results of SAU shown in [43], we see that SASB has less degradation in imaging quality in deeper depths compared to SAU.

## 3.3 Multiple Transmit and Multiple Receive Firing Scheme

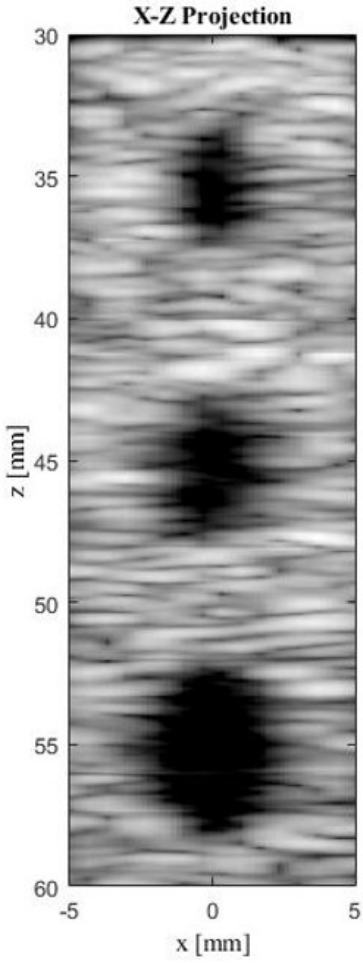
The MTMR firing scheme uses the concept of multiple line transmission (MLT), which has been shown to be an efficient way of increasing volume rate [44–47]. In the proposed MTMR firing scheme, we increase the number of firing subapertures from one to four (factor of two increase in both X and Y dimensions). Thus, the time taken by the transmit and receive process is reduced by 4×, and the volume rate is increased by 4×. Firing more than two subapertures simultaneously in each dimension results in a drastic increase in the computational complexity of the front-end, which is not acceptable. Also, increasing the number of simultaneously firing subapertures to more than four results in serious degradation in the image quality and hence is not considered here.

### *3.3.1 Multiple Transmit and Multiple Receive*

In the traditional single transmit and single receive (STSR) scheme, only one subaperture transmits and receives at a time. Figure 2.2 showed an example with two VEs. In the MTMR firing scheme, where two subapertures fire simultaneously, apart

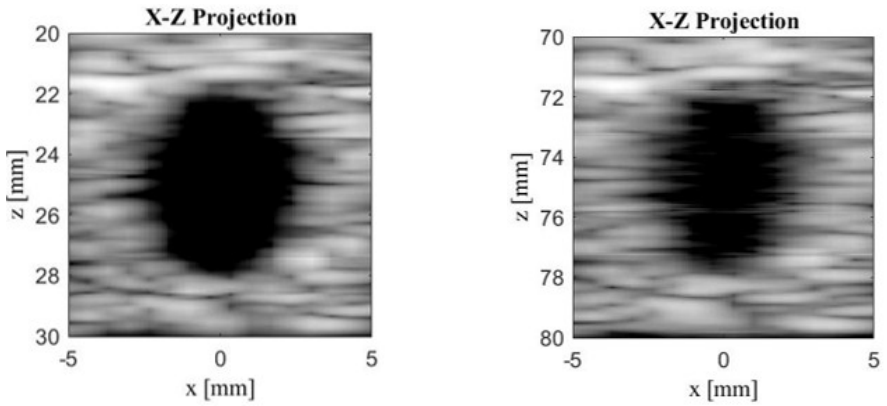



(a) 16×16 active receive elements. The average CNR is 2.11.




(b) 32×32 active receive elements. The average CNR is 2.12.

Figure 3.1: Cyst Performance Comparison Between Different Number of Receive Elements for STSR2.




(a) Cyst image centered at depth 25 mm. The CNR is 3.43.

(b) Cyst image centered at depth 75 mm. The CNR is 2.70.

Figure 3.2: Cyst Performance Comparison Between Shallow and Deep Depths for STSR2.

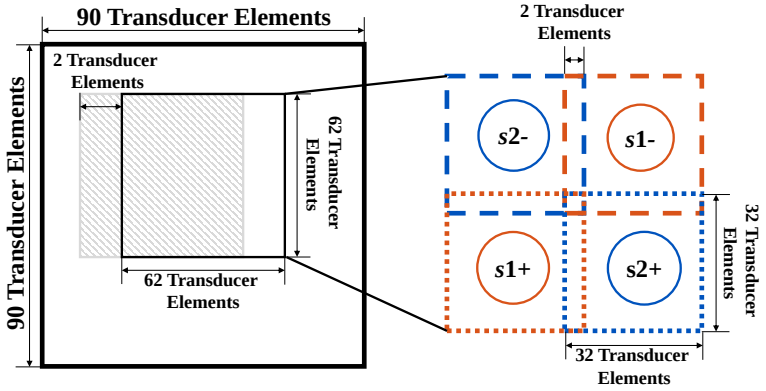



Figure 3.3: Layout of Four Subapertures in MTMR Firing Scheme.

from the transmit and receive paths for VE A and VE B, there are two additional delay paths associated with VE A and VE B: $l_{AB} = l_2 + l_7$ and $l_{BA} = l_6 + l_3$, where $l_{AB}$ represents the path when the wavefront is transmitted by VE A and received by VE B, and $l_{BA}$ represents the path when the wavefront is transmitted by VE B and received by VE A. These additional delay calculation paths apply to all image voxels

which are located in the overlapped region that is serviced by the spherical wavefronts emitted by both VE A and VE B.

In the proposed MTMR firing scheme, a group of four subapertures, which are located in different quadrants of the 2-D transducer array, transmit and receive simultaneously. After each firing, the group shifts by two elements and then fires again. The group transits in line scan mode from left to right and top to bottom, to cover the whole array. In the setup shown in Figure 3.3, each subaperture has 32×32 transducer elements. Two simultaneously firing subapertures in the same direction overlap by two rows (or two columns) of elements. Thus, a group of four subapertures consist of 62×62 active transducer elements. Since the 2-D transducer array consists of 90×90 transducer elements, a group of four subapertures fires 15 by 15, which is 225, times to cover the whole array.

Compared to the STSR-based firing scheme, each MTMR receive subaperture receives signals that are mixed with waveforms transmitted by four simultaneously firing subapertures. Signals transmitted and received by different subapertures cause interference, which may cause artifacts in the image volume. If a conventional sinusoid excitation is used, there will be significant interference resulting in image quality degradation when the imaged voxel is far from the VE.

To maintain good image quality with MLT, different methods have been proposed to reduce the interference between simultaneous transmissions. The method in [48] combines MLT with linear constraint minimum variance beamforming and the method in [49] applies a filtered-delay multiply-and-sum beamforming (F-DMAS). These two methods reduce the interference but have high front-end computational complexity. The method in [50] applies orthogonal frequency division multiplexing (OFDM) to isolate the signals in each transmission, but that leads to different brightness and lateral resolution across the image. The method in [51] makes use of the

transducer geometry to implement spatial separation. While it successfully reduces the interference, it imposes a restriction on the layout of the transducer array. In this work, we describe a simpler way to reduce interference through the use of linear FM chirps. This method provides good image quality without increasing the front-end beamforming computation requirement.

### *3.3.2 Coded Excitation Using Linear Chirp*

A linear chirp is a frequency modulated signal that is popular in radar and communication systems. It is also being applied in ultrasound imaging to increase SNR and volume rates in 3-D imaging [52, 53]. Compared to sinusoidal excitation, the frequency of a chirp signal changes linearly with time. A real-valued modulated chirp signal is expressed as:

$$s(t) = a(t) \cdot \cos\left[2\pi\left(f_0 t + \frac{B}{2T}t^2\right)\right], \quad -\frac{T}{2} \leq t \leq \frac{T}{2} \tag{3.10}$$

where $a(t)$ is the magnitude of the chirp, $f_0$ is the center frequency, $B$ is the bandwidth of the frequency band, and $T$ is the time duration. The performance of a chirp signal depends on $T$, $B$, and their product TBP. Low TBP leads to low pulse compression, modest increase in SNR, and large cross-correlation, causing high interference. High TBP results in high pulse compression, but high axial sidelobe levels [52,54]. Limited bandwidth also leads to a long transmit duration, which restricts imaging in shallow regions.

In an imaging system that uses sinusoidal excitation, the received signals are directly beamformed. But in a system based on linear chirps, the received signals have to be decoded either before or after beamforming. We choose to decode after the first beamforming stage to avoid increasing the front-end complexity. The linear chirp $s_a(t)$ is decoded using a matched filter [52, 55]. The convolution of $s_a(t)$ and

$s_a(-t)$ results in the auto-correlation function of $s_a(t)$, referred to as $AC_a(t)$. The mainlobe width of $AC_a(t)$ is proportional to the reciprocal of the bandwidth $(1/B)$ and the peak amplitude is proportional to the time duration $T$. Now, if the chirp $s_b(t)$ is passed through $s_a(-t)$, we obtain the cross-correlation, $CC_{ab}(t)$. Since the amplitude of $CC_{ab}(t)$ is lower than $AC_a(t)$, $s_a(t)$ can be extracted from the mixed signal containing both $s_a(t)$ and $s_b(t)$.

The chirp design parameters are: (1) frequency band, (2) time duration $T$, and (3) chirp rate $B/T$. The frequency band has two parameters: the band center frequency $f_0$, and bandwidth $B$. The chirp rate can be either positive or negative, depending on whether the frequency increases or decreases.

Current chirp designs focus on changing these three parameters to reduce cross-correlation. For instance, when only two chirps are required, we can choose their time duration and frequency band to be the same while their chirp rates are opposite. The cross-correlation can be also reduced by dividing the available transducer bandwidth and allocating it to different chirps [53]. The method in [56] designed waveforms that are formed by the combinations of two chirps with different time durations and different chirp rates. While this method provided a large number of different FM chirps to support simultaneous transmission, the cross-correlation was high and so the interference between different firings was large. Non-linear chirps were investigated in [57]. Compared to linear chirps, non-linear chirps have lower cross-correlation, but the difference in effective center frequencies is larger, which results in larger difference in resolution of images generated by different chirps.

In this work, we design four linear chirps to satisfy the requirements of 3-D SASB using MTMR with four simultaneous firings. The chirp parameters are selected to satisfy the following constraints:

1) The brightness in each image volume is similar

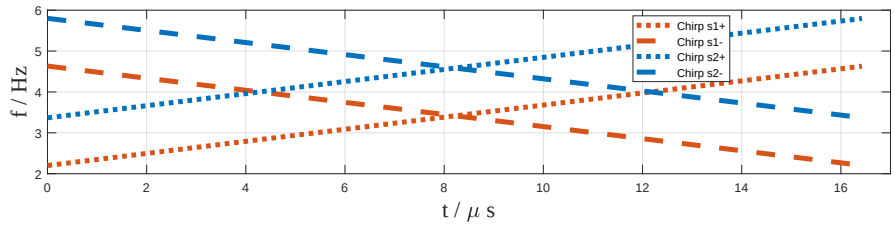
2) The axial resolution and the lateral resolution in each image volume are similar

3) The interference between simultaneously firing subapertures is minimized

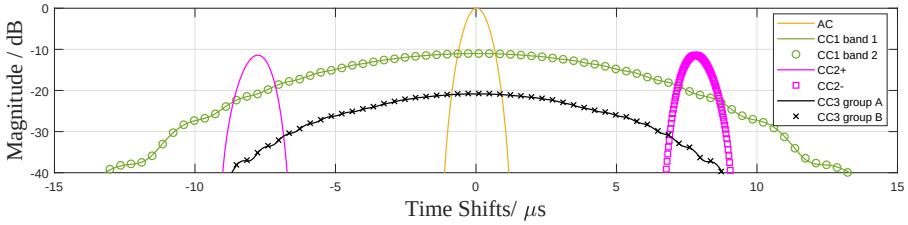
Since the axial resolution depends on the bandwidth of the frequency band, we choose all chirps to have the same bandwidth $B$. To ensure that the brightness in each image volume is the same, subapertures that form one image volume should use chirps with the same time duration $T$. In Figure 3.4(a), chirps $s1+$, $s1-$, $s2+$, and $s2-$ have the same bandwidth $B$ and the same time duration $T$.

In chirp based coded excitation, interference is caused by cross-correlation between different chirps. The cross-correlation is a function of TBP and the overlap between frequency bands, with lower TBP and lower frequency overlap generating lower cross-correlation. Even though it is desirable to reduce the frequency overlap as much as possible, using chirps with completely non-overlapped frequency bands results in poor image quality. Because the transducer bandwidth is limited, if the two frequency bands are non-overlapped, the available bandwidth for each chirp becomes too small. Chirps with low bandwidth generate images with poor axial resolution. To add to the complexity, a large difference in center frequencies causes large difference in lateral resolution. So, the size of the overlapped frequency band has to be appropriately chosen to achieve a balance between low cross-correlation and small difference in the center frequencies.

Based on this analysis, we divide the transducer's frequency band into two bands with appropriate frequency overlap. In each frequency band, we choose two chirps with opposite chirp rates. These four chirps are the excitation waveforms for the four simultaneously firing subapertures. The expressions for the four chirps are given by:



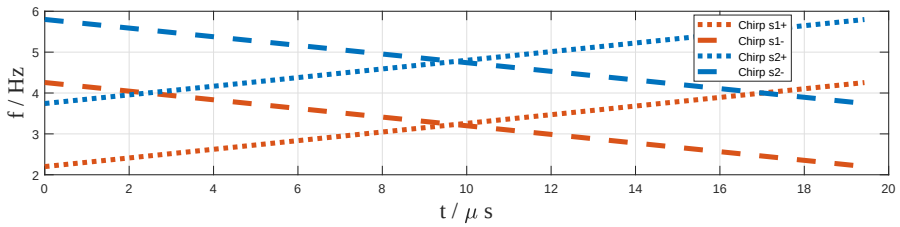

(a) Frequency distribution of four chirps with 50% frequency overlap; TBP = 40. The bandwidth of the two frequency bands is 2.4 $MHz$, and the frequency overlap is 1.2 $MHz$.



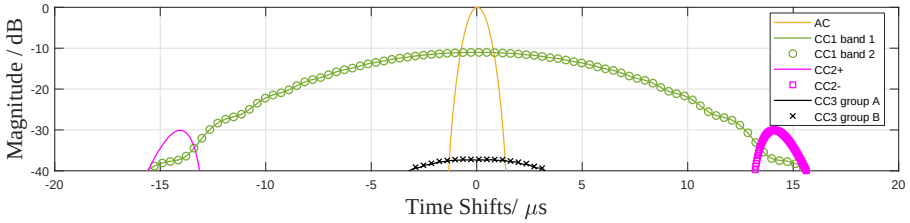

(b) The auto-correlation, AC, and cross-correlation CC1, CC2, and CC3 as a function of time shifts. CC1 band 1 is between chirps {$s1+$, $s1-$} and CC1 band 2 is between chirps {$s2+$, $s2-$}. CC2+ and CC2- are between chirps {$s1+$, $s2+$} and {$s1-$, $s2-$}, respectively. Both CC1 and CC2 are 11 dB lower than AC. CC3 group A and CC3 group B are between chirps {$s1+$, $s2-$} and chirps {$s1-$, $s2+$}, respectively, which are 20 dB lower than AC.

Figure 3.4: Frequency Distribution, Auto-correlation, and Cross-correlations of Chirps with 50% Overlap.

$$
\begin{aligned}
s1+(t) &= A \cdot \cos[2\pi(f_1 t + \frac{B}{2T}t^2)], \quad -\frac{T}{2} \le t \le \frac{T}{2} \\
s1\text{-}(t) &= A \cdot \cos[2\pi(f_1 t - \frac{B}{2T}t^2)], \quad -\frac{T}{2} \le t \le \frac{T}{2} \\
s2+(t) &= A \cdot \cos[2\pi(f_2 t + \frac{B}{2T}t^2)], \quad -\frac{T}{2} \le t \le \frac{T}{2} \\
s2\text{-}(t) &= A \cdot \cos[2\pi(f_2 t - \frac{B}{2T}t^2)], \quad -\frac{T}{2} \le t \le \frac{T}{2}
\end{aligned}
\tag{3.11}
$$




(a) Frequency distribution of four chirps with 25% frequency overlap; TBP = 40. The bandwidth of the two frequency bands is 2.06 $MHz$, and the frequency overlap is 0.51 $MHz$.




(b) CC1, CC2, and CC3 are generated from the same set of chirps as in Figure 3.4. Here, CC1 is 11 dB lower than AC, CC2 is 30 dB lower than AC, and CC3 is 37 dB lower than AC.

Figure 3.5: Frequency Distribution, Auto-correlation, and Cross-correlations of Chirps with 25% Overlap.

where $f_1$ is the center frequency for chirps $s1+$ and $s1-$, and $f_2$ is the center frequency for chirps $s2+$ and $s2-$. We use '+' to represent chirps with positive chirp rates, and '-' to represent chirps with negative chirp rates.

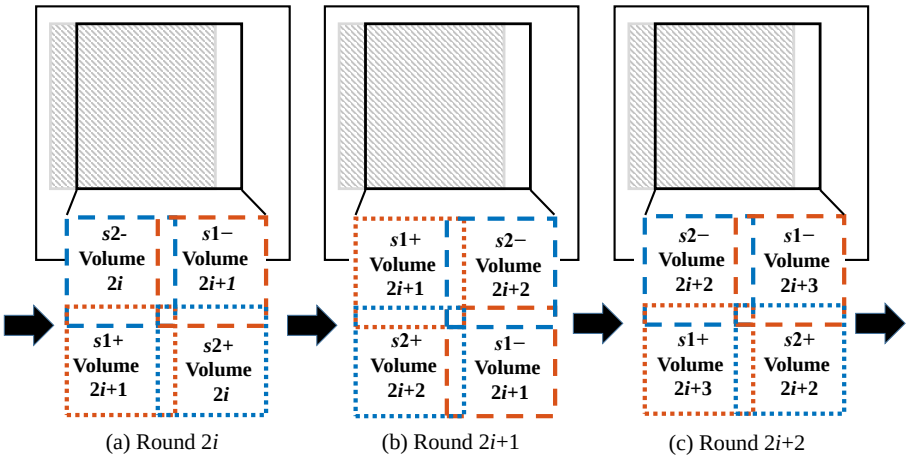
Figure 3.4 and Figure 3.5 show two groups of chirps with the same TBP. For frequency overlap of 50%, $f_1 = 3.4\ MHz$, $f_2 = 4.6\ MHz$, the bandwidth of each chirp is 2.4 $MHz$, and for frequency overlap of 25%, $f_1 = 3.23\ MHz$, $f_2 = 4.77\ MHz$, and the bandwidth is 2.06 $MHz$. There are three forms of cross-correlations: (a) the cross-correlation between opposite chirps in the same band, referred to as CC1; (b) the cross-correlation between chirps with same rate but in different frequency bands, referred to as CC2; (c) the cross-correlation between chirps with opposite

rate in different bands, referred to as CC3. CC1 is determined by TBP and CC2 is determined by the frequency overlap. CC3 depends on factors that affect both CC1 and CC2. However, since CC3 has lower amplitude compared to both CC1 and CC2, and CC3 reduces as CC1 and CC2 reduce, CC3 is not considered as a design metric. We see that for 50% frequency overlap (see Figure 3.4(b)), CC1 and CC2 are 11 dB lower than AC, and CC3 is 20 dB lower than AC. For frequency overlap of 25% (see Figure 3.5(b)), CC1 is 11 dB, CC2 is 30 dB, and CC3 is 37 dB lower than AC. In Section 3.3.6, we show through cyst simulations that a larger CC2 caused by a larger frequency overlap results in degraded image quality.

### *3.3.3 Overlapped Firing Scheme*

The lateral resolution of ultrasound imaging depends on the center frequency of the chirps. We see that if chirps with different center frequencies are used to form an image volume, there will be differences in the lateral resolution. To avoid this difference, we propose to generate each image volume using two chirps with opposite rates in the same frequency band. We maintain a high volume rate by overlapping the transmit and receive process of two consecutive image volumes.

