Sonic Millip3De: A Massively Parallel 3D-Stacked Accelerator for 3D Ultrasound

Richard Sampson*, Ming Yang†, Siyuan Wei†, Chaitali Chakrabarti†, and Thomas F. Wenisch*

*Department of EECS, University of Michigan
†School of ECEE, Arizona State University

Abstract

Three-dimensional (3D) ultrasound is becoming common for non-invasive medical imaging because of its high accuracy, safety, and ease of use. Unlike other modalities, ultrasound transducers require little power, which makes hand-held imaging platforms possible, and several low-resolution 2D devices are commercially available today. However, the extreme computational requirements (and associated power requirements) of 3D ultrasound image formation has, to date, precluded hand-held 3D capable devices.

We describe the Sonic Millip3De*, a new system architecture and accelerator for 3D ultrasound beamformation—the most computationally intensive aspect of image formation. Our three-layer die-stacked design features a custom beamsum accelerator that employs massive data parallelism and a streaming* transform-select-reduce *pipeline architecture enabled by our new iterative beamsum delay calculation algorithm. Based on RTL-level design and floorplanning for an industrial 45nm process, we show* Sonic Millip3De *can enable 3D ultrasound with a fully sampled 128x96 transducer array within a 16W full-system power budget (400x less than a conventional DSP solution) and will meet a 5W safe power target by the 11nm node.*

1. Introduction

Sustained advancement in mobile processor performance and display technology is bringing hand-held medical imaging from science fiction to reality. Conventional X-ray, ultrasound, and MRI systems are large and bulky; many are immobile, and even "portable" devices are often larger than household appliances. However, recent work ranging from specialized architectures (e.g., the MEDICS accelerator for portable X-ray CT [11]) to data acquisition enhancements (e.g., DSIQ A/D conversion [36]) and FPGA-based system designs (e.g., Sonic Window [15]) are advancing performance within limited power envelopes to make hand-held operation feasible. The portability of hand-held imaging devices is not simply a matter of convenience; clinical studies have demonstrated that patient outcomes improve [17, 41], especially for patients in critical condition. Moreover, improved portability holds the potential to bring advanced medical imaging to rural populations in the developing world.

Three-dimensional (3D) ultrasound is a particularly attractive modality for hand-held imaging because ultrasound transducers use little power (limited by FDA regulations to a few hundred milliwatts [31]) and pose no known dangers or side-effects, in contrast to X-ray and MRI [35, 38]. 3D ultrasound provides numerous benefits over its 2D counterpart. Not only are 3D images easier to interpret, reducing effort (and errors) for technicians to locate relevant anatomy, they also provide accurate volumetric measurements of cysts and tumors that 2D cannot match. In fact, prior to 3D imaging, technicians sometimes resorted to estimating cyst volumes by mentally piecing together 2D slice images [8].

However, the benefits of 3D also come with numerous hardware challenges that are only exacerbated when trying make the system hand-held. To construct a 3D volumetric image, a conventional linear transducer array (e.g., 128 elements) for 2D imaging must be replaced with a rectangular array (128×96 in our aggressive design), increasing the incoming data rate by 100×. Furthermore, rather than reconstruct a typical 2D image resolution of 50×4096 focal points, the 3D image comprises 50×50×4096 focal points, another factor of 50 increase. The computational requirements increase by the product of these factors (nearly 5000×).

Because it is in close contact with human skin, an ultrasound scan head must operate within a tight power budget (about 5W) to maintain safe temperatures. Though transducer power is negligible relative to this limit, the raw data rate produced by a 128×96 high-resolution transducer array exceeds 6 Tb/s—so high that it cannot even be transferred off chip for processing. In 2D systems, delay constants used for beamforming are easily pre-computed and stored; for our target 3D system, over 125 billion such constants are required and must be computed on-the-fly, nominally requiring billions of square root and trigonometric operations. The challenge of 3D hand-

978-1-4673-5587-2/13/$31.00 ©2013 IEEE

held ultrasound lies in performing these computations within a 5W budget.

Implementing 3D ultrasound with commercially available DSP/GPU chips and conventional beamformation algorithms is simply infeasible, requiring over 700 DSP chips with a total power budget of 7.1kW.

In this paper, we describe the *Sonic Millip3De*, a system architecture and specialized accelerator unit for low-power 3D ultrasound beamformation. The Sonic Millip3De makes use of a massively parallel hardware design and state-of-the-art 3D die stacking [7, 14, 24, 26], splitting analog components, analog-to-digital (ADC) converters and SRAM storage, and a 1024-unit beamsum accelerator array across three silicon layers for a compact design with short (and hence low-power) wires. The accelerator array is organized according to a streaming *transform-select-reduce* design paradigm and is enabled by a novel algorithm for iteratively computing beamformation delay constants that balances pre-computed value storage with on-the-fly calculations while requiring only table lookup and add operations. The system architecture builds on recent ultrasound advances including sub-aperture multiplexing [18, 23] and virtual sources [22, 33].