In the proposed method, in every round of firing, four subapertures, with different chirps, transmit and receive simultaneously, capturing the information to generate two image volumes. For instance, in round $2i$ (shown in Figure 3.6), the subapertures on the left top and right bottom, which use $s2-$ and $s2+$, respectively, generate volume $2i$, and subapertures on the left bottom and right top, which use $s1+$ and $s1-$, respectively, generate volume $2i + 1$. After these four subapertures complete all 225 firing events that are required to cover the whole array, the two sets of chirps switch subapertures, and start a new round of transmit and receive processes. Specifically, in round $2i + 1$, $s1+$ and $s1-$ switch to the subapertures on the left top and right




(a) Round $2i$ (b) Round $2i$+1 (c) Round $2i$+2

Figure 3.6: The Overlapped Transmit and Receive Process.

bottom, respectively, and generate volume $2i + 1$, while chirps $s2+$ and $s2$- switch to subapertures on the left bottom and right top, respectively, and generate volume $2i + 2$. Thus, both sets of chirps cover the whole volume after two rounds of firings. Since each volume is generated using chirps with the same center frequencies, the asymmetric PSF is avoided and the lateral resolution is the same throughout the image volume. As the generation of two image volumes is interleaved in time, a new image volume is generated after each round of transmit and receive events.

### *3.3.4 Sparase Array Design*

In Section 3.2.2, we reduce the number of elements involved in the first beamforming stage by using a 16×16 array with $2\lambda$ spacing as the receive subaperture. However, this array cannot be used for MTMR since it results in severe grating lobes that degrade image quality.

In this section, we present the design of a sparse array with 256 active elements for chirp excitation that avoids the grating lobes caused by MTMR. The elements of the sparse array are organized in a non-uniform manner, such that it has a beam pattern

comparable with the $\lambda$ spaced dense array in the directions of other simultaneously firing subapertures.

Sparse arrays have also been used to reduce the number of active transducer elements in ultrasound and sonar systems [58, 59]. Different methods of sparse array design have been proposed to reduce the number of active elements without reducing resolution or increasing sidelobe levels [60–63]. However, many of the existing designs are based on non-grid transducer arrays which cannot be applied to our subaperture-based firing scheme.

In a uniformly distributed array without steering, grating lobes occur at the following angles [28]:

$$\sin\phi = \frac{\lambda}{d} \cdot m, \quad m = \pm 1, \pm 2, \pm 3... \tag{3.12}$$

where $\phi$ is the elevational angle of grating lobes, $d$ is the spacing between transducers, and $\lambda$ is the wavelength. Since the chirp frequency varies in a band, the mainlobe energy ($m = 0$ in Eq. 3.12) focuses at $0^o$, but the grating lobes exist in a range of angles around the angle corresponding to the center frequency. We apply a Hamming window, so that the energy of the grating lobes is maximized at the angle corresponding to the chirp's center frequency. The sparse array design optimization focuses on minimizing the grating lobes at the chirp's center frequency.

Recall that in MTMR, there are four chirps organized into two bands with center frequencies $f_1$ and $f_2$, where $f_1 < f_2$. For the uniformly distributed array without steering, the waveforms with wavelengths higher than the element spacing do not generate grating lobes. But the waveforms with wavelengths smaller than the element spacing may have grating lobes that fall into the field-of-view. We denote the wavelength corresponding to $f_2$ as $\lambda_0$.

Our sparse array is based on a transducer array with element spacing of at most $\lambda_0$. Thus, the grating lobes of the transmit subaperture are located in the vicinity

of $\pm 90^o$. The transmit power is focused around the VE, which is at $0^o$ elevational angle. Since the receive subaperture has active transducer elements with $2\lambda_0$ spacing, it could cause grating lobes around $\pm 30^o$ elevational angle.

The grating lobe problem is exacerbated in MTMR since the receive subaperture receives scattered signals transmitted from all four subapertures. Although the cross-correlations of linear chirps are lower than the auto-correlation, the interference caused by grating lobes still has a high amplitude, which causes image artifacts. Here, the possible lateral angles at which the signals arrive are in vicinities of $0^o$, $\pm 45^o$, $\pm 90^o$, $\pm 135^o$, $180^o$. At all other lateral angles, the grating lobe is small compared to the main lobe, and is not considered.

To reduce the grating lobe levels, prior work described a bin-based random array with reduced number of elements [64]. There, a 2-D transducer array is divided into small bins of 2×2 elements. In each bin, one of the four elements is randomly chosen as the active element of the receive aperture. Since the grating lobes do not sum coherently, as in the case of the uniformly distributed array, the grating lobe levels are reduced. Unfortunately, the power of the grating lobes spreads, resulting in increased sidelobe levels.

To address this issue, we set up an optimization problem that identifies locations of the active transducer elements in the bin-based random array such that the sidelobe levels are minimized. This optimization is equivalent to setting a target beam pattern value and minimizing the difference between the transducer array's beam pattern in terms of the center frequency of the chirp and the target beam pattern. Let $(x, y)$ denote the coordinate of a transducer element, and let $(\mathbf{x}, \mathbf{y})$ denote the locations of 256 transducer elements in a 32×32 array with $\lambda_0$ spacing. For an elevational angle $\theta$ and a lateral angle $\phi$, let $\mathrm{BP}(\theta, \phi; \mathbf{x}, \mathbf{y})$ denote the beam pattern of the transducer array in terms of the center frequency of the chirp, and $\mathrm{BP}_0(\theta, \phi)$ denote the target

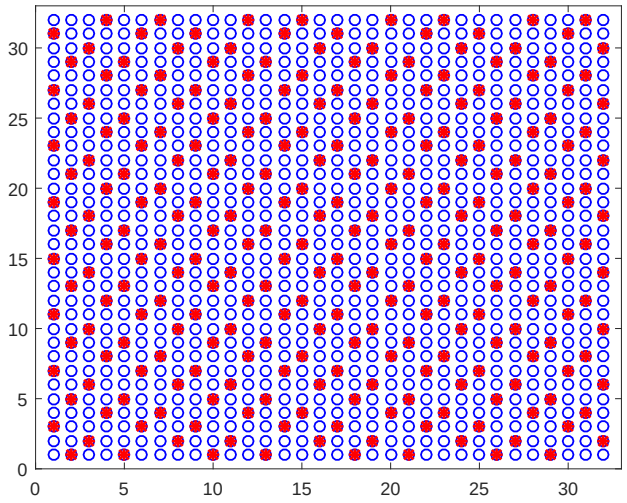



Figure 3.7: Layout of the 256 Active Transducer Elements After Optimization.

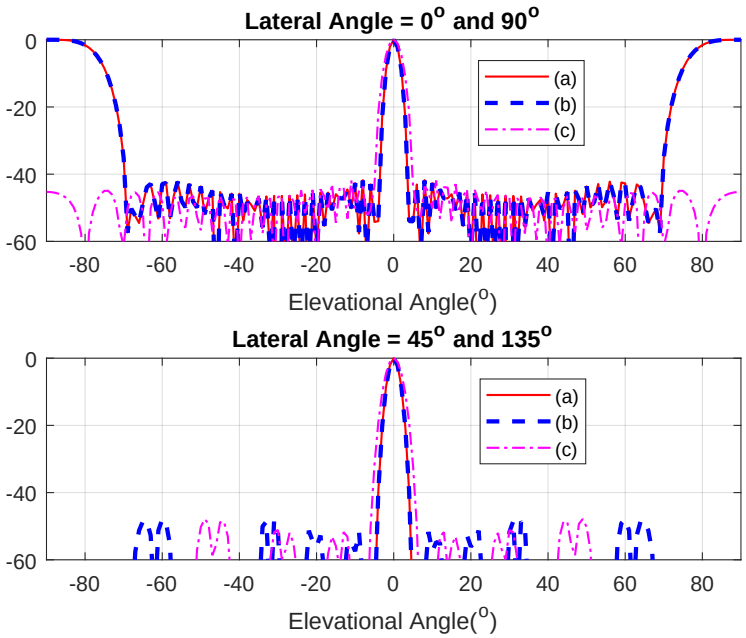
beam pattern value. The target value is set to -40 dB, which is the dynamic range used for imaging. The objective function is given by:

$$\min f(\mathbf{x},\mathbf{y}) = \sum_{\theta}\sum_{\phi}(\mathrm{BP}(\theta,\phi;\mathbf{x},\mathbf{y}) - \mathrm{BP}_0(\theta,\phi))^+ \tag{3.13}$$

$$\mathrm{BP}(\theta,\phi;\mathbf{x},\mathbf{y}) = |\sum_{x\in\mathbf{x}}\sum_{y\in\mathbf{y}} a(x,y)e^{-j2\pi(x\cos\theta\sin\phi+y\cos\theta\cos\phi)}| \tag{3.14}$$

where $|\cdot|$ stands for absolute value, $a(x,y)$ is the apodization coefficient for the signal received by the transducer element located at $(x,y)$. $()^+$ is a function that only returns a positive value and thus forces the optimization engine to only sum up the beam patterns that are larger than the target value. The values of $\theta$ range from $10^o$ to $90^o$ in steps of $1^o$ (the main lobe ranges from $0^o$ to $10^o$), and the values of $\phi$ are in $\pm5^o$ range of lateral angles $\{0^o, \pm45^o, \pm90^o$, and $\pm135^o\}$ in steps of $1^o$.

We solve the optimization problem with simulated annealing, as in [59, 65–67]. The layout of the 256 active transducer elements after the optimization is shown in




The element spacing of the basic grid is $\lambda_0$. (a) 32×32 uniformly distributed array, the chirp center frequency is $f_2$; (b) Sparse array with 256 elements, the chirp center frequency is $f_2$; (c) Sparse array with 256 elements, the chirp center frequency is $f_1$.

Figure 3.8: Beam Pattern Comparison as A Function of Elevational Angle for Different Transducer Array Layouts.

Figure 3.7 where the active elements are marked as red dots. Comparison of the beam pattern between the optimized sparse array scheme and the original dense array with 32×32 elements is shown in Figure 3.8. We see that, for chirps with center frequency $f_2$, all sidelobes are lower than -40 dB. The grating lobes in lateral angles of $0^o$ (which is same as $\pm 90^o$) and $\pm 45^o$ (which is same as $\pm 135^o$) due to other simultaneous firings have been removed. While there are grating lobes near $\pm 90^o$ in the elevational direction, these do not cause any image artifacts, and are hence ignored. For chirps with center frequency $f_1$, the sidelobes do not increase and there are no grating lobes near $\pm 90^o$. We refer to the resulting sparse array scheme as MTMRS.

### *3.3.5 Computational Complexity and Volume Rate*

The first beamforming stage of 3-D SASB is computed inside the transducer probe and the second beamforming stage is computed in an off-line processing unit. Thus, it is important to reduce the computational complexity of the first beamforming stage. In STSR2, we use 16×16 active transducer elements with $2\lambda$ spacing as the receive subaperture, and 32×32 elements with $\lambda$ spacing as the transmit subaperture. The number of subapertures in this firing scheme is 900. Considering the ADC rate of 40 $MHz$, the number of imaging points on one RF-line is 5195, and thus the number of multiply-and-accumulate (MAC) operations to generate one image volume is $1.19\times10^9$ in the first stage.

In MTMR-based firing scheme, as the number of elements in the sparse array is the same as the uniformly distributed array in STSR2, the number of MAC operations in the first beamforming stage to generate one image volume is also the same. In the second beamforming stage, the number of computations is higher than STSR2 due to more delay paths. Since linear chirp is used as the excitation, the received signals have to be convolved with the matched filter. To avoid the large increase in the computational complexity at the front-end, the matched filtering process is placed after the first beamforming stage, so that it can be computed in the separate computing unit along with the second beamforming stage.

The volume rate depends on the number of transmit and receive events as well as the round-trip propagation time. In STSR-based firing schemes, the number of transmit and receive events is same as the number of subapertures, which is 30×30. For maximum imaging depth of 10 $cm$, the volume rate of STSR2 is 8.56 volumes per second. In MTMR-based firing schemes, four subapertures transmit and receive

Table 3.2: System Configuration

| **Parameters** | **Values** |
|---|---|
| Array size (element) | 90×90 |
| Number of subapertures | 900 |
| Transmit size (element) | 32×32 |
| Transducer Element Spacing ($\mu m$) | 335 |
| Center frequency ($MHz$) | 4 |
| A/D sampling rate ($MHz$) | 40 |
| 6 dB transducer bandwidth ($MHz$) | 3.6 |
| f number | 3 |
| Speed of sound ($m/s$) | 1540 |
| Time-bandwidth product | 40 |
| Frequency band 1 ($MHz$) | 2.2 - 4.6, $f_1 = 3.4$ |
| Frequency band 2 ($MHz$) | 3.4 - 5.8, $f_2 = 4.6$ |
| Dynamic Range ($dB$) | 40 |
| Maximum imaging depth ($mm$) | 100 |

simultaneously, and thus the number of transmit and receive events is reduced by 4× to 15×15, and the volume rate is increased to 34.2 volumes per second.

### *3.3.6 Simulation Results*

We simulate the proposed methods with the ultrasound simulation platform Field II [19, 20]. We compare the quality of cyst images generated by the proposed scheme and competing schemes. The configuration of the 3-D ultrasound imaging system is shown in Table 3.2.

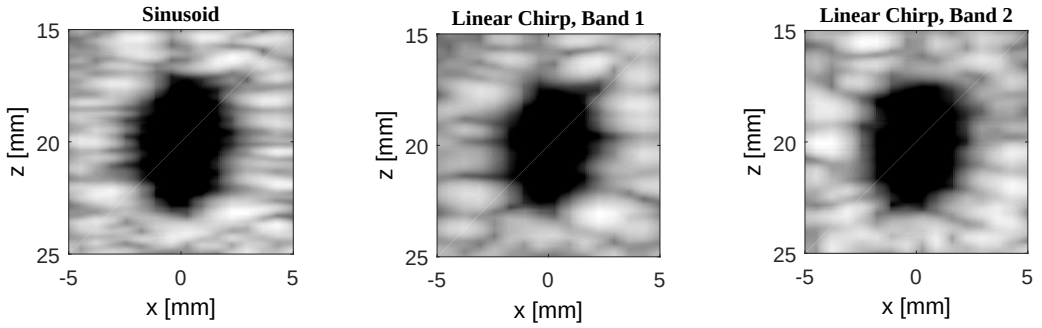
The transmit subaperture of all schemes consists of 32×32 elements with $\lambda_0 = 335\ \mu m$, where $\lambda_0$ is the wavelength corresponding to $f_2 = 4.6\ MHz$. The receive subaperture for each scheme is summarized below:

- **STSR2**: uniformly distributed array consisting of 16×16 active receive elements with $2\lambda_0$ spacing
- **MTMR2**: uniformly distributed array consisting of 16×16 active receive elements with $2\lambda_0$ spacing
- **MTMRS**: optimized sparse array with 256 active receive elements (proposed method)

The four-linear-chirp system has two frequency bands, with band 1 consisting of two chirps centered at $f_1 = 3.4\ MHz$, and band 2 consisting of two chirps centered at $f_2 = 4.6\ MHz$ (see Figure 3.4). This two-band system is suitable for kidney and obstetric B-mode imaging [68, 69]. However, applications require other center frequencies can be supported by changing the relevant parameters. The bandwidths of both frequency bands are $2.4\ MHz$, and the overlapped part is $1.2\ MHz$, which is 50% of the chirp bandwidth. We choose TBP = 40 to achieve a balance between the imaging depth and quality in a limited bandwidth system. The time duration of each chirp is $16.7\ \mu s$, which allows for minimum imaging depth of 13 $mm$.

### Linear Chirp v.s. Sinusoid

Figures 3.9 and 3.10 compare the image quality generated using the sinusoidal and linear chirp excitations when the cyst is located at 20 $mm$ (shallow region), and 75 $mm$ (deep region), respectively. The firing scheme for both types of excitations is MTMRS. Both linear chirp and the sinusoidal excitation achieve comparable image

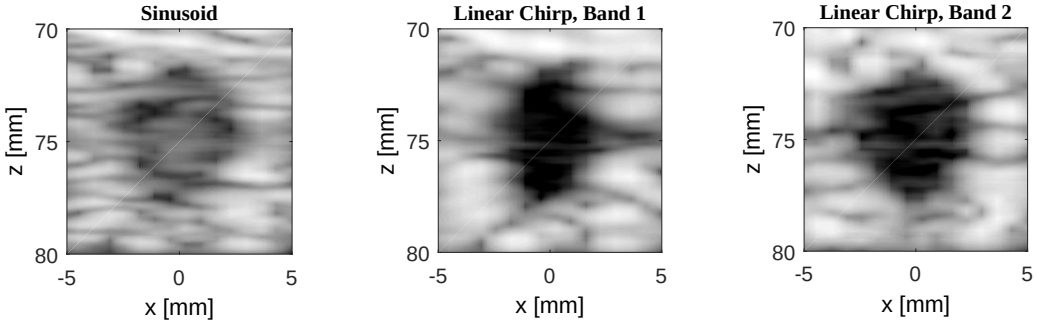



(a) Sinusoid at 4 $MHz$, CNR = 3.36.

(b) Chirps in band 1, CNR = 2.85.

(c) Chirps in band 2, CNR = 4.01.

Figure 3.9: Imaging Quality Comparison Between Sinusoid and Chirp Excitations with Cyst Located at 20 *mm*.



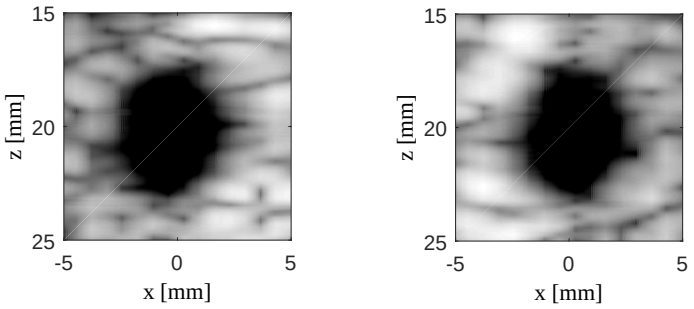

(a) Sinusoid at 4 $MHz$, CNR = 3.36.

(b) Chirps in band 1, CNR = 2.85.

(c) Chirps in band 2, CNR = 4.01.

Figure 3.10: Imaging Quality Comparison Between Sinusoid and Chirp Excitations with Cyst Located at 75 *mm*.

quality in the shallow region. This is because the first stage of SASB is a fixed transmit and receive process. The energy is maximized at VE and so there are no imaging artifacts due to interference for imaging points close to VE. Based on CNR, the image quality generated by the sinusoidal excitation is better than the cyst image generated by chirps in band 1, but is worse than the image generated by chirps in band 2. This is because the lateral resolution depends on waveform's center frequency.




(a) left half: band 2, right half: band 1.

(b) left half: band 1, right half: band 2.

Figure 3.11: Cyst Images Located at 20 *mm* Generated Using Chirps in Different Bands.

The center frequency of the sinusoidal excitation is 4 $MHz$, which is between $f_1 = 3.4$ $MHz$, and $f_2 = 4.6$ $MHz$. Since the mainlobe width of the sinusoidal excitation is smaller than the auto-correlation of the chirps, the sinusoidal excitation generates better axial resolution.

In the deep region, however, the image generated using sinusoidal excitation is a lot worse and the CNR value is reduced by half compared to that in the shallow region. As the distance between the imaging point and VE increases, the beamforming gain in the first beamforming stage decreases, resulting in more artifacts due to interference. In comparison, the cyst images generated by linear chirps have good quality due to lower interference.

To demonstrate that the image volume should be generated using chirps in the same frequency band, we generate cyst images using chirps in different frequency bands. In Figure 3.11(a), the left half is generated using band 2, and the right half is generated using band 1. The region generated using band 1 has higher brightness than the region generated using band 2. The brightness difference is also shown in

Figure 3.11(b), where the left half is generated using band 1, and the right half is generated using band 2.

In the MTMRS firing scheme with the sinusoidal excitation, signals received by four simultaneously firing subapertures are beamformed to generate one image volume. But this firing scheme leads to different brightness across an image volume if chirps with different center frequencies are used. For the imaging region where the brightness is lower, the dynamic range is smaller than the region with higher brightness, resulting in the potential loss of visibility of some low-amplitude scatterers.

To avoid generating image volumes using chirps with different center frequencies, we apply the overlapped firing scheme. In the overlapped firing scheme, each image volume is generated after two rounds of transmit and receive processes. During these two rounds, chirps with different center frequencies generate different image volumes, and each image volume is generated using two chirps in the same band, thereby avoiding the difference in brightness.

Furthermore, the overlapped MTMR firing scheme reduces the data acquisition time, making it less sensitive to tissue motion. The time to generate one volume is 58.4 ms. And the maximum lateral resolution (distance between two scanlines) is 670 $\mu m$. For the worst case scenario, where all firings contribute towards a voxel, if the tissue motion velocity is less than $670\ \mu m \div 58.4\ ms = 1.15\ cm/s$, we anticipate very little blurring due to motion. However, for larger motion, some form of motion estimation and compensation will be needed as in all synthetic aperture based imaging modalities.

**Effect of Fractional Overlap Between Bands**

Since the transducer bandwidth is fixed, if two frequency bands do not overlap, the bandwidth of each chirp would be small resulting in poor axial resolution. The

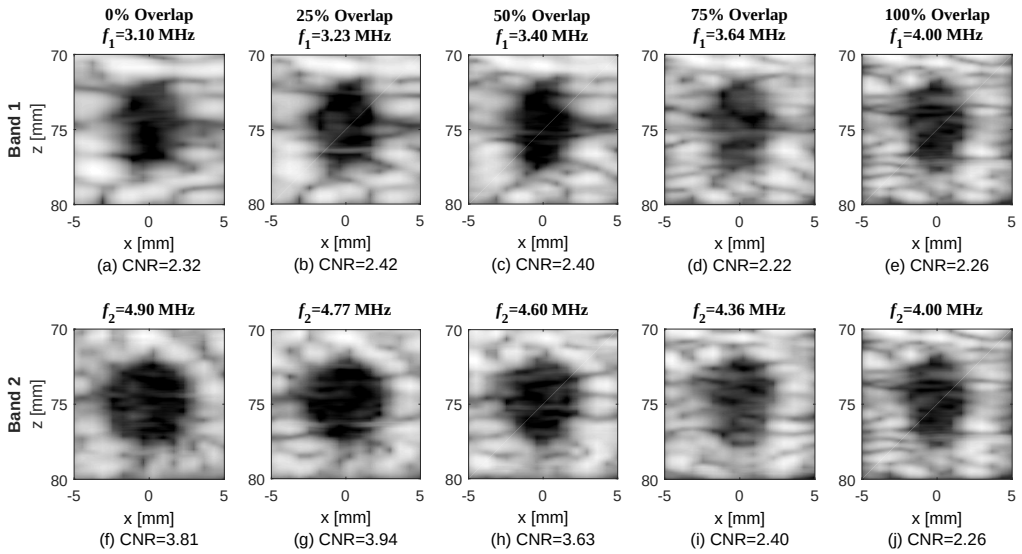



Figure 3.12: Cyst Images Located at 75 *mm* and Generated Using Chirps in Frequency Bands with Different Amount of Overlap.

difference between the center frequencies of two bands would also be large, leading to a large difference in image quality. To demonstrate the effect of the amount of frequency overlap on the image quality, we present simulation results generated using chirps in frequency bands with differing overlap. We define the percent overlap as the ratio of the frequency overlap width and chirp bandwidth B.

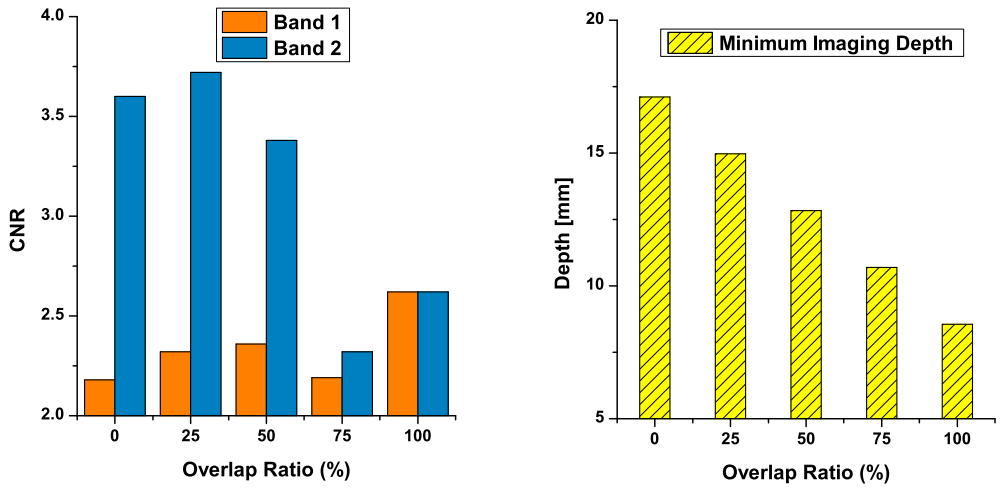
In this set of simulations, the f number is fixed at 3. The spacing of the basic transducer grid is not $\lambda_0 = 335\ \mu m$ but rather $c/f_2$ where $f_2$ is the center frequency of band 2 and may reduce as the amount of overlap increases. The simulation results of cyst images at 75 *mm* are shown in Figure 3.12. We see that images generated using 0%, 25% and 50% frequency overlap have good visual quality and comparable CNR values, while the images generated using 75% and 100% frequency overlap are severely degraded. For frequency overlaps of 0%, 25%, and 50%, the differences in CNR values between images generated using band 1 and band 2 decrease with the increase in

the overlap. This is because the lateral resolution depends on the waveform center frequency, and the difference between center frequencies of two frequency bands, $f_1$ and $f_2$, decreases as the frequency overlap increases. Since the transducer bandwidth is fixed, the bandwidths of the two frequency bands increase when increasing the frequency overlap. Thus, $f_1$ increases, $f_2$ decreases and their difference diminishes.

Figure 3.12 also shows that the imaging artifacts caused by interference increase with the increase in the frequency overlap. This is due to the fact that the cross-correlation between chirps in different frequency bands (CC2) increase with increasing frequency overlap. For MTMR-based firing schemes, the interference between simultaneous firing subapertures depends on two forms of cross-correlations: CC1 and CC2. CC1 depends on TBP, and thus is fixed in this case. When the frequency overlap is 0%, CC2 equals 0 and there is no interference between chirps in different frequency bands. When two frequency bands fully overlap, CC2 equals AC, and the interference is maximized.

The minimum imaging depth is another important parameter which decreases when increasing the frequency overlap. For a fixed TBP, the increase in frequency overlap results in larger chirp bandwidth, which leads to smaller waveform time duration. A smaller waveform time duration allows transducer elements to receive signals echoed from shallower regions, thereby achieving smaller minimum imaging depth.

Figure 3.13 summarizes values of CNR and the corresponding minimum imaging depths in terms of frequency overlap. The image quality is comparable for frequency overlaps of 0%, 25%, and 50%, but drops significantly when the frequency overlap is greater than 50%. Since 50% frequency overlap achieves lower minimum imaging depth compared to 0% and 25%, we choose 50% frequency overlap to be the setting for the simulations using MTMR-based firing schemes. We also run simulations with




(a) CNR (b) Minimum Imaging Depth

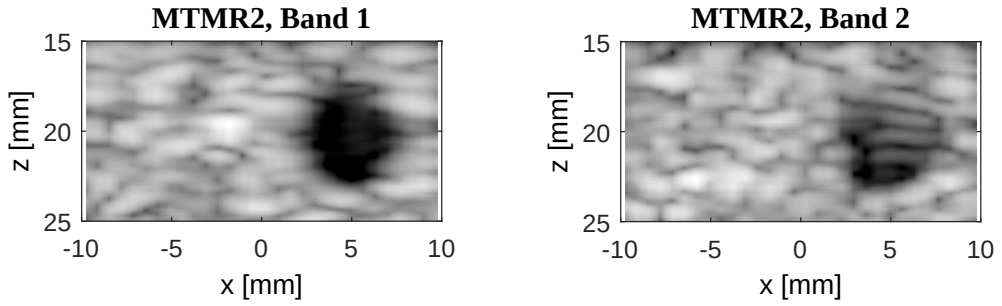
Figure 3.13: Bar Chart of CNR and Minimum Imaging Depth as A Function of the Amount of Overlap Between the Two Frequency Bands.

different scatter distributions for 50% overlap. We find that the standard deviation for band 1 is 0.13, and for band 2 is 0.10.

### MTMR with Sparse Array

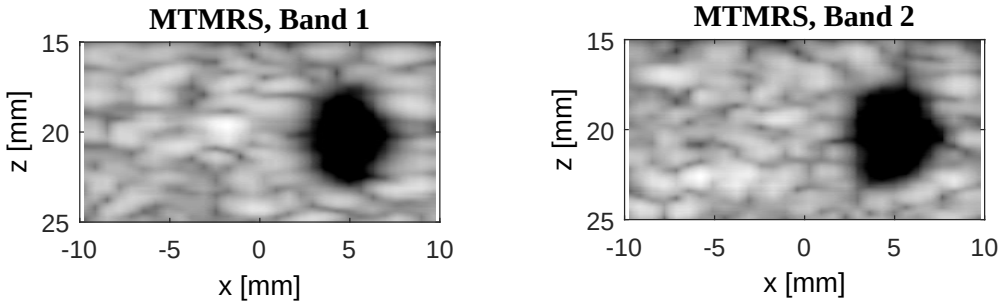
To demonstrate that the proposed sparse array removes the grating lobes, we compare the quality of the cyst image generated using firing schemes MTMRS and MTMR2. The cyst is located at 20 *mm*. To better display the imaging artifacts caused by grating lobes, the cyst in this simulation is shifted off the central axis (at 5 *mm* in X axis), in the corner of the imaging region.

Figure 3.14 shows the cyst images generated using chirps in both frequency bands for both firing schemes. The cyst images generated using MTMR2 + chirps in band 1 as well as band 2 have artifacts inside the cyst. The image generated using MTMR2 + chirps in band 2 is degraded due to grating lobes, while the cyst image generated using MTMRS + chirps in band 2 has good quality. The cyst image generated using




(a) CNR = 2.76

(b) CNR = 1.54



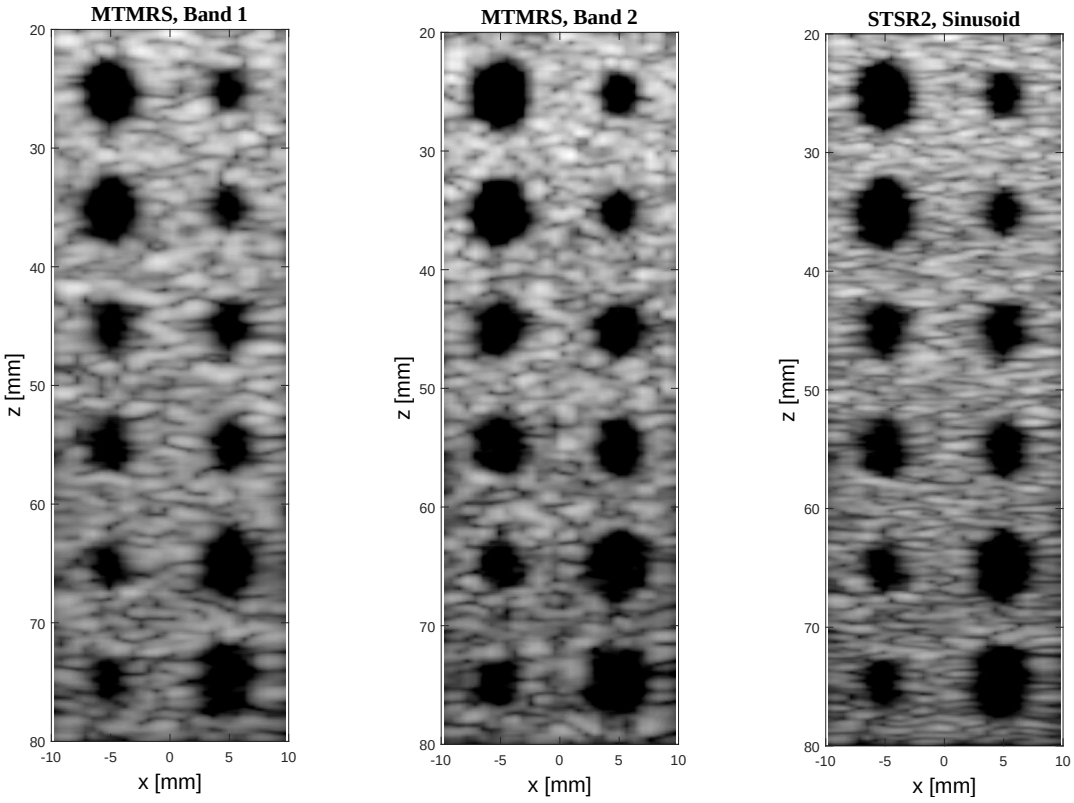

(c) CNR = 2.97

(d) CNR = 3.74

Figure 3.14: Imaging Quality Comparison Between Cyst Images Located at 20 *mm* Generated by MTMR2 and MTMRS.

MTMR2 + chirps in band 1 has less grating lobes compared to the one generated using MTMR2 + chirps in band 2. But the image quality is still lower than the one generated using MTMRS + chirps in band 1. The CNR values confirm that MTMRS achieves better image quality compared to MTMR2.

In MTMR2, as the active transducer elements have $2\lambda_0$ spacing, the grating lobes exist in vicinities of $30^o$ elevational angle (Equation 3.12), which fall in the shallow region. This is confirmed in Figure 3.14 (b), which shows that the grating lobes exist in the cyst image generated using MTMR2 + band 2. In Figure 3.14 (a), since $f_1$ is smaller than $f_2$, the grating lobes in the cyst image generated using MTMR2 + band 1 are reduced, though they still exist.




(a) Average CNR = 2.35 (b) Average CNR = 3.16 (c) Average CNR = 2.85

Figure 3.15: Comparison of Cyst Images Generated using MTMRS and STSR2.

In MTMRS, the sparse array layout is based on the transducer grid with spacing $\lambda_0$. As the active elements are distributed in a non-uniform manner, the signals are not summed coherently in the lateral directions of other simultaneously firing subapertures, and thus the grating lobe levels are reduced. Beam patterns in Figure 3.8 show that grating lobe and sidelobe levels of the sparse array are lower than 40 dB. This is also confirmed in Figure 3.14 (c), (d) which show good image quality of cysts generated using chirps in both bands.

**MTMRS vs STSR2**

To show that the proposed MTMRS method achieves image quality comparable to the STSR2 firing scheme for a wide range of depths, we compare the quality of cyst images generated using MTMRS with chirp excitations and the cyst image generated using STSR2 with the sinusoidal excitation for depths ranging from 20 *mm* to 80 *mm*.

Figure 3.15 shows that both MTMRS and STSR2 firing schemes achieve good image quality. The image quality generated by STSR2 is better than that generated by MTMRS + chirps in band 1, but is worse than for the image generated by MTMRS + chirps in band 2. This trend follows the pattern in Figure 3.9 where the image quality generated using the sinusoidal excitation is between the image quality generated using chirps in band 1 and band 2.

## 3.4 Summary

In this chapter, we proposed a 3-D extension of SASB. To reduce the computational complexity, we reduced the number of active receive elements in the first beamforming stage and applied separable beamforming in the second stage. Since 3-D SASB has low volume rate due to physical constraint of round-trip propagation time of sound, we proposed MTMR firing scheme to increase the volume rate by 4×. To reduce the interference between signals transmitted by different subaperture simultaneously, we used linear chirps, instead of sinusoid, as the excitation waveform, and conducted matched filtering after the signals are received. However, since linear chirps in different frequency bands result in different brightness in the image, we proposed an overlapped firing scheme where the reconstruction of two volumes are overlapped, so that each imaging volume is generated by chirps in the same frequency band. We also designed a sparse array to avoid the grating lobes caused by large

space between active receive elements and the MTMR firing scheme. Compared to the uniformly distributed array in the STSR firing scheme, the sparse array has the same number of active receive elements, and so the number of computations are the same.

We evaluated the proposed techniques using cyst images. In the STSR firing scheme, simulation results show that the reduction in the number of active receive elements from 32×32 to 16×16 does not degrade the imaging quality. In the MTMR firing scheme, while both the linear chirps and sinusoid excitations generate good imaging quality in the shallow region, linear chirps perform much better in the deep region compared to sinusoid. The proposed sparse array design successfully removed the grating lobes that existed in the images generated using uniformly distributed array. Compared to the STSR firing scheme, the MTMR firing scheme generates comparable imaging quality while increasing the volume rate by 4×.

# Chapter 4

# FRONT-END ARCHITECTURE DESIGN FOR 3-D SASB

## 4.1 Overview

In this chapter, we describe the hardware design of the front-end of a 3-D SASB unit. We focus on efficient implementation of Stage 1 computation. As a first step, we investigate techniques to reduce the number of computations in Stage 1, and design an architecture that accelerates the data processing for this stage in the transducer head. We propose a Sum-before-Multiply scheme that sums up the data corresponding to the same apodization coefficient and then multiplies the sum with the coefficient. We also reduce the number of distinct apodization coefficients by clustering the coefficients that have similar values, so that the number of multiplications can be further reduced.

To support the proposed Sum-before-Multiply scheme, we propose a highly parallel architecture. In this architecture, signals received by 961 active receive elements are digitized by 961 ADCs in parallel. The digitized samples are routed to their corresponding digital processing channels through a Network-on-Chip (NoC). Then, the routed samples are delayed and interpolated by 961 processing channels in parallel and the interpolated samples are summed up through a bus-based structure that traverses through all 961 processing channels.

We synthesize the proposed Stage 1 architecture in the Taiwan Semiconductor Manufacturing Company (TSMC) *28 nm* technology node. Synthesis results show that the power consumption of the data path in the proposed architecture is around 1 W. We simulate NoC using *BookSim* [27], which estimates the power consumption

to be 66 $mW$. We also estimate the power consumption of transducers and ADCs based on existing work. The power consumption of the overall system is below 2 W.

### *4.1.1 Related Work*

Existing SASB imaging architectures focus on 2-D imaging. Two wireless probes were designed in [70] and [71], where the first beamforming stage is implemented by analog circuits. After the first beamforming stage, the signals are digitized and sent to a separate mobile device. The second stage is computed by software on the mobile device. This architecture well demonstrated the benefit of SASB in allowing the data after the first beamforming stage to be easily transferred. There have been several papers on accelerating the second beamforming stage computation on different platforms. These include general-purpose graphic processing unit (GPGPU) computing with OpenGL [72], and central processing unit (CPU) with Single Instruction Multiple Data (SIMD) [73]. Another extension of SASB proposed to replace DAS with F-k domain beamforming in the second stage, so that FFT accelerators can be applied to increase the processing rate [16].

### *4.1.2 Design Challenges*

The main challenges for 3-D imaging architectures are the large size of external storage, computation of a very large number of delay values, and the large number of digital processing channels. As SASB applies fixed delay values to all channels in the first beamforming stage, the set of delay values do not change for each transmit and receive event. Specifically, if each processing channel processes data received by transducer element located in the same relative position within the subaperture, the delay values do not change, and thus there is no need for delay storage or online calculation.

In the first beamforming stage, the number of computations is dominated by multiplication of the received data with their corresponding apodization coefficients. Having a multiplier in each of the 961 processing channels results in large area and power consumption. On the other hand, summing up data from 961 processing channels into 4 arrays of outputs (corresponding to the 4 subapertures) results in complicated wiring. In this work, we present methods to reduce the number of multipliers and propose a ring-based architecture to sum the data processed by each channel prior to multiplication.