Based on RTL-level design and floorplanning for an industrial 45nm process, we estimate a full-system power requirement of 16W for Sonic Millip3De and project that it will meet the 5W target power budget by the 11nm node.

In brief, we present the following contributions:

- A 3D stacked system design that allows transducer arrays, SRAMs, and the computation engine to be manufactured in separate technologies and connected via short, vertical links, minimizing power while meeting the extreme bandwidth requirements of 3D ultrasound beamformation.
- An iterative algorithm for beam delay calculation that reduces pre-computed constant storage by over 400 times, eliminates all multiply and square root operations, and admits greater data locality than conventional algorithms.
- A *transform-select-reduce* framework for designing accelerators for data-intensive streaming algorithms and our beamsum accelerator array, which implements the delay calculation algorithm using only table lookups and narrow-bit-width adds.
- An image quality analysis using Field II simulations [19, 20] demonstrating the accuracy of our delay calculation algorithm and low-power narrow-bit-width fixed-point functional units.
- Power, performance, and area results from our RTL-level design demonstrating that Sonic Millip3De can meet a 16W full-system power budget in 45nm technology (over 400× reduction compared to a conventional DSP solution) and will meet out 5W goal by the 11nm node.

In Section 2, we provide a brief overview of ultrasound imaging. Then, in Section 3, we describe the novel delay calculation algorithm. In Section 4, we introduce the *transform-select-reduce* accelerator paradigm and describe Sonic Millip3De. We report on image quality, power and performance in Section 5. In Section 6, we discuss related work. We reflect on our design process in Section 7 and in Section 8, we conclude.

2. Background

We begin with a brief introduction to ultrasound and summary of essential techniques from the ultrasound literature on which the Sonic Millip3De builds.

2.1. Ultrasound Overview

Ultrasound is performed by sending high frequency pulses (typically 1-15MHz) into a medium and constructing an image from the pulse signals that are reflected back. The process comprises three stages: transmit, receive, and beamsum. Transmission and reception are both done using an array of capacitive micromachined ultrasonic transducers (CMUTs) that are electrically stimulated to produce the outgoing signal and generate current when they vibrate from the returning echo. After all echo data is received, the beamsum process (the compute intensive stage) combines the data into a partial image. The partial image corresponds to echoes from a single transmission. Several transmissions from different locations on the transducer array are needed to produce high quality images, so several iterations of transmit, receive, and beamsum are necessary to construct a complete frame. We use 16 transmit locations in our design.

Each transmission is a pulsed signal conceptually originating from a single location in the array, shown in Figure 1(a). To improve signal strength, multiple transducers typically fire together in a pattern to emulate a single *virtual source* located behind the transducer array [33]. The pulse expands into the medium radially, and as it encounters interfaces between materials of differing density, the signal will partially transmit and partially reflect as shown in Figure 1(b). The returning echoes cause the transducers to vibrate, generating a current signal that is digitized and stored in a memory array associated with each transducer. Each position within these arrays corresponds to a different round-trip time from the emitting transducer to the receiving transducer. Because transducers cannot distinguish the direction of an incoming echo, each array element

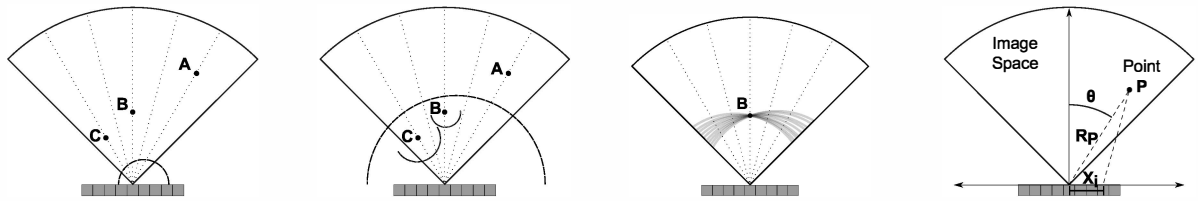


Figure 1: Ultrasound Background. (a) Pulse leaving transmit transducer. (b) Echo pulses reflecting from points *B* and *C*. All transducers in array (or sub-aperture) will receive the echo data, but at different times due to different round trip distances. (c) All of the reconstructed data for point B from each of the transducers added together. By adding thousands of "views" together, crisp points become visible. (d) Variables used in calculating round trip distance, d_p, for the *i*-th transducer and point *P* in Eq. 1.

contains the superimposed echoes from all locations in the imaging volume with equal round-trip times (i.e., an arc in the imaging volume). The beamsum operation sums the echo intensity observed by all transducers for the arcs intersecting a particular *focal point* (i.e., a location in the imaging volume), yielding a strong signal (i.e., a bright point in the image) when the focal point lies on an echoic boundary.