## 4.2 Hardware-Oriented Complexity Reduction Algorithms

### *4.2.1 Sum-before-Multiply Computation*

In the first beamforming stage, data samples received by each active receive element are multiplied with the apodization coefficient and then summed up to form one output sample. The apodization coefficients are 2-D window functions, for instance, the 2-D Hamming window, where the value of the coefficients decreases as the distance between the receive element and the center of the subaperture increases. Thus the data samples received by transducer elements that have the same distance to the subaperture center are multiplied with the same apodization coefficient.

To avoid repeated multiplication with the same coefficient, the data samples received by transducer elements that have the same distance to the center can be summed up before the multiplication. We define the received signals that share the same apodization coefficient as a *group*, and the sum of these signals as *group-sum*. After summing all received signals in a group, the group-sum is multiplied with a distinct apodization coefficient to generate a scaled group sum. These scaled group-sums are then summed up to form an output sample. This process can be represented

as:

$$F_1(t) = \sum_{u=1}^{U} A_u \cdot g_u(t) \tag{4.1}$$

$$g_u(t) = \sum_{i \in G_u} r_i(t - \tau_u) \tag{4.2}$$

where $A_u$ is the value of the distinct apodization coefficient, $U$ is the number of distinct coefficients, $g_u$ is the group-sum, and $G_u$ is the subset of transducer elements corresponding to coefficient $A_u$. Compared to the MAC-based computations, the number of additions remains the same, but the number of multiplications is largely reduced.

For the 2-D transducer array with 32×32 elements, there are 136 distinct apodization coefficients. With the Sum-before-Multiply scheme, the number of multiplications is reduced from 256 to 136, without reducing the accuracy of computation. To further reduce the number of multiplications, we propose to reduce $U$, the number of distinct apodization coefficients.

### *4.2.2 Fewer Apodization Coefficients*

The apodization process scales the received data with weighting factors or coefficients. Its intent is to reduce the sidelobe level in the beamformed signal, which helps remove the imaging artifacts in the reconstructed image. With the Sum-before-Multiply scheme, the number of multiplications depends on the number of distinct apodization coefficients. So our goal is to reduce the number of distinct apodization coefficients which is equivalent to reducing the number of groups. If the number of groups is large, the apodization is close to the original 2-D Hamming window, the sidelobe level is low, but the number of computations is large. In the other extreme, if the number of groups is 1, the apodization is a plain rectangular window. The sidelobe level is then high, but there is only one multiplication.

We design a new set of apodization coefficients by fixing the number of groups and choosing the coefficients such that the mean-square-error (MSE) between the new coefficients and the original window function is minimized. Let $\tilde{A}(s_x, s_y)$ denote the original apodization coefficients for transducer element $(s_x, s_y)$, and let $A_U(s_x, s_y)$ denote the new set of apodization coefficients, where $U$ is the number of distinct values. The objective function is given by:

$$\min \sum_{s_x=1}^{32} \sum_{s_y=1}^{32} ||A_U(s_x, s_y) - \tilde{A}(s_x, s_y)||^2 \tag{4.3}$$

where $||\cdot||^2$ stands for the square of the absolute difference.

We solve this optimization problem using *K-means Clustering* [74]. The original window function is chosen to be the 2-D Hamming window. Simulation results show that the simplified apodization coefficients scheme generates good imaging quality if $U$ is larger than 16, compared to that generated by using original apodization coefficients.

A combination of Sum-before-Multiply scheme and simplified apodization scheme reduces the number of multiplications dramatically. We summarize the number of multiplications in the original MAC-based computations, and the Sum-before-Multiply scheme with different values of $U$ in Figure 4.1. Compared to the original MAC-based computations, the Sum-before-Multiply scheme with $U = 16$ reduces the number of multiplications by 8.5×.

### *4.2.3 Reduced Precision Arithmetic*

We investigate narrow-bit-width fixed-point arithmetic to further reduce the storage and the bandwidth requirements. To determine the lowest bit-width that generates similar imaging quality compared to the floating-point data, we explore different data precisions for both data-path and ADC in the first beamforming stage

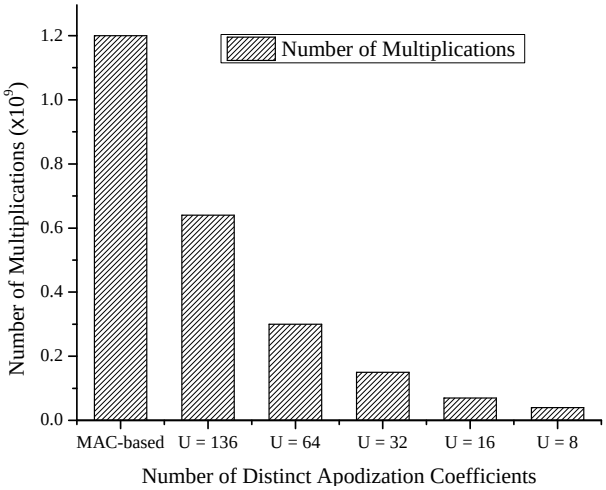


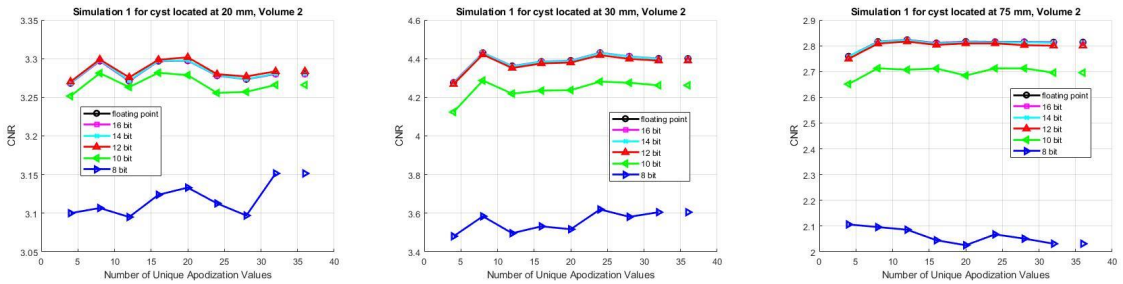

Figure 4.1: Number of Multiplications Comparison for MAC-based Computations.

of SASB. Simulation results presented in Section 4.2.4 show that the combination of 8-bit ADC precision and 10-bit arithmetic precision generates images with little degradation compared with double-precision floating point.

### *4.2.4 Simulation Results*

We simulate a complete SASB system in Matlab. Here, the first beamforming stage implements the proposed simplified apodization method with reduced data precision settings while the second beamforming stage is implemented in double precision floating point. The ultrasound radio-frequency (RF) data is simulated using *Field-II* platform, an ultrasound simulation platform [19, 20]. The configuration of the 3-D ultrasound imaging system is shown in Table 3.2.

Next, we describe the experiments that were used to determine the data-path and ADC precision. For each set of simulations, we compare the CNR of cysts located at 20 *mm*, 30 *mm*, and 75 *mm* to represent the shallow region, the region near VE, and the deep region, respectively.




(a) Cyst located at 20 *mm*. (b) Cyst located at 30 *mm*. (c) Cyst located at 75 *mm*.

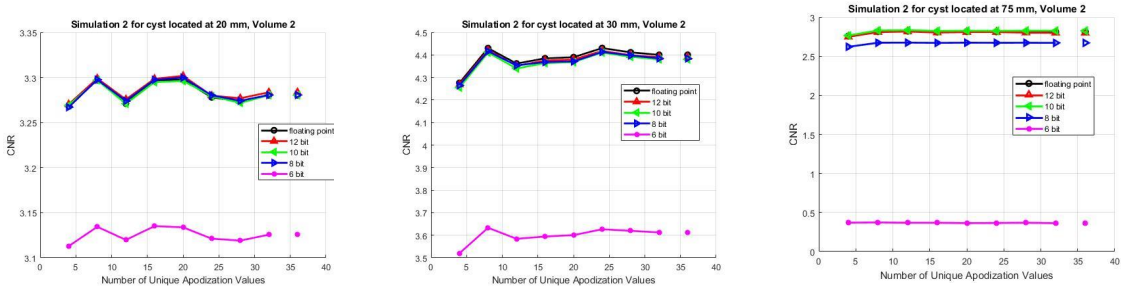
Figure 4.2: CNR of Cyst Images Generated Using Different Data-path Precision Bit-width and Different Number of Unique Values in the Apodization Coefficients.

### Data-path Precision

To determine the minimum data-path bit-width for the first beamforming stage, we compute the first beamforming stage with precision ranging from 8 to 16 bits and compare it with that of double precision floating point data. Note that this is the precision used for the inputs of interpolation and summation. To avoid overflow, we use 4 additional bits for the output of summing operations and multiplication operands. Next, we investigate the imaging quality for different number of unique apodization coefficient values, or $U$. We vary $U$ from 4 to 32 for each data-path precision setting. We also beamform using original Hamming window for comparison. In this set of simulations, the ADC precision is the same as that of the data-path precision.

Figure 4.2 shows the CNR values for different data-path bit-widths and different number of apodization coefficients. In the three different regions, the imaging quality corresponding to each data-path bit-width varies mildly when the number of distinct apodization coefficients $U$ is larger than 8. So we pick $U = 8$.

Figure 4.2 also shows that the CNR curves of cysts generated using fixed-point (12 bits, 14 bits, and 16 bits) and double-precision floating point data are comparable.




(a) Cyst located at 20 *mm*. (b) Cyst located at 30 *mm*. (c) Cyst located at 75 *mm*.

Figure 4.3: CNR of Cyst Images Generated Using Different ADC Precision Bit-width and Different Number of Unique Values in the Apodization Coefficients.

The CNR of cysts generated using 8-bit data-path precision is significantly lower than that of the others. The CNR of cysts generated using 10 bits is 0.1 to 0.15 lower in the different regions. Based on these results, we conclude that 12 bits is the lowest possible data-path bitwidth for good image quality.

### ADC Precision

As the area and power of ADC grow exponentially with precision, our next goal is to determine the lowest ADC precision that generates imaging quality without degradation. We fix the data-path precision to 12 bits and compute the first beamforming stage based on ADC (input) bit-widths ranging from 6 bits to 12 bits.

Figure 4.3 presents the CNR generated using different ADC bit-widths as a function of $U$. The curve demonstrates that 10-bit ADC precision generates the same imaging quality compared to 12-bit ADC. 8-bit ADC precision generates imaging quality comparable to 12-bit ADC in shallow region and the region near VE, but does a little worse in the deep region. 6-bit ADC precision results in significant lower CNR and is not acceptable. So we choose the ADC precision to be 8 bits.

## 4.3 Hardware Architecture Design

Next, we describe a highly parallel Application-Specific Integrated Circuit (ASIC) design for the front end of a chirp-based high-volume 3-D SASB ultrasound imaging device. Figure 4.4 gives a system level view of the proposed 3-D die stacked architecture. Here, the transducers are housed in the top layer, the ADC and the network-on-chip (NoC) are housed in the middle layer, and the digital processing channels are housed in the bottom layer. The computing engine consists of 961 processing channels working in parallel. Data from 8100 transducer elements are fed to these channels selectively.

We propose an architecture to support the Sum-before-Multiply scheme. In this scheme, we sum up the samples corresponding to the same apodization coefficient. Directly summing using a tree of adders results in large area and power consumption. To simplify the design, we propose a bus-based architecture where an input sample is added to the group-sum as it propagates from channel to channel through the bus. The group-sum is computed through the 961-stage pipeline.

### *4.3.1 System Overview*

The 2-D transducer array consists of $90 \times 90$ transducer elements with spacing of 335 $\mu m$, which corresponds to the center frequency of band 2. We wire the $90 \times 90$ transducer elements to the 961 analog-to-digital converters (ADC) through the analog multiplexers. As the proposed sparse array is based on the bin-based random array, only one transducer element is active in each bin of $2 \times 2$ elements in every round. Thus, four transducer elements in each bin can be multiplexed to share one processing channel during the receive process, simplifying the connection from 8100-to-961 to 2025-to-961.

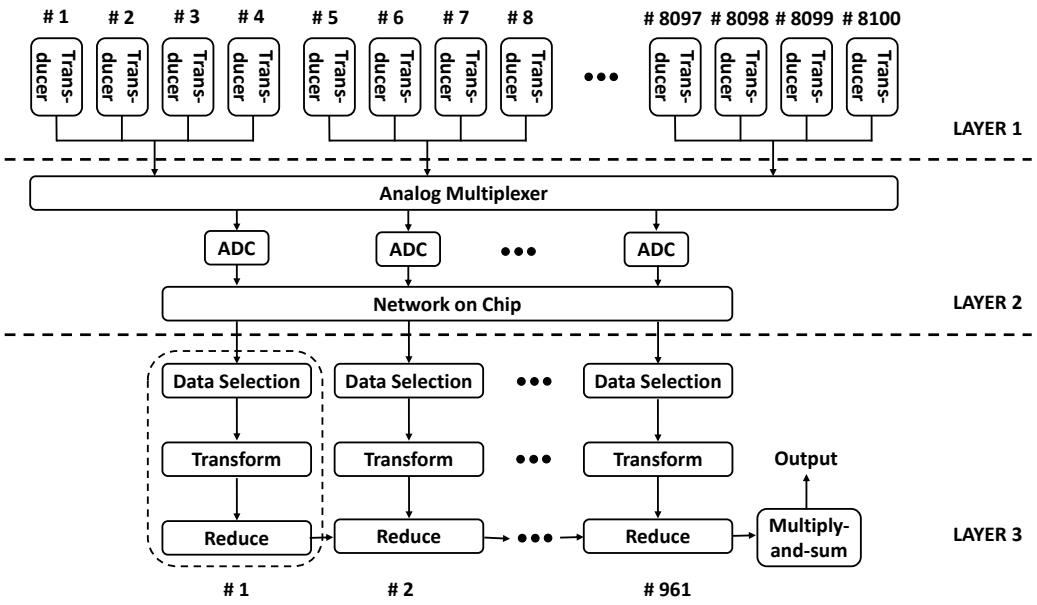



Figure 4.4: 3-D Die Stacking Overview of the Proposed Architecture Consisting of Transducers in Layer 1, ADC and NoC in Layer 2, and 961 Digital Processing Channels in Layer 3.

Of the 2025 bins of transducer elements, only 961 bins contain active transducer elements. Bins that do not include active elements in the same transmit and receive event can share a digital processing channel. This can be implemented by connecting together the transducers which can be shared using wired OR connections. The number of transducer elements wired together can be 2 or 4, depending on the locations of the transducer elements. The bins in the central region of the 2-D transducer array are active in all 225 transmit and receive events, and these can be directly wired to their assigned ADC.

Owing to the subapertures shifting after each round, each transducer element on the 2-D array corresponds to different locations within the subaperture in different firings. To avoid changing the computations in a digital processing channel, we propose to fix the computation in each digital processing channel and then utilize

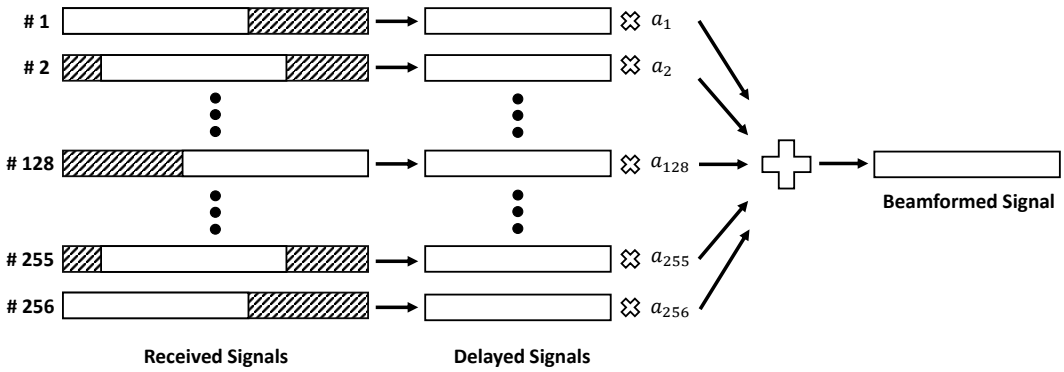



Figure 4.5: Overview of the Operations in the Front-end for One Subaperture.

a NoC to route the 961 digitized samples to the 961 digital processing channels based on the the location of the receive transducer element in the subaperture.

The 961 processing channels work in parallel. The accurately delayed data samples are summed up through a 961-stage pipeline. In each transmit and receive event, the signals received by 961 active transducer elements are beamformed to generate four arrays corresponding to four subapertures. A cartoon figure of the computations in the front end for one subaperture, where signals come from 256 transducer elements, is shown in Figure 4.5. Different segments of data are appropriately chosen from the interpolated received signals, so that the wavefronts in all channels are aligned. The aligned data are multiplied with apodization coefficients, and then summed up to generate one array of outputs.

Each processing channel is formed by one data selection unit to delay and temporarily store the incoming data, one transform unit to interpolate the received data, and one reduce unit to add data to its corresponding group-sum. In addition to

the 961 processing channels, there is a Multiply-and-Sum unit to multiply the group-sums with the corresponding apodization coefficients and then sum up the partial products to form the final output.

**Delay Operation**:A fixed delay is applied to the signal received by each channel for alignment. The delay value is determined by the distance between the VE and the corresponding transducer element. In DAS beamforming, to delay samples with a fine granularity, the data should be sampled at 4-10×of the Nyquist sampling rate [75]. To achieve such a fine delay, we implement the delay operation in a hierarchical fashion. The data selection unit performs a coarse delay operation by dropping the initial samples as shown in Figure 4.5. The coarse delay value corresponds to a 40 $MHz$ clock. Then the transform unit, which operates at 160 $MHz$, performs a fine delay operation by selecting one of the four interpolated samples. Thus a combination of coarse and fine delay operations helps derive data delayed at a fine granularity.

### *4.3.2 Analog-to-Digital Converter*

The analog signals received by the transducers are routed through the analog multiplexer to the corresponding ADCs in parallel, as in [9]. The ADC digitizes the received signals into 8-bit fixed point data and then stores the data into the input buffer of NoC. To achieve high imaging quality, the sampling frequency of ADC should be 4-10 times higher than the Nyquist frequency of the waveform [75, 76]. In this work, we set the sampling frequency of ADC to be 40 $MHz$.

### *4.3.3 Network-on-Chip*

A straightforward implementation involves connecting the ADCs and the digital processing channels with a fixed connection pattern. When the subaperture shifts, the signals streamed into a channel are delayed differently depending on the relative

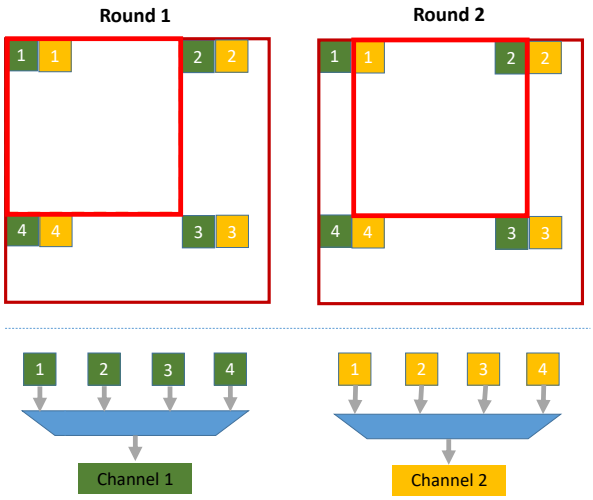



Figure 4.6: Example of Transducer Elements Multiplexing for the First Two Digital Processing Channels

location of the transducer elements in the subaperture. Figure 4.6 shows an example for the first two channels. Here, the four green transducer elements are multiplexed into digital processing channel 1 and the four yellow transducer elements are multiplexed into channel 2. Suppose, the location of a transducer element within the subaperture is $(x, y)$ where $x$ is the horizontal index and $y$ is the vertical index. In firing round 1, green element no. 1 and yellow element no. 1 are active, thus channel 1 and channel 2 perform the operations for transducer elements $(0, 0)$ and $(0, 1)$, respectively. In firing round 2, yellow element no. 1 and green element no. 2 are active, thus channel 2 and channel 1 perform operations for transducer elements $(0, 0)$ and $(0, 31)$, respectively. Since the channels perform different operations depending on the relative locations within the subaperture, this connection results in large storage size for the control signals. An alternative design is to increase the analog multiplexer size so that each transducer element is wired to the corresponding processing channel depending on its location within the subaperture. The corresponding design results in 961 analog multiplexers with size of 225-to-1, which is impractical.

We propose to use a NoC[1], which routes the digital sample from the source ADC to the destination channel through a regular network structure. Compared to the point-to-point (P2P) connections, NoC has smaller area and power overhead [77]. Since NoCs allow concurrent transactions, the end-to-end latency of an NoC is significantly lower than P2P connections. In this NoC, at the source side, the digital sample from the ADC is enclosed in a packet through a network interface, stored in the input buffer, and then waits for the router to serve. Each router makes a routing decision and allocates a channel. The packet is then routed to the next router on its path. The processes repeat until the packet arrives its destination [24].

The routing path of each packet depends on the routing algorithm. Congestion occurs when several packets request for the same path. Usually, the router arbitrates the conflict using a round-robin fashion. Nevertheless congestion increases the latency, which increases the data processing time of each firing, thus affecting the volume rate of the system.

One solution to reduce the latency due to congestion is to increase the number of virtual channels, which are parallel FIFOs that are present in the input and output of each router. Use of virtual channels reduces *Head of Line* blocking, which further reduces the end-to-end latency. However, the virtual channels also increase the area and power consumption of the NoC. Thus the choice of number of virtual channels is a balance between the volume rate and the system's power and area overhead.

We use a mesh NoC due to its scalability and regular structure. We adopt XY routing algorithm for the NoC. In XY routing algorithm, a router has connection to its four neighbor routers in the horizontal and vertical directions. As it is similar to the physical layout, the coordinates of the routers can be easily defined as x–y coordinates, thus the design of local routers is simple and has low power and area

[1]The NoC was designed with the supervision of *Sumit K. Mandal*, ASU.

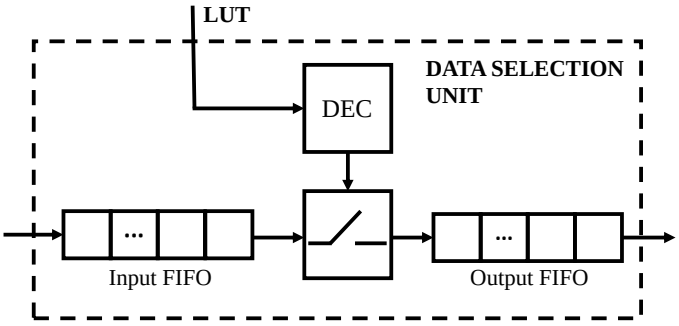



Figure 4.7: Data Selection Unit.

overhead [78]. Since the numbers of signals that have to be routed every time is 961 = 31×31, our proposed mesh size is 31×31. At each node of this NoC, the input is the sample digitized by ADCs and the output is the digital processing channel. Table 4.2 shows different design parameters of the proposed NoC.

The routing algorithm decides the path of each packet. Since ultrasound imaging has strong requirements on timing, and the connection pattern of all 225 firing schemes are fixed, we choose a deterministic routing algorithm. Compared to the adaptive routing algorithm which chooses paths based on the current congestion, the deterministic routing algorithm guarantees minimum latency for each connection pattern, so that the size of virtual channels and the buffer size can be customized accordingly.

### *4.3.4 Data-Selection Unit*

The digital samples routed by NoC are fed into the data-selection unit. The data-selection unit applies the delay to the received signals and aligns the received signals for the multiplication and summation operations. In the architecture designed for SAU [9], the received samples are stored in on-chip SRAM memories. The received signals have to be stored in memory because the same signal is used to compute data

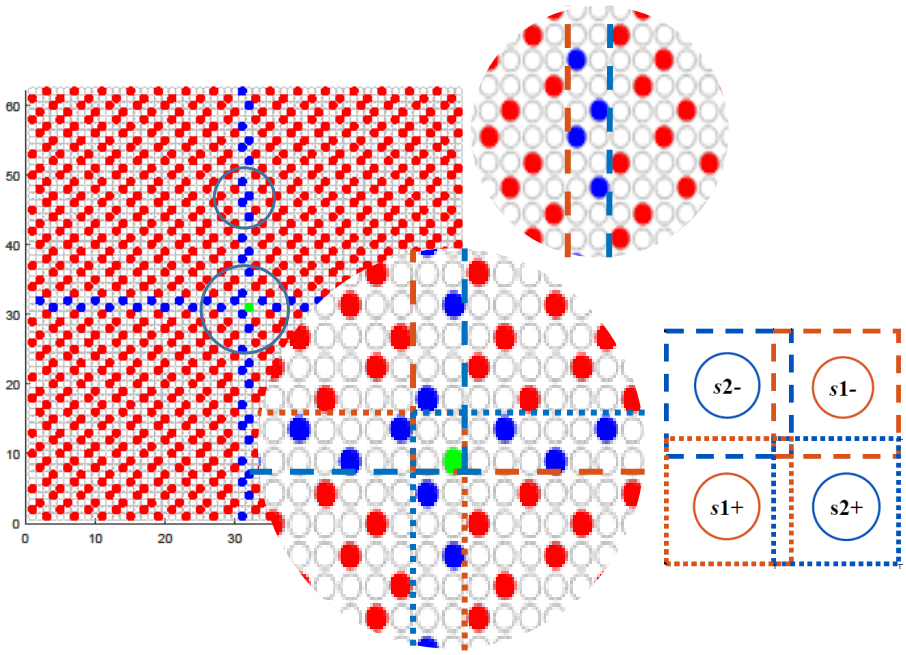



Figure 4.8: Relative Locations of the Overlapped Transducer Elements in Different Subapertures.

in multiple scanlines. In SASB, each subaperture generates only one array of RF-line in the first stage. The received signals are applied with fixed delay and each sample in the subaperture is accessed only once. Thus, it is sufficient to store the received samples in a first in, first out (FIFO) buffer, and pop the data out sequentially for the interpolation operations.

We implement the delay operations by dropping the initial samples that are never used and passing the rest of samples to the output FIFO. The number of the samples that have to be dropped depend on the distance between the transducer element and VE. By outputting the sample from output FIFO at the proper time, the interpolated sample after the transform unit can be added to the right group sum on the bus. However, in MTMR firing scheme, subapertures that transmit and receive simultaneously share few active receive transducer elements. Consequentially,

the signals received by these active receive transducer elements should be delayed by different amounts since they are beamformed for different subapertures. This also means that the starting index of data from the same channel could be different for different subapertures.

The data-selection unit shown in Figure 4.7 consists of a first-in, first-out (FIFO) buffer at the input, a decrementer, and a FIFO buffer at the output. The decrementer is initialized by a locally stored value, which is the number of initial samples that have to be dropped to align the wavefront (the shaded area shown in Figure 4.5). In each cycle, one sample is popped from the input FIFO and the decrementer reduces its value by 1. If the value of decrementer is zero, the popped sample is passed to the output FIFO. Otherwise, the sample is dropped. Here, as the largest delay value needed for the alignment is 32, we design the size of the input FIFO to be 32 samples. To buffer the data for the 961-stage pipeline in the reduce unit, we need the output FIFO to be 1024 samples. However, if the digital processing channel operates at higher frequency compared to the ADC rate, the size of the output FIFO can be reduced. If the clock frequency of digital processing channel is 1 $GHz$, we can design the output FIFO with 64 samples.

Figure 4.8 shows the active transducer elements corresponding to the four subapertures. The active transducer elements in the region where the two subapertures overlap along the row or column are marked in blue and the region where all four subapertures overlap is marked in green. We find that the difference in delay values applied to 'blue' channels is either 1 or 2 samples, and the largest difference in delay values for the green channel is 3 samples. Specifically, 29 channels have to be delayed by 1 sample, 33 channels have to be delayed by 2 samples, and only one channel has to be delayed by 3 samples.

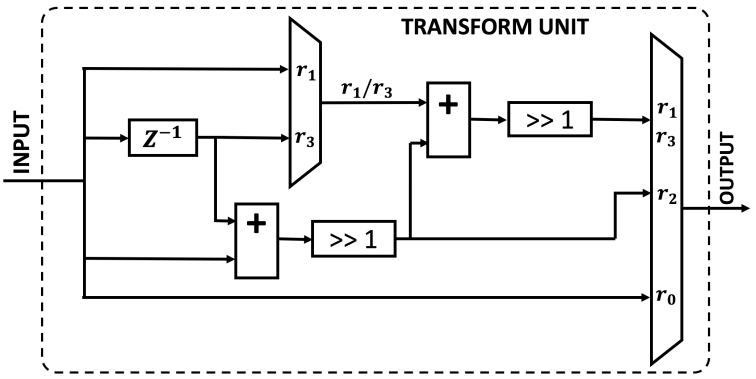



Figure 4.9: Design of the Transform Unit.

To compensate the difference in delay values required by beamforming for different subapertures, additional registers are needed in each digital processing channel. However, while the number of additional registers is quite small, the control overhead of such a scheme is not small. Experimental results show that a simple difference of one or two in the delay values does not incur difference in the image quality. So our approach is to delay the input of all channels by 2 samples to reduce the hardware complexity.

### *4.3.5 Transform Unit*

The transform unit is used to interpolate the data obtained by sampling at 40 $MHz$. The data sampled at 40 $MHz$ is not sufficient to correctly reconstruct the image [28], so the received signals are interpolated to substantially increase the sampling rate. In this work, we propose to increase the data rate of received signals from 40 $MHz$ to 160 $MHz$ through linear interpolation. In this process, three new samples are computed and inserted between any two samples in the received data. The operations

to generate the three new samples are given by

$$\begin{aligned} r_1(t) &= 3/4 \cdot r_0(t) + 1/4 \cdot r_0(t-1) \\ r_2(t) &= 1/2 \cdot r_0(t) + 1/2 \cdot r_0(t-1) \\ r_3(t) &= 1/4 \cdot r_0(t) + 3/4 \cdot r_0(t-1) \end{aligned} \tag{4.4}$$

where $r_1(t)$, $r_2(t)$, and $r_3(t)$ are the interpolated samples, $r_0(t)$ and $r_0(t-1)$ are the two original adjacent data samples in the received signals. Since the output of the first beamforming stage is again down-sampled to 40 $MHz$, only one sample among $r_0$, $r_1$, $r_2$, and $r_3$ is sent out. As the first beamforming stage applies fixed-delay to each received signal, the time-difference between any two consecutive samples in the output array is the same. Thus for one subaperture, only one of the three operations shown in Equation 4.4 is performed to generate the interpolated samples.

The block diagram of the transform unit is shown in Figure 4.9. $r_0(t)$ is directly output. $r_2(t)$ is computed by shifting the sum of input $r_0(t)$ and its delayed value $r_0(t-1)$. To make use of the result of $r_2(t)$, $r_1(t)$ is computed by $r_1(t) = \frac{1}{2}r_2(t) + \frac{1}{2}r_0(t)$, and $r_3(t)$ is computed by $r_3(t) = \frac{1}{2}r_2(t) + \frac{1}{2}r_0(t-1)$. The selection signal is stored locally. For the processing channels that correspond to transducer elements only employed by one subaperture, only one operation is performed in each transmit and receive event. For the processing channels that beamform for different subapertures, the operations change based on the subaperture which the interpolated sample belongs to.

### *4.3.6 Reduce Unit*

In the reduce unit, we add the accurately delayed data sample to its corresponding group-sum. The difficulty of implementing the Multiply-before-Sum scheme is that the data in a group could be located in processing channels that are far from each other. The point-to-point connection between any pair of processing channels results in complicated wiring. In addition, when the subaperture shifts, the correspondence

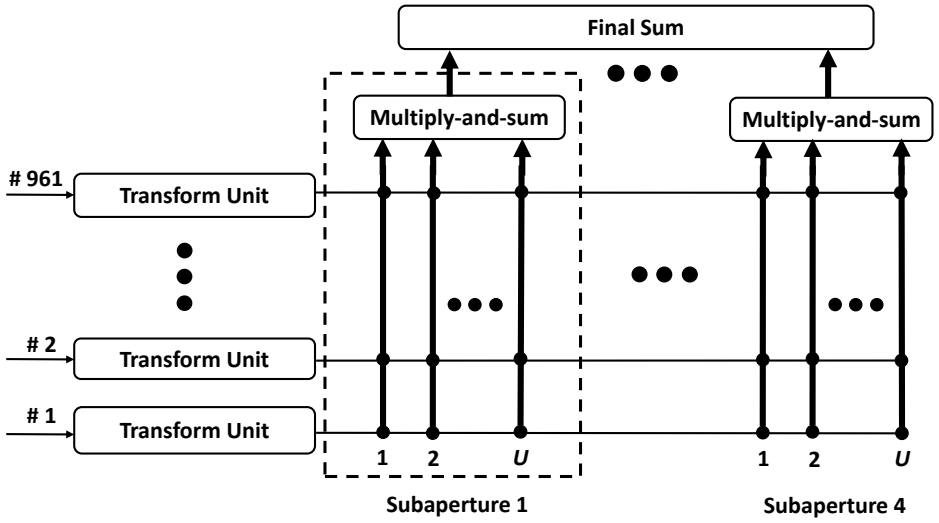



Figure 4.10: Overview of the Bus Structure.

between the processing channel and the relative location of transducer element in the subaperture changes. Thus after each transmit and receive event, the connection between processing channels have to be reconfigured. So our approach is to implement a bus-based architecture which picks up the relevant data and updates the group-sum.

High level design of the bus-based structure is shown in Figure 4.10. We implement the Multiply-before-Sum scheme using several data buses, where each bus carries a group-sum associated with a specific apodization coefficient. Since the number of simultaneously firing subapertures is 4, the number of buses in a straight-forward design is $4U$. Such a large number of buses results in large area and high power consumption. So to keep the number of buses to $U$, we propose to time multiplex the reduce unit and beamform different subapertures in different time cycles. Since the ADC clock frequency is 40 $MHz$, if we time multiplex the reduce unit by 4, its

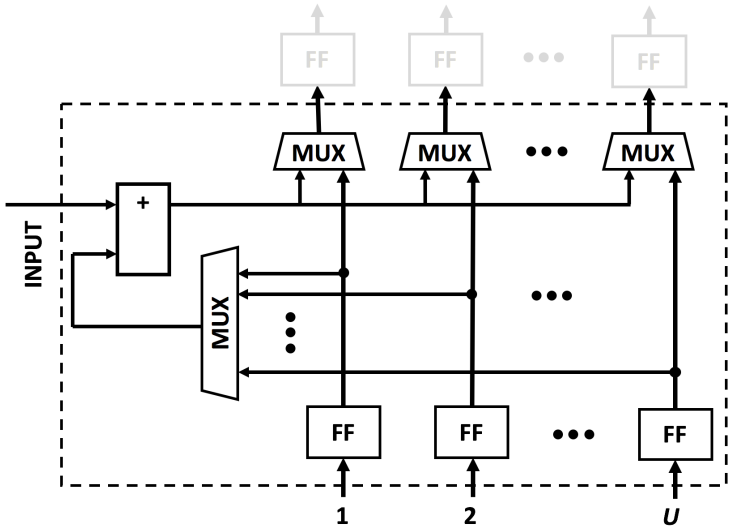



Figure 4.11: Design of the Reduce Unit.

clock frequency is 160 $MHz$. Note that the number of buses can be further reduced if required. For instance, if $U$ = 12 and we clock the reduce unit at 960 $MHz$, the number of required buses becomes 2.

The detailed design of the reduce unit is shown in Figure 4.11. In each cycle, the group-sums are sent from one reduce unit to the next through buses. If the input sample belongs to the same group as one of the group-sums, the multiplexer (MUX) selects the corresponding group-sum. This input sample is then added to the group-sum using the local adder and the updated value is written back to the corresponding bus. If the input sample does not match any of the group-sums in this cycle, all the group-sums are directly sent to the next reduce unit without updating. The MUX and decoders select signals are stored in the local LUT.

### *4.3.7 Multiply-and-Sum Unit and Final Sum Unit*

The final Multiply-and-Sum unit is implemented using several MAC units and a tree of adders, where the number of MAC units is the same as the number of buses. Each MAC unit multiplies the group-sum with the corresponding apodization

Table 4.1: TSMC 28 nm ASIC Synthesis Result for Each Unit

| | Data Selection Unit | Transform Unit | Reduce Unit (U = 2) | Reduce Unit (U = 4) | Multiply-and-sum Unit (U = 2) | Multiply-and-sum Unit (U = 4) |
|---|---|---|---|---|---|---|
| Area | 2447.68 $\mu m^2$ | 134 $\mu m^2$ | 184.212 $\mu m^2$ | 338.94 $\mu m^2$ | 1577.26 $\mu m^2$ | 3174.82 $\mu m^2$ |
| Latency | 0.295 $ns$ | 0.321 $ns$ | 0.336 $ns$ | 0.24 $ns$ | 0.547 $ns$ | 0.555 $ns$ |
| Power | 0.74 $mW$ | 60.2 $\mu W$ | 0.113 $mW$ | 0.204 $mW$ | 0.952 $mW$ | 1.917 $mW$ |

coefficient and accumulates the group-sums. This is multiplied with a 16-bit coefficient to generate a 16-bit output. The accumulated values are then summed up through a tree pf adders to generate the final 16-bit output.

### *4.3.8 Synthesis Results*

**Data Path:** We implement the data path design (data-selection unit, transform unit, reduce unit, and the multiply-and-sum unit) using System Verilog and synthesize using TSMC 28$nm$ technology node. Since the proposed architecture is a streaming architecture, there is no need for on-chip memory. We synthesize the reduce unit with 2 buses and 4 buses. The synthesis data for each unit is presented in Table 4.1. The power numbers are for a clock frequency of 1 GHz and supply voltage of 0.9V. The total power consumption for the digital data processing unit is the power consumption of 961 channels plus the power of the corresponding multiply-and-sum unit. The power consumption is 878.54 $mW$ and 966.95 $mW$ for $U = 2$ and $U = 4$ system, respectively.

The data-selection unit and the multiply-and-sum unit have larger area and power consumption compared to the transform unit and the reduce unit. The data-selection unit buffers the input so it has large number of registers, which result in large area and power consumption. The multiply-and-sum unit includes multipliers for 16-bit data, which also incurs large area and power overhead. Since the number of multipliers is

same as the number of buses, the overhead of $U = 4$ system is twice of $U = 2$ system. The transform unit has two adders and one register, and the reduce unit has one adder and $U$ registers, thus the power and area of these two units are quite small.

The latency of the data selection unit, the transform unit, and the reduce unit are around 0.3 $ns$ while the latency of the multiply-and-sum unit is around 0.55 $ns$. This is to be expected since the multiplier has much higher latency compared to the MUXes and adders in the rest of the units. The highest latency is in the multiply-and-sum unit with $U = 4$. This latency corresponds to a maximum clock frequency of 1.80 $GHz$. In reality, we can reduce the clock frequency to save power consumption.

The majority of power and area overhead is incurred by the data selection units. We plan to investigate techniques that would result in use of smaller sized FIFO queue. Recall that the VE depth can be increased resulting in smaller delay values, and hence smaller sized FIFO queue. But increasing VE depth also increases $f$ number, which leads to poor resolution. So further improvements to this architecture require detailed study of trade-offs between imaging quality and power consumption.

**Transducer:** We use an array of 90×90 transducer elements with spacing of 335 $\mu m$, which is the wavelength corresponding to the center frequency of band 2. The total area of the transducers is 909.02 $mm^2$. Previous work in [9] proposes a beamforming technique with 1024 active transducer elements for receive. The power consumption of transducers is reported to be 0.3 W. In this work, we apply 961 active elements in each firing event, which is 6% lower. Thus we estimate that the power consumption of the transducers is around 0.3 W.

**ADC:** We plan to utilize a 8-bit 750 MS/s SAR ADC in 28-nm CMOS technology [79]. The reported area is 40 $\mu m$ ×100 $\mu m$ and the power consumption is 4.5 $mW$. In our system, we need 961 ADC operating at 40 MS/s with 8 bit precision. So we

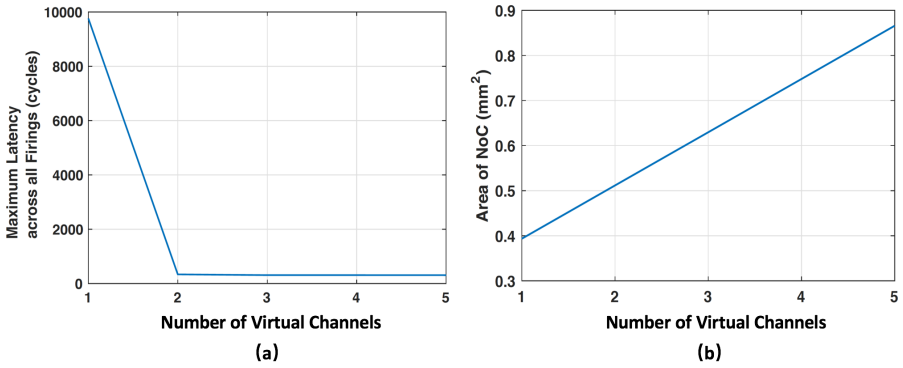



Figure 4.12: (a) Maximum Latency and (b) Area of NoC as A Function of Number of Virtual Channels.

linearly scale down the ADC power in [79] to fit our requirement of 40 MS/s. The area of 961 ADCs is 3.84 $mm^2$ and the power consumption is 0.23 $W$.

**NoC:** In this work, we use a cycle-accurate NoC simulator – *BookSim* [27] to extract NoC performance as *BookSim* provides more flexibility in perofrmance analysis. From *BookSim*, we obtain maximum latency, area, and power consumption for different virtual channel sizes. Based on the trade-off between the latency and power and area overhead, we determine the final configuration.

Using X-Y topology and mesh size of 31×31, we simulate NoC with different numbers of virtual channels. We find the minimum latency that does not affect the volume rate significantly and also does not incur significant power and area overhead either. The procedure is as follows. We summarize the connection patterns in all 225 firing events and represent it in the adjacency matrix format, where each entry is a Boolean variable representing whether there is a connection between the ADC and the digital processing channel. We feed this adjacency matrix into *BookSim* to simulate the maximum latency, area, and power consumption of the input configuration. The operating clock frequency of NoC is 1 GHz and the technology node is 28$nm$ [2].

[2]This simulation is done by *Sumit K. Mandal*.

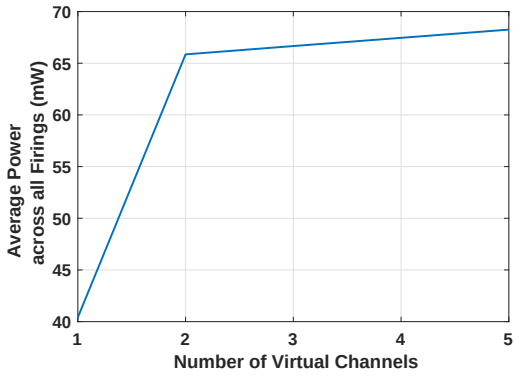



Figure 4.13: Average Power Consumption as A Function of Number of Virtual Channels.

Figure 4.12(a) shows the maximum latency of all 961 channels in 225 firing events as a function of number of virtual channel. The maximum latency is above 9000 cycles when there is only one virtual channel and then drops below 500 cycles when the number of virtual channels is equal or larger than 2. This trend shows that the use of virtual channels effectively avoids the congestion and there is no benefit when the number of virtual channels is larger than 2.

Figure 4.12(b) shows the area of the NoC as a function of the number of virtual channels. The area increases linearly as the number of virtual channels increases. This is because each connection in the NoC needs additional concurrent FIFO queues to accommodate the increase of the number of virtual channels.

Figure 4.13 shows the power consumption of the NoC as a number of virtual channels. The power consumption increases from 41 $mW$ to 66 $mW$ when the number of virtual channels increases from 1 to 2 and then increases slightly as the number of virtual channels continues to increase. Since the maximum latency does not reduce as the virtual channel size is larger than 2, most of the channels are not utilized, so the power consumption does not increase much.

Table 4.2: NoC Configuration and Performance Results

| **Settings** | **Values** |
|---|---|
| Topology | 31×31 Mesh |
| Routing Algorithm | X-Y Deterministic |
| Number of Virtual Channels | 2 |
| Buffer Size | 10 |
| Channel Width | 8 bits |
| Clock Frequency | 1 $GHz$ |
| Area | 0.51 $mm^2$ |
| Power | 66 $mW$ |

So based on the maximum latency, area, and power consumption evaluation, we choose number of virtual channels to be 2 as the best configuration. The final configuration of the NoC is shown in Table 4.2. With 1 GHz clocking frequency and 28$nm$ technology node, the area is 0.51 $mm^2$ and the average power consumption over all 225 firing events is 66 $mW$.

With 2 virtual channels, the maximum delay is 308 cycles which is 308 $ns$. If we include this delay into the time taken by each transmit and receive event, the volume rate is reduced from 34.22 volumes / second to 34.14 volumes / second, which is very minor. Thus the delay due to congestions in NoC causes very little change in the volume rate.

**Total Area and Power Consumption:** The power consumption of the overall system is summarized in Table 4.3. Based the synthesis results of the data path, the simulated power and area of NoC, and the estimates from the existing work, we estimate the power consumption and the area in 28 $nm$ technology node of the overall system to be 1.475 W and 916.03 $mm^2$, respectively. The power consumption of our

Table 4.3: Overall Power Consumption

| Components | Transducer | ADC | NoC | Data Path (U = 2) | Overall System |
|---|---|---|---|---|---|
| Power | 0.3 $W$ | 0.23 $W$ | 0.066 $W$ | 0.879 $W$ | 1.475 $W$ |
| Area | 909.02 $mm^2$ | 3.84 $mm^2$ | 0.51 $mm^2$ | 2.66 $mm^2$ | 916.03 $mm^2$ |

system is quite low compared to the existing works in [43] and [41], due to avoidance of on-chip memory and low computational complexity in beamforming computations.

From Table 4.3, we see that the data path occupies 59.6% of the total power consumption. The NoC has the smallest area and the smallest power consumption. The transducer elements occupy the maximum area. This is because the 90×90 transducer elements are laid out with spacing of 335 $\mu m$. The power consumption values are for 0.9 V supply voltage and clock frequency of 1 GHz. Since the $U = 2$ system can be clocked at 1.8 GHz, we could lower the supply voltage along with the clock frequency to reduce the power consumption of the data path further.

## 4.4 Summary

In this chapter, we proposed a sum-before-multiply computation scheme where the signals corresponding to the same coefficient are summed up before multiplication. Furthermore, we proposed to reduce the number of apodization coefficients by clustering the coefficients into a small number of groups and replace the coefficients in each group with the average value. Field-II simulations show that apodization with 8 distinct values generate cyst images with good quality compared with the original Hamming window, while reducing the number of multiplications by 17×. We also found that the lowest bit-widths that preserve the imaging quality in the data-path and ADC are 12 bits and 8 bits, respectively.

To support the sum-before-multiply computation scheme, we utilized the 3-D die stacking architecture and designed a highly parallel architecture. In this architecture, 961 ADCs digitize the signals received by 961 active elements, a NoC routes the digital samples to their corresponding channels, and 961 digital processing channels delay and interpolate the received signals in parallel. The interpolated samples are summed up through a bus-based structure that traverses through all 961 processing channels. We synthesized the proposed architecture using TSMC *28 nm* technology node and estimate the area and power consumption of the NoC using *BookSim*. Synthesis results show that the power consumption of this architecture is well below 2W.

# Chapter 5

# MAPPING SASB STAGE TWO COMPUTATION TO A MULTI-CORE ARCHITECTURE

After the Stage 1 computations of SASB are done in the front-end, the data is transferred to a separate computing unit for the Stage 2 computations. We propose to implement the Stage 2 computation on a multi-core architecture, TRANSFORMER, that was designed at the University of Michigan. In Stage 2, the data is first passed through a matched filter, to compress the linear chirp and filter the interference generated by other simultaneously firing subapertures. Then a dynamic focus beamforming is performed on the filtered signals to generate the imaging voxels. In this chapter, we describe our plan for mapping matched filtering (Section 5.2) and dynamic focusing (Section 5.3) onto TRANSFORMER.

## 5.1 TRANSFORMER Architecture

TRANSFORMER [1] is a multi-core reconfigurable architecture designed by *Subhankar Pal* and others at the University of Michigan. The architecture consists of multiple tiles, where each tile has multiple cores, or general processing elements (GPE), that are coordinated by a local control processor (LCP). It supports a two-level cache hierarchy, where the caches can be reconfigured at run-time.

Figure 5.1 shows an example of TRANSFORMER with 4 tiles and 16 GPEs per tile. The 16 GPEs are connected with 16 in-tile L-1 memory banks through a high-speed cross-bar. The cross-bar makes it possible for each GPE to access any memory bank. If there is an access conflict between more than 1 GPE, the cross-bar makes the arbitration. The LCP communicates with the GPE through a work queue and a

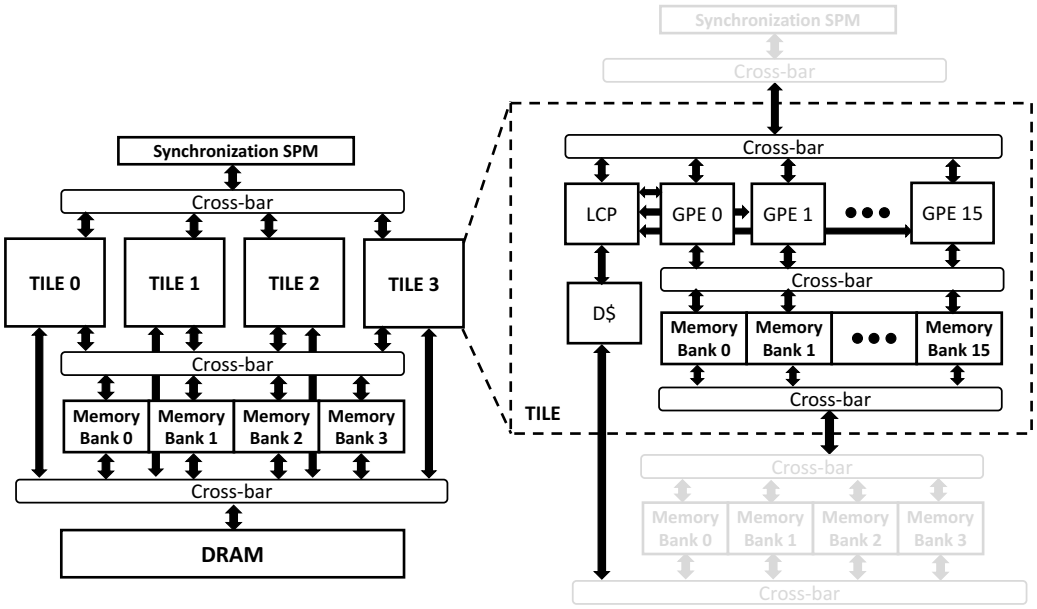



Figure 5.1: Block Diagram Architecture of Multi-core Architecture, TRANSFORMER (Adapted from [1]).

status queue. For example, LCP sends data to GPE through work queue to request start of computation and GPE sends the result back to LCP through the status queue.

In TRANSFORMER, the local memories form a two-level memory hierarchy. The memory banks within the tiles form the L-1 memory and the memory banks shared between tiles form the L-2 memory. The 4 memory banks are connected to a high-bandwidth DRAM memory through a high-speed cross-bar, which is a programmable multiplexer. If more than one memory bank accesses the high-bandwidth memory at the same time, the high-speed cross-bar serves as an arbitrator. There is also a synchronization scratch-pad memory which can be accessed by all processing tiles. Global variables, such as the common coefficient and the synchronization signals, can be stored in the synchronization scratch-pad.

The L-1 memory within each tile and the L-2 memory shared by between tiles can be configured in either scratch-pad, cache, or hybrid mode. In the scratch-pad mode,

the data in the memory are only temporarily stored and read by GPEs while in the cache mode, the data are fetched from the DRAM memory and written back after the cacheline is flushed. In the hybrid mode, half of the memory banks are configured as scratch-pads and the other half are configured as caches.

The L-1 and L-2 memory banks can also operate in the shared mode or private mode. If the cross-bar allows GPEs to access all the memory banks, it is in shared mode. If the cross-bar only allows each GPE to access its allocated memory bank, it is in private mode. In the shared mode, as the memory banks are shared by all the GPEs, the memory size available for one GPE is large. In the private mode, as there is no access conflict, the L-1 access efficiency is higher, but the GPEs can not access L-1 banks of other GPEs.

TRANSFORMER also supports the systolic array mode. As the name implies, this configuration mode allows data to be transmitted between GPEs in a systolic fashion. In this mode, the memory banks are configured as scratch-pad, and one part of the scratch-pad is used as an intermediate queue. The data is passed from one GPE to the next in the following way: GPE 1 pushes the data to GPE 2 by calling '*push*' API, where the data is written into the intermediate queue within its scratch-pad. Then GPE 2 retrieves the data by calling the '*pop*' API, where the data in the queue written by GPE 1 is read out. In this mode, there are two different connections between GPEs and memory banks: one is when GPEs write data to their queues, the other is when GPEs read data from their previous GPE's queues. The high-speed cross-bars keep switching between these two connection patterns to avoid the access conflicts and ensure high efficiency.

## 5.2 Matched Filtering

In the matched filtering step, the signal output from Stage 1 is convolved with the time-reversed version of the waveform, to filter out the interference due to other simultaneously firing subapertures. The waveform is the digitized linear chirp described in Chapter 3. For the sake of simplicity, we use 'signal' to represent the partial beamformed sequence output from Stage 1. The length of the signal is 5195 samples and the length of the waveform is 667 samples. To keep the size of the output signal to 5195 samples, we only compute the middle 5195 convolution outputs. Convolution can be done in time domain (direct convolution) or frequency domain.

**Direct Convolution:** The number of MAC operations to generate one output sample is 667, thus the total number of MAC operations for computations on one subaperture is $3.47\times10^6$. As there are four subapertures transmitting and receiving simultaneously in the MTMR firing scheme, we convolve four arrays of the Stage 1 outputs with corresponding time-reversed waveforms in parallel. If $N$ GPEs are available for matched filtering, the number of GPEs for one subaperture is $N_{sa} = N/4$. To avoid data over-writing, we use the destination-partition scheme where the computation of each output sample is assigned to one single GPE. Thus the number of data samples taken care of by each GPE is $\lceil 5195/N_{sa} \rceil$.

For a TRANSFORMER configuration with 4 tiles and 16 GPEs, each GPE computes 325 output samples. The total number of MAC operations in each GPE is $0.2\times10^6$. We can implement direct convolution in either the hybrid mode, private cache mode, or shared cache mode. As the size of the signal is very large (signal is stored in the DRAM), a configuration without cache results in large DRAM access time, thus it is not considered here.

**Shared / Private Cache Mode:** In the initialization phase, the LCP writes four arrays of signals, one per subaperture, and four waveforms into DRAM. The GPEs read out both the signal and the waveform from the cache, perform the multiplication, and then add it to the partial sum that is locally stored. After the accumulation of 667 partial products is complete, the GPE writes the final sum to the result vector stored in cache and then starts to compute the next output.

In the shared cache mode, each memory bank can be shared by all GPEs, and so the size of data stored in cache is large. However, if large number of GPEs access the same cache line, the access conflicts increase the data access latency. In the private cache mode, each memory bank can be only accessed by its corresponding GPE. The size of data stored by the cache is smaller but the data access latency is small as there are no access conflicts.

**Hybrid Mode:** We configure the L-1 memory in shared cache scratchpad mode, where half of the memory is configured as scratchpad, and L-2 memory in the shared cache mode. In the hybrid mode, we store all the waveforms in the L-1 scratchpad, so that the latency to access the waveform is guaranteed to be small. If single precision floating point arithmetic is used (each data sample has 4 bytes), the memory required to store four waveforms is 10,672 Bytes, which is much smaller than the size of L-1 scratchpad.

In the initialization step, after LCP reads all data into DRAM, GPEs in each tile read the waveform through the L-1 and L-2 cache from DRAM and then write them into the L-1 scratchpad. Since the size of the signal is large, it cannot be housed in the scratchpad memory and hence has to be fetched into the L-1/L-2 cache. The sequence of computations is the same as that of the shared / private cache mode except that the waveforms are fetched from L-1 scratchpad instead of cache.

**FFT:** Convolution can also be computed by taking the Fourier transform of both the input signal and the convolution kernel, multiplying the two frequency-domain sequences, and then taking the inverse Fourier transform of the resulting product. To avoid aliasing, the length of discrete Fourier transform has to be larger or equal to the sum of the two input sequences. Due to the logarithmic complexity of FFT, this implementation has lower computational complexity when the size of input signal is large. For signal size of 5195 and waveform size of 667, the smallest FFT size is 8192. This step includes two FFTs, each of length 8192, one set of 8192 complex number multiplications, and one inverse FFT.

To implement FFT on TRANSFORMER, we configure TRANSFORMER in systolic array mode, where each GPE performs one stage of FFT. For example, for an FFT with size 8192, we use 13 GPEs to compute the FFT using a streaming pipeline fashion. Each GPE receives the partial processed samples from its previous GPE, stores it into the local memory, processes a pair of complex numbers from the local memory, and then sends the results to the next GPE. For the 16 GPEs in a tile, we use 13 GPEs to perform FFT or inverse FFT, 2 GPEs to perform the complex multiplications, and one GPE to write the data.

## 5.3 Dynamic Beamforming

In dynamic receive beamforming, the imaging points at different depths are delayed by different amounts, where the delay values depend on the distance between the imaging points and VE. The delayed samples are multiplied with the corresponding apodization coefficient, and then added to generate the output for each imaging point. Unlike the fixed delay operation in Stage 1, we implement delay operations in Stage 2 by picking up the corresponding sample from the convolved signal using the delay value as index.

In Stage 2, one set of delay constants are shared by all the scanlines. If each VE contributes to the outputs in 19×19 scanlines, there are 55 arrays of constants (owing to the eight-way symmetry), where the size of each array is 5195. This results in storage of $1.14\times10^6$ bytes of delay and apodization coefficient constants. We need to store same number of apodization coefficients. While this size of storage is smaller compared to other beamforming modalities, such as SAU, it is still large for L-1 memory. So we implement the beamforming step using both the shared cache mode and the private cache mode, where both the delay constants and the apodization coefficients are stored in DRAM and read into L-1 caches.

We parallelize the beamforming computation by partitioning the computation of 30×30 scanlines to different GPEs, so that different GPEs write into different locations. If there are $N$ GPEs, each GPE processes $\lceil 900/N \rceil$ scanlines. Each GPE first checks whether the VE corresponding to the firing subpaperture contributes to a scanline that has been assigned to it. Whether the VE contributes or not is determined by both the lateral distance between the scanline and VE and also the vertical distance between the VE and the imaging point. If the input sample contributes to the scanline, the GPE reads out the delay constant and the apodization coefficient. The delay constant is then used as the index to read out the signal sample. The signal sample is multiplied with the apodization coefficient, and then added to output corresponding to the imaging point on the scanline. This computation is repeated until all the scanlines assigned to the GPE are processed.

## 5.4 Simulation Results

### *5.4.1 Simulation Setup*

We simulate different implementation schemes using Gem5 [80]. The basic configuration for TRANSFORMER is 4 tiles with 16 GPEs in each tile. The L-1 memory bank size is 4096 Bytes and the L-2 memory bank size is 65536 Bytes. The DRAM size is 4 GBytes. We vary the number of active tiles and fix the number of GPEs in each tile to 16. For each implementation scheme, we use the execution time and the Giga-operations-per-second/Watt (GOPS/W) as the performance metric. We retrieve the cache miss rates, simulated cycles, GPE idle cycles, and number of memory accesses from the *stats* file generated from Gem5 simulator. Based on the simulated cycles GPE idle cycles, we compute the GPE utilization through

$$GPE\_Utilization = \frac{Simulated\ Cycles - GPE\ Idle\ Cycles}{Simulated\ Cycles}$$

We compute the average GPE utilization by taking the average value of the utilizations of GPEs in all tiles.

The power consumption includes the static power and the dynamic power. We calculate the overall static power by summing up the static power of each components in TRANSFORMER. For the dynamic power, we take the average of dynamic power over the execution time. We estimate the ARM core power consumption by quoting its online specification for both dynamic power and static power, which is based on $40nm$ technology node, and then scaled down to 14 $nm$. For the on-chip memory, the static power and transaction energy per access of the reconfigurable cache banks are modeled using CACTI 7.0 cache model in 14 $nm$ technology node [81]. This power estimation model was designed by *Siying Feng* from University of Michigan.

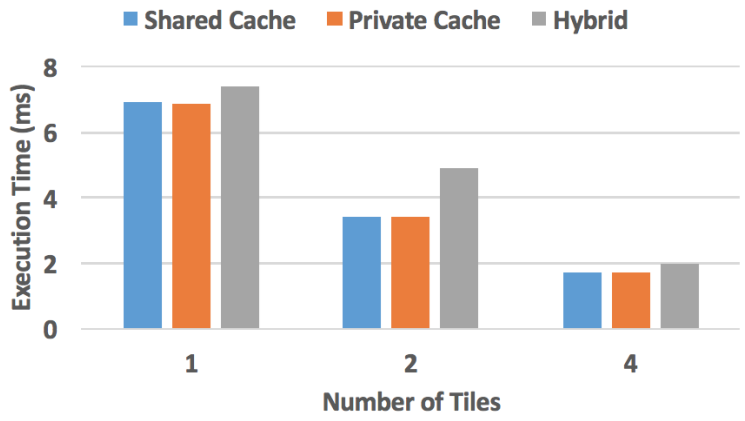



Figure 5.2: Execution Time of Convolution Step Using Shared Cache Mode, Private Cache Mode, and Hybrid Mode

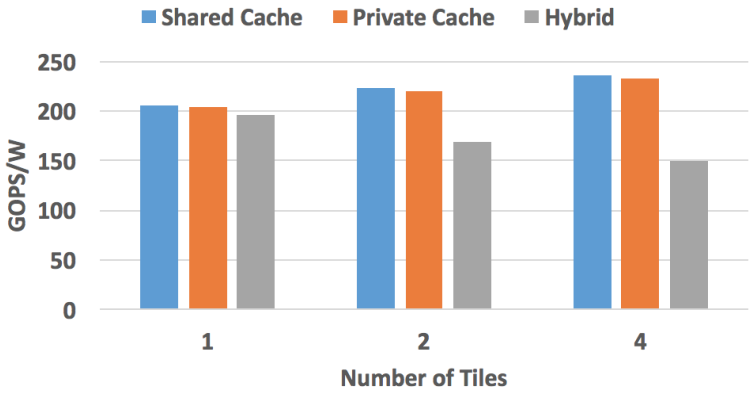



Figure 5.3: GOPS/W of Convolution Step Using Shared Cache Mode, Private Cache Mode, and Hybrid Mode

### *5.4.2 Convolution Results*

We implement the convolution step using shared cache mode, private cache mode, and the hybrid mode. Figure 5.2 and Figure 5.3 show the execution time and GOPS/W for the convolution step, respectively. For all three modes, the execution time reduces as the number of tiles increase, as expected. Figure 5.4 and Figure 5.5 shows the L-1 and L-2 miss rates for the convolution step, respectively. All three

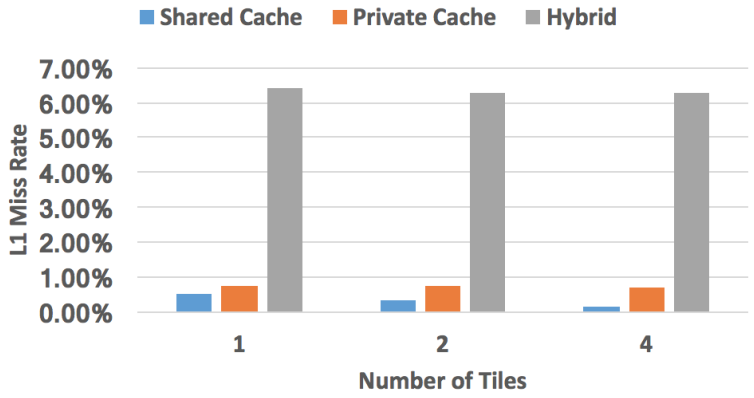



Figure 5.4: L-1 Miss Rates for Convolution Step Using Shared Cache Mode, Private Cache Mode, and Hybrid Mode

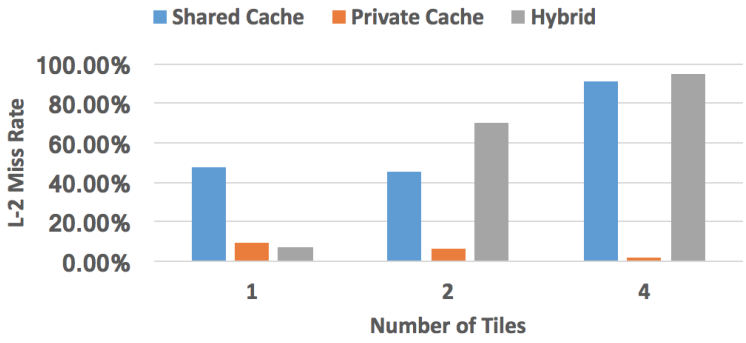



Figure 5.5: L-2 Miss Rates for Convolution Step Using Shared Cache Mode, Private Cache Mode, and Hybrid Mode

modes have high L-2 miss rate and low L-1 miss rate. Compared to the hybrid mode, the shared cache mode and the private cache mode have lower L-1 miss rate, as expected. Compared to the shared cache mode and the hybrid mode, the private cache mode has lower L-2 miss rate when the number of tiles is equal or larger than 2. This is due to the fact that, in the private cache mode, the size of L-1 cache utilized by each GPE is much smaller than the shared mode, which leads to more frequent accesses to L-2 cache. As the L-2 cache size is the same in all three cases, the private cache has lower L-2 miss rate.

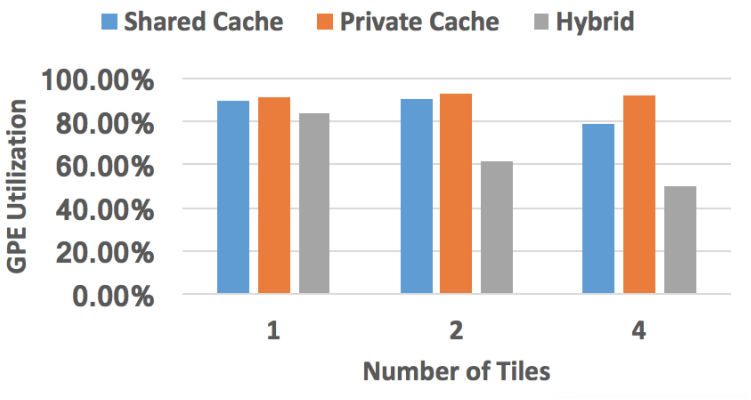



Figure 5.6: Average GPE Utilization for Convolution Step Using Shared Cache Mode, Private Cache Mode, and Hybrid Mode

As analysis of these three implementations shows that the shared and private cache mode implementations have smaller computation time and higher efficiency compared to the hybrid mode. Since the computations in these three modes are the same, the difference in execution time and efficiency suggest that the local scratchpad does not help to reduce the overall memory access time. Although the waveforms are locally stored, the size of cache is reduced by half, thus the cache miss rate increases. Even though the waveforms take up only 32% of the scratchpad size, the remaining 68% is not utilized, the utilization of the hybrid mode is the lowest.

Figure 5.6 shows the average GPE utilization for the three implementation modes. For shared cache mode, the utilization remains the same when the number of tiles increases from 1 to 2. However, the utilization reduces as the number of tiles increase to 4. For the hybrid mode, the utilization reduces as the number of tiles increase from 1 to 2 and 4. When the cache is in shared mode, the four input arrays are shared by all GPEs. A conflict occurs when more than two GPEs access the same cache bank. The cross-bar arbitrates the requests and only one GPE can access successfully while the other GPEs wait for the access. When the number of tiles increases, more GPEs

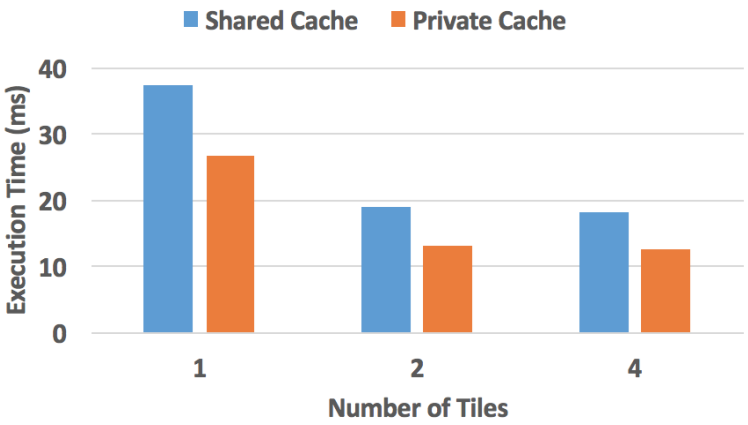



Figure 5.7: Execution Time for Beamforming Step Using Shared Cache Mode and Private Cache Mode

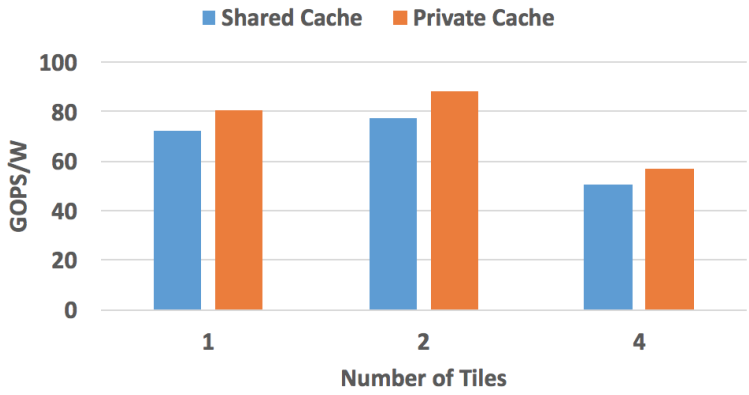



Figure 5.8: GOPS/W for Beamforming Step Using Shared Cache Mode and Private Cache Mode

access the four input arrays and this conflict happens more frequently. Thus the utilization of GPEs reduces as the number of tiles increases. When the cache is in private mode, this conflict does not exist, so the GPE utilization keeps the same.

### *5.4.3 Beamforming Results*

We implement beamforming step in both shared cache mode and private cache mode. Figure 5.7 and Figure 5.8 show the execution time and the GOPS/W for

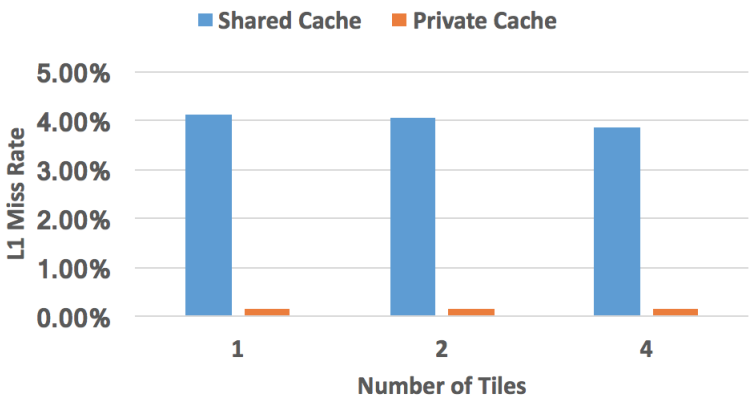



Figure 5.9: L-1 Miss Rate for Beamforming Step Using Shared Cache Mode and Private Cache Mode

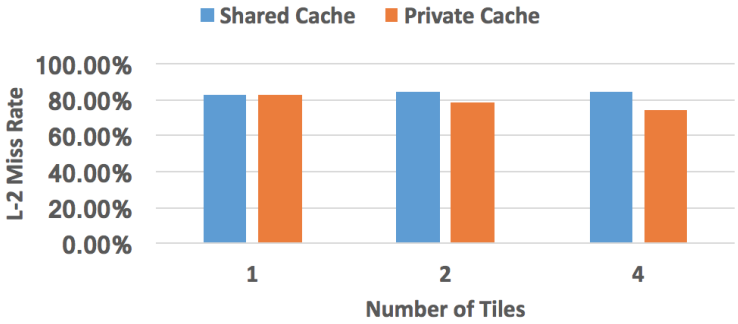



Figure 5.10: L-2 Miss Rate for Beamforming Step Using Shared Cache Mode and Private Cache Mode

the beamforming step, respectively. For both implementations, the execution time reduces when the number of tiles increase from 1 to 2, but stays the same even when the number of tiles increases to 4. The GOPS/W is the same when the number of tiles is 1 or 2, but drops as the number of tiles increases to 4. Compared to the shared cache mode, the private cache mode has shorter computation time and higher GOPS/W. This is because the beamforming computation is data bound. As different GPEs frequently access the data in the same memory bank in the shared cache mode, the access conflicts lead to large data access latency, resulting in lower efficiency.

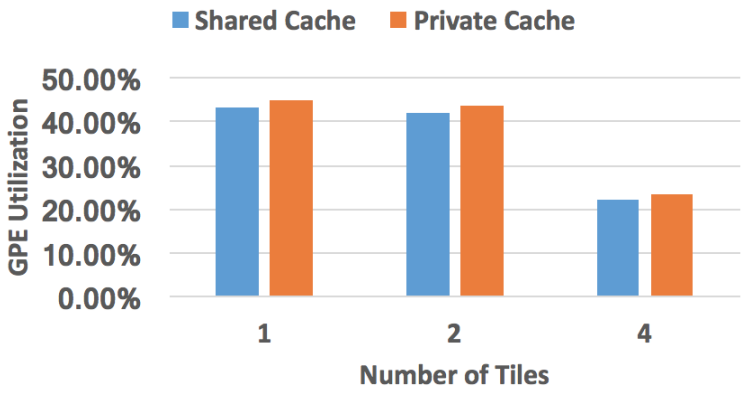



Figure 5.11: Utilization for Beamforming Step Using Shared Cache Mode and Private Cache Mode

Figure 5.9 and Figure 5.10 shows the L-1 and L-2 miss rates for the beamforming step, respectively. The L-1 miss rate is low and the L-2 miss rate is high for all cases. This is due to the fact that L-1 memory has similar size as the L-2 memory. As the data in the L-1 cache is fetched through L-2 cache, if there is a L-1 miss, it is highly probable that there will be a L-2 miss as well. So the L-2 miss rate in the shared cache mode is high. The shared cache mode has higher L-1 miss rate compared to the private mode. This is because there is little intersection between the data accessed by different GPEs. In this way, the cache sharing does not contribute to reduction in the cache misses, but it leads to more thrashing instead.

Figure 5.11 shows the average utilization of GPEs in the beamforming step. The utilization drops as the number of tiles increases for both implementations. The utilization of the configuration with 4 tiles is nearly half that of the configuration with 2 tiles. For the 4-tile case, the utilization is 23.3% which implies that the GPEs are waiting for data 76.7% of the time. Although use of L-1 private cache reduces the L-1 access conflicts, there are still conflicts in L-2 cache. So we project that the increase in conflicts in data accesses when the number of GPEs increases is the reason

for low utilization. So for beamforming step, there is no advantage to using 4 tiles and so we choose to use 2 tiles with 16 GPEs in each tile.

## 5.5 Summary

In this chapter, we described implementation of Stage 2 beamforming of 3-D SASB on TRANSFORMER. We divided Stage 2 into convolution step and beamforming step. We implemented the convolution step by configuring TRANSFORMER to operate in either shared cache mode, private cache mode, or hybrid mode. Simulation results show that implementations in cache-only modes have shorter execution time and higher efficiency compared to the hybrid mode. We implemented the beamforming step in the shared cache and private cache modes. Since the number of delay constants and apodization coefficients is large, cache-only modes are better options. In the beamforming step, the execution time does not reduce when the number of tiles increases. This step has low GOPS/W and needs to be investigated further. Beamforming on TRANSFORMER takes 13.03 $ms$ compared to 1.7 $ms$ for the filtering step. The total execution time is 14.73 $ms$, the average power consumption in 14 $nm$ technology node is 0.14 $W$, and the average GOPS/W is 103.84.

# Chapter 6

# CONCLUSIONS

Synthetic aperture sequential beamforming (SASB) divides the dynamic beamforming process into two stages, where the first stage performs a fixed focus beamforming and the second stage performs a dynamic focus beamforming. It has the advantage of achieving range-independent resolution along each scanline. From the hardware implementation aspect, it has the advantage that the data volume is largely compressed after the first stage, thereby enabling it to be transferred out and be further processed in a separate computing unit. This characteristic of SASB benefits 3-D ultrasound imaging where the data volume at the front end is much larger than its 2-D counterpart. Our main contributions are summarized below.

## 6.1 3-D Extension of SASB

In this work, we first proposed a 3-D extension of 2-D SASB that was originally proposed in [7]. We first reduced the number of active receive elements by increasing the spacing between elements. Field-II simulation results show that reducing the number of active elements from 32×32 to 16×16 does not degrade the imaging quality. We also applied separable beamforming to the second beamforming stage to reduce its computational complexity. In this implementation, the volume rate of 3-D SASB is 8.56 volumes/second, which is still quite low [18].

## 6.2 Multiple Transmit and Multiple Receive Firing Scheme (MTMR)

To increase the volume rate of 3-D SASB, we proposed the MTMR firing scheme, where four subapertures transmit and receive simultaneously. To reduce

the interference between signals transmitted by different subapertures, we used linear chirps as the excitation waveform. Compared to sinusoids, linear chirps achieve better imaging quality in the deep region. However, linear chirps in different bands generate different brightness in the reconstructed imaging volume. So we overlapped the firing process of two firing events in succession, so that each imaging volume is generated by chirps in the same band. We also designed a sparse array to avoid the grating lobes caused by large spaced uniformly distributed array in the MTMR firing scheme. In the resulting implementation, the grating lobes were reduced without increasing the computational complexity. Overall, the proposed MTMR firing scheme increases the volume rate by 4×while maintaining imaging quality with the STSR scheme [22].

## 6.3 Front-end Architecture Design for 3-D SASB

We designed the hardware architecture to implement Stage 1 of 3-D SASB. To reduce the number of computations in Stage 1, we proposed a scheme to reduce the number of distinct values in the apodization window coefficients. This resulted in the number of multiplications in the first beamforming stage being reduced by 17×. To support the proposed 3-D SASB and the sum-before-multiply scheme, we designed a highly parallel hardware architecture for Stage 1 processing. In this architecture, the signals received by 961 active transducer elements are digitized by ADCs and then routed to the corresponding digital processing channel through an NoC. The 961 digital processing channels delayed and interpolated the signals in parallel to appropriately align the wavefronts. The interpolated data are then summed up through a bus-based structure. We synthesized the data path using TSMC *28 nm* technology node, simulated NoC using *BookSim*, and estimated the power and area overhead of transducers and ADC based on existing work. The power consumption of the overall system is 1.475 W and the area is 916.03 $mm^2$.

## 6.4 Mapping Stage 2 onto TRANSFORMER

We mapped the Stage 2 computation onto a reconfigurable multi-core architecture, TRANSFORMER, that has been designed at the University of Michigan. This architecture consists of a number of tiles where each tile has multiple GPEs coordinated by a LCP. The on-chip memory types and the connection between GPEs and memories can be reconfigured at run-time to maximize the computation efficiency. We investigated different configurations of TRANSFORMER to implement the filtering step and the beamforming step in Stage 2. We implemented the filtering step using the hybrid mode, shared cache mode, and the private cache mode. Gem5 simulation results on a 4 tile – 16 GPE architecture showed that the cache-only mode outperformed the hybrid mode with shorter execution time and GOPS/W. We implemented the beamforming step using shared cache mode and the private cache mode. The private cache mode has shorter execution time and higher GOPS/W. The GPE utilization is low when the number of tiles is larger than 2 and so we choose to implement this step with two tiles. The total execution time of the two steps is 14.73 $ms$, the average power consumption is 0.14 $W$, and the average GOPS/W is 103.84.

## 6.5 Future Work

Since the beamforming step in Stage 2 has low GOPS/W, we plan to investigate techniques to hide the high data access latency. One possible solution is to design a data-structure-based pre-fetcher. Similar to a graph-based pre-fetcher in [82], we plan to propose a beamforming-based pre-fetcher, which is able to detect the delay value that the GPE is currently accessing and then pre-fetch the signal samples as well as the apodization coefficients that the GPE will access in the near future.

In the algorithm front, the design and simulations developed in this work are for ideal conditions. However, in real ultrasound imaging, the non-linearities, such as the waveform skew caused by transducer and phase aberration due to different media densities, degrade imaging quality. To address waveform skew, detailed experiments can be done to understand the real waveform generated by the transducer and a matched filtering based computation technique designed to address the skew [83]. For phase aberration, nearest-neighbor cross-correlation or near-field signal redundancy as in [84] can be studied.

REFERENCES


[1] S. Pal, J. Beaumont, D.-H. Park, A. Amarnath, S. Feng, C. Chakrabarti, H.-S. Kim, D. Blaauw, T. Mudge, and R. Dreslinski, "Outerspace: an outer product based sparse matrix multiplication accelerator," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2018, pp. 724–736.

[2] S. Stergiopoulos, *Advanced signal processing: theory and implementation for sonar, radar, and non-invasive medical diagnostic systems*. CRC Press, 2009.

[3] J. L. Prince and J. M. Links, *Medical imaging signals and systems*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[4] K. Martin, "Introduction to b-mode imaging," *Diagnostic ultrasound: physics and equipment*, vol. 2, pp. 1–10, 2010.

[5] B. P. Nelson, E. R. Melnick, and J. Li, "Portable ultrasound for remote environments, part i: Feasibility of field deployment," *The Journal of emergency medicine*, vol. 40, no. 2, pp. 190–197, 2011.

[6] M. I. Fuller, K. Owen, T. N. Blalock, J. A. Hossack, and W. F. Walker, "Real time imaging with the sonic window: A pocket-sized, c-scan, medical ultrasound device," in *2009 IEEE International Ultrasonics Symposium*. IEEE, 2009, pp. 196–199.

[7] J. Kortbek, J. A. Jensen, and K. L. Gammelmark, "Synthetic aperture sequential beamforming," in *2008 IEEE Ultrasonics Symposium*. IEEE, 2008, pp. 966–969.

[8] ——, "Sequential beamforming for synthetic aperture imaging," *Ultrasonics*, vol. 53, no. 1, pp. 1–16, 2013.

[9] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wenisch, "Sonic millip3De: an architecture for handheld 3D ultrasound," *IEEE MICRO*, vol. 34, no. 3, pp. 100–108, 2014.

[10] M. Yang, R. Sampson, S. Wei, T. F. Wenisch, and C. Chakrabarti, "Separable beamforming for 3-d medical ultrasound imaging," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 279–290, 2015.

[11] ——, "High frame rate 3-d ultrasound imaging using separable beamforming," *Journal of Signal Processing Systems*, vol. 78, no. 1, pp. 73–84, 2015.

[12] O. Lorintiu, H. Liebgott, M. Alessandrini, O. Bernard, and D. Friboulet, "Compressed sensing reconstruction of 3D ultrasound data using dictionary learning and line-wise subsampling," *IEEE Trans. on Medical Imaging*, vol. 34, no. 12, pp. 2467–2477, 2015.

[13] C. Schretter, S. Bundervoet, D. Blinder, A. Dooms, J. D'hooge, and P. Schelkens, "Ultrasound imaging from sparse rf samples using system point spread functions," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, no. 99, 2017.

[14] H. Liebgott, R. Prost, and D. Friboulet, "Pre-beamformed rf signal reconstruction in medical ultrasound using compressive sensing," *Ultrasonics*, vol. 53, no. 2, pp. 525–533, 2013.

[15] A. Burshtein, M. Birk, T. Chernyakova, A. Eilam, A. Kempinski, and Y. C. Eldar, "Sub-nyquist sampling and fourier domain beamforming in volumetric ultrasound imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 5, pp. 703–716, 2016.

[16] H. J. Vos, P. L. van Neer, M. M. Mota, M. D. Verweij, A. F. van der Steen, and A. W. Volker, "F–k domain imaging for synthetic aperture sequential beamforming," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 1, pp. 60–71, 2016.

[17] M. Schou, T. di Ianni, H. Bouzari *et al.*, "Synthetic aperture sequential beamforming using spatial matched filtering," in *2017 IEEE International Ultrasonics Symposium (IUS)*.

[18] J. Zhou, S. Wei, R. Sampson, M. Yang, R. Jintamethasawat, O. D. Kripfgans, J. B. Fowlkes, T. F. Wenisch, and C. Chakrabarti, "Low complexity 3d ultrasound imaging using synthetic aperture sequential beamforming," in *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2016, pp. 33–38.

[19] J. A. Jensen, "Field: A program for simulating ultrasound systems," in *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING*, vol. 4. Citeseer, 1996.

[20] J. A. Jensen and N. B. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 39, no. 2, pp. 262–267, 1992.

[21] J. Zhou, S. Wei, R. Sampson, R. Jintamethasawat, O. D. Kripfgans, J. B. Fowlkes, T. F. Wenisch, and C. Chakrabarti, "High volume rate 3d ultrasound imaging based on synthetic aperture sequential beamforming," in *2017 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2017, pp. 1–4.

[22] J. Zhou, S. Wei, R. Jintamethasawat, R. Sampson, O. D. Kripfgans, J. B. Fowlkes, T. F. Wenisch, and C. Chakrabarti, "High-volume-rate 3-d ultrasound imaging based on synthetic aperture sequential beamforming with chirp-coded excitation," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 65, no. 8, pp. 1346–1358, 2018.

[23] W. J. Dally and B. Towles, "Route packets, not wires: on-chip inteconnection networks," in *Proceedings of the 38th annual Design Automation Conference*. Acm, 2001, pp. 684–689.

[24] R. Marculescu, U. Y. Ogras, L.-S. Peh, N. E. Jerger, and Y. Hoskote, "Outstanding research problems in noc design: system, microarchitecture, and circuit perspectives," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 3–21, 2008.

[25] S. K. Mandal, R. Ayoub, M. Kishinevsky, and U. Y. Ogras, "Analytical performance models for nocs with multiple priority traffic classes," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, p. 52, 2019.

[26] U. Y. Ogras and R. Marculescu, *Modeling, analysis and optimization of network-on-chip communication architectures*. Springer Science & Business Media, 2013, vol. 184.

[27] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J. Kim, and W. J. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2013, pp. 86–96.

[28] R. S. Cobbold, *Foundations of biomedical ultrasound*. Oxford University Press, 2006.

[29] J. A. Jensen, S. I. Nikolov, K. L. Gammelmark, and M. H. Pedersen, "Synthetic aperture ultrasound imaging," *Ultrasonics*, vol. 44, pp. e5–e15, 2006.

[30] M. Yang, R. Sampson, S. Wei, T. F. Wenisch, and C. Chakrabarti, "Separable beamforming for 3-d medical ultrasound imaging," *IEEE transactions on signal processing*, vol. 63, no. 2, pp. 279–290, 2014.

[31] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wenisch, "Sonic millip3de: A massively parallel 3d-stacked accelerator for 3d ultrasound," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2013, pp. 318–329.

[32] T. Chernyakova and Y. C. Eldar, "Fourier-domain beamforming: the path to compressed ultrasound imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 61, no. 8, pp. 1252–1267, 2014.

[33] D. Garcia, L. Le Tarnec, S. Muth, E. Montagnon, J. Porée, and G. Cloutier, "Stolt's fk migration for plane wave ultrasound imaging," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 60, no. 9, pp. 1853–1867, 2013.

[34] E. Moghimirad, C. A. V. Hoyos, A. Mahloojifar, B. M. Asl, and J. A. Jensen, "Synthetic aperture ultrasound fourier beamformation using virtual sources," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 63, no. 12, pp. 2018–2030, 2016.

[35] H. J. Vos, P. L. van Neer, M. M. Mota, M. D. Verweij, A. F. van der Steen, and A. W. Volker, "F–kdomain imaging for synthetic aperture sequential beamforming," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 63, no. 1, pp. 60–71, 2015.

[36] J.-Y. Um, Y.-J. Kim, S.-E. Cho, M.-K. Chae, J. Song, B. Kim, S. Lee, J. Bang, Y. Kim, K. Cho *et al.*, "An analog-digital hybrid rx beamformer chip with non-uniform sampling for ultrasound medical imaging with 2d cmut array," *IEEE transactions on biomedical circuits and systems*, vol. 8, no. 6, pp. 799–809, 2014.

[37] B. Savord and R. Solomon, "Fully sampled matrix transducer for real time 3d ultrasonic imaging," in *Ultrasonics, 2003 IEEE Symposium on*, vol. 1. IEEE, 2003, pp. 945–953.

[38] J.-Y. Um, Y.-J. Kim, S.-E. Cho, M.-K. Chae, B. Kim, J.-Y. Sim, and H.-J. Park, "A single-chip 32-channel analog beamformer with 4-ns delay resolution and 768-ns maximum delay range for ultrasound medical imaging with a linear array transducer," *IEEE transactions on biomedical circuits and systems*, vol. 9, no. 1, pp. 138–151, 2015.

[39] H. gil Kang, S. Bae, P. Kim, J. Park, G. Lee, W. Jung, M. Park, K. Kim, W. Lee, and T.-K. Song, "Column-based micro-beamformer for improved 2d beamforming using a matrix array transducer," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[40] A. Ibrahim, P. A. Hager, A. Bartolini, F. Angiolini, M. Arditi, J.-P. Thiran, L. Benini, and G. De Micheli, "Efficient sample delay calculation for 2-d and 3-d ultrasound imaging," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 4, pp. 815–831, 2017.

[41] P. A. Hager, A. Bartolini, and L. Benini, "Ekho: A 30.3 w, 10k-channel fully digital integrated 3-d beamformer for medical ultrasound imaging achieving 298m focal points per second." *IEEE Trans. VLSI Syst.*, vol. 24, no. 5, pp. 1936–1949, 2016.

[42] A. Ibrahim, S. Zhang, F. Angiolini, M. Arditi, S. Kimura, S. Goto, J.-P. Thiran, and G. De Micheli, "Towards ultrasound everywhere: A portable 3d digital back-end capable of zone and compound imaging," *IEEE transactions on biomedical circuits and systems*, no. 99, pp. 1–14, 2018.

[43] R. Sampson, M. Yang, S. Wei, C. Chakrabarti, and T. F. Wenisch, "Sonic millip3de with dynamic receive focusing and apodization optimization," in *IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2013, pp. 557–560.

[44] D. P. Shattuck, M. D. Weinshenker, S. W. Smith, and O. T. von Ramm, "Explososcan: A parallel processing technique for high speed ultrasound imaging with linear phased arrays," *The Journal of the Acoustical Society of America*, vol. 75, no. 4, pp. 1273–1282, 1984.

[45] R. Mallart and M. Fink, "Improved imaging rate through simultaneous transmission of several ultrasound beams," in *Proc. SPIE*, vol. 1733, no. 1992, 1992, pp. 120–130.

[46] L. Tong, A. Ramalli, R. Jasaityte, P. Tortoli, and J. D'hooge, "Multi-transmit beam forming for fast cardiac imaging—experimental validation and in vivo application," *IEEE Trans. on Medical Imaging*, vol. 33, no. 6, pp. 1205–1219, 2014.

[47] F. Gran and J. A. Jensen, "Frequency division transmission imaging and synthetic aperture reconstruction," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 5, pp. 900–911, 2006.

[48] A. Rabinovich, A. Feuer, and Z. Friedman, "Multi-line transmission combined with minimum variance beamforming in medical ultrasound imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, no. 5, pp. 814–827, 2015.

[49] G. Matrone, A. Ramalli, A. S. Savoia, P. Tortoli, and G. Magenes, "High frame-rate, high resolution ultrasound imaging with multi-line transmission and filtered-delay multiply and sum beamforming," *IEEE Trans. on Medical Imaging*, vol. 36, no. 2, pp. 478–486, 2017.

[50] L. Demi, J. Viti, L. Kusters, F. Guidi, P. Tortoli, and M. Mischi, "Implementation of parallel transmit beamforming using orthogonal frequency division multiplexing-achievable resolution and interbeam interference," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 11, pp. 2310–2320, 2013.

[51] B. Denarie, T. Bjastad, and H. Torp, "Multi-line transmission in 3-D with reduced crosstalk artifacts: A proof of concept study," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 8, pp. 1708–1718, 2013.

[52] T. Misaridis and J. A. Jensen, "Use of modulated excitation signals in medical ultrasound. part i: Basic concepts and expected benefits," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 2, pp. 177–191, 2005.

[53] ——, "Use of modulated excitation signals in medical ultrasound. part iii: High frame rate imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 2, pp. 208–219, 2005.

[54] ——, "Use of modulated excitation signals in medical ultrasound. part ii: Design and performance for medical imaging applications," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency control*, vol. 52, no. 2, pp. 192–207, 2005.

[55] H. L. Van Trees, *Detection, estimation, and modulation theory.* John Wiley & Sons, 2004.

[56] B. Lashkari, K. Zhang, and A. Mandelis, "High-frame-rate synthetic aperture ultrasound imaging using mismatched coded excitation waveform engineering: A feasibility study," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 6, pp. 828–841, 2016.

[57] C. Yoon, Y. Yoo, T.-K. Song, and J. H. Chang, "Orthogonal quadratic chirp signals for simultaneous multi-zone focusing in medical ultrasound imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 5, pp. 1061–1069, 2012.

[58] A. Trucco, M. Palmese, and S. Repetto, "Devising an affordable sonar system for underwater 3-D vision," *IEEE Trans. on Instrumentation and Measurement*, vol. 57, no. 10, pp. 2348–2354, 2008.

[59] E. Roux, A. Ramalli, P. Tortoli, C. Cachard, M. C. Robini, and H. Liebgott, "2-D ultrasound sparse arrays multidepth radiation optimization using simulated annealing and spiral-array inspired energy functions," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 12, pp. 2138–2149, 2016.

[60] A. Austeng and S. Holm, "Sparse 2-D arrays for 3-D phased array imaging-design methods," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 49, no. 8, pp. 1073–1086, 2002.

[61] A. Ramalli, E. Boni, A. S. Savoia, and P. Tortoli, "Density-tapered spiral arrays for ultrasound 3-D imaging," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, no. 8, pp. 1580–1588, 2015.

[62] B. Diarra, M. Robini, P. Tortoli, C. Cachard, and H. Liebgott, "Design of optimal 2-D nongrid sparse arrays for medical ultrasound," *IEEE Trans. on Biomedical Engineering*, vol. 60, no. 11, pp. 3093–3102, 2013.

[63] C. Sciallero and A. Trucco, "Design of a sparse planar array for optimized 3D medical ultrasound imaging," in *23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 1341–1345.

[64] W. J. Hendricks, "The totally random versus the bin approach for random arrays," *IEEE Trans. on Antennas and Propagation*, vol. 39, no. 12, pp. 1757–1762, 1991.

[65] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein, "The thermodynamic approach to the structure analysis of crystals," *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, vol. 37, no. 5, pp. 742–754, 1981.

[66] P. Chen, Y. Zheng, and W. Zhu, "Optimized simulated annealing algorithm for thinning and weighting large planar arrays in both far-field and near-field," *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 658–664, 2011.

[67] A. Trucco, "Thinning and weighting of large planar arrays by simulated annealing," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 46, no. 2, pp. 347–355, 1999.

[68] T. L. Szabo and P. A. Lewin, "Ultrasound transducer selection in clinical imaging practice," *Journal of Ultrasound in Medicine*, vol. 32, no. 4, pp. 573–582, 2013.

[69] T. Schmidt, C. Hohl, P. Haage, M. Blaum, D. Honnef, C. Wei$\beta$, G. Staatz, and R. Gunther, "Diagnostic accuracy of phase-inversion tissue harmonic imaging versus fundamental b-mode sonography in the evaluation of focal lesions of the kidney," *American Journal of Roentgenology*, vol. 180, no. 6, pp. 1639–1647, 2003.

[70] T. Di Ianni, M. C. Hemmsen, P. L. Muntal, I. H. H. Jørgensen, and J. A. Jensen, "System-level design of an integrated receiver front end for a wireless ultrasound probe," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 11, pp. 1935–1946, 2016.

[71] M. Bae, N. O. Kim, S.-B. Park, and S.-J. Kwon, "A novel beamforming method for wireless ultrasound smart probe," in *Ultrasonics Symposium (IUS), 2014 IEEE International*. IEEE, 2014, pp. 2185–2188.

[72] T. Di Ianni, T. K. Kjeldsen, C. A. V. Hoyos, J. Mosegaard, and J. A. Jensen, "Real-time implementation of synthetic aperture vector flow imaging on a consumer-level tablet," in *Proc. IEEE Ultrason. Symp*, 2017, pp. 1–4.

[73] T. Kjeldsen, L. Lassen, M. C. Hemmsen, C. Kjær, B. G. Tomov, J. Mosegaard, and J. A. Jensen, "Synthetic aperture sequential beamforming implemented on multi-core platforms," in *Proc. IEEE Ultrason. Symp*, 2014, pp. 2181–2184.

[74] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[75] B. D. Steinberg, "Digital beamforming in ultrasound," *IEEE Transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 39, no. 6, pp. 716–721, 1992.

[76] R. G. Pridham and R. A. Mucci, "Digital interpolation beamforming for low-pass and bandpass signals," *Proceedings of the IEEE*, vol. 67, no. 6, pp. 904–919, 1979.

[77] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 12, no. 3, p. 23, 2007.

[78] V. Rantala, T. Lehtonen, J. Plosila *et al.*, *Network on chip routing algorithms*. Citeseer, 2006.

[79] Y.-C. Lien, "A 4.5-mw 8-b 750-ms/s 2-b/step asynchronous subranged sar adc in 28-nm cmos technology," in *2012 Symposium on VLSI Circuits (VLSIC)*. IEEE, 2012, pp. 88–89.

[80] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, "The gem5 simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[81] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 14, no. 2, p. 14, 2017.

[82] S. Ainsworth and T. M. Jones, "Graph prefetching using data structure knowledge," in *Proceedings of the 2016 International Conference on Supercomputing*. ACM, 2016, p. 39.

[83] M. Pollakowski and H. Ermert, "Chirp signal matching and signal power optimization in pulse-echo mode ultrasonic nondestructive testing," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 41, no. 5, pp. 655–659, 1994.

[84] Y. Li and R. Gill, "A comparison of matched signals used in three different phase-aberration correction algorithms," in *1998 IEEE Ultrasonics Symposium. Proceedings (Cat. No. 98CH36102)*, vol. 2, 1998, pp. 1707–1712.