A beamsum pipeline must first transform the raw signal received from each transducer for further processing. Typically, an interpolation filter first generates additional data points between received samples, enabling greater resolution. Then a constant apodization scaling factor is applied to each signal, which weights the transducer's contribution to the beamsum based on the ultrasound beam's angle of incidence.

The imaging volume geometry is described by a grid of *scanlines* that radiate at a constant angular increment from the center of the transducer array into the image volume. Focal points are located at even spacing along each scanline. In essence, the beamsum operations entail calculating the round-trip delay between the emitting transducer and all receiving transducers through a particular focal point, converting these delays into indices in each transducers' received signal array, retrieving the corresponding data, and summing these values. Figure 1(c) illustrates this process. An image is formed by iterating over all desired focal points and performing beamsum for each. Once an image has been formed, a demodulation step removes the ultrasound carrier signal.

2.2. Delay Calculation

The delay calculation (identifying the right index within each receive array) is the most computationally intensive aspect of beamsum as it must be completed for every {focal point, transmit transducer, receive transducer} trio. Typically, delays are calculated via

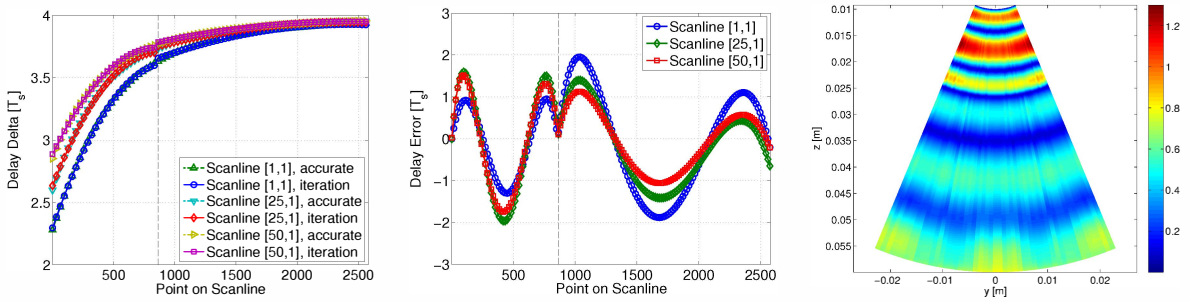
$$d_P = \frac{1}{c}\left(R_P + \sqrt{R_P^2 + x_i^2 - 2x_i R_P sin\theta}\right) \qquad (1)$$

where d_P is the roundtrip delay from the center transducer to the point P to transducer i, c is the speed of sound in tissue (1540 m/s), R_P is the radial distance of point P from the center of the transducer, θ is the angular distance of point P from the line normal to the center transducer, and x_i is the distance of transducer i from the center. Figure 1(d) shows variables as they correspond to the system geometry. This formula applies the law of cosines to calculate the round-trip distance, and requires extensive evaluation of both trigonometric functions and square roots. Hence, many 2D ultrasound systems pre-calculate all delays and store them in a lookup table (LUT) [1, 21]. However a typical 3D system requires roughly 250 billion delay values, making a LUT implementation impractical. Instead, delays are calculated as needed [42]. One of the central innovations of our design is to replace this expensive delay calculation with an iterative approximation algorithm that stores far fewer pre-computed values and requires only add operations at runtime. We describe this algorithm in Section 3.

2.3. Receive Sub-aperture Multiplexing

Another challenge of 3D beamformation lies in managing the deluge of data that must be transferred from receiving transducers to functional units that perform the summation. In existing 3D ultrasound systems, the receive data is transferred from the scan head (which typically does not contain significant compute capability) via cable to separate systems that perform beamsum. For the transducer array geometry we assume, the receive data arrives at a rate of approximately 6Tb/s and comprises roughly 100MB per transmit; hence it is both too large to store in the scan head and arriving to quickly to transfer over cables.

To manage data transfer, modern systems employ *sub-aperture multiplexing*, wherein only data from a sub-aperture (a part of the imaging volume) are stored and transferred upon each transmit [18, 23]. Sub-apertures are sized based on the bandwidth available in the link from the transducer head to the computing



(a) (b) (c)

Figure 2: Delay curve fitting and analysis. (a) The exact delta between neighboring points for representative scanlines and the estimates from our iterative algorithm. The dotted line indicates the boundary of the 2-section piecewise approximation. (b) The error between our approximation and the exact delta, normalized to the index unit (T_s). With two sections, our algorithm never errs by more than 3 samples. (c) Root mean square error for an entire y-z image slice.

platform. Transmission from a single source is repeated several times in succession, capturing a different sub-aperture each time. Hence, over a sequence of transmits, all {transmit, receive, focal point} trios are obtained. Though effective at reducing data rates, this technique entails some compromise in image quality as neither the patient nor the scan head is entirely still between transmits, resulting in some motion blur.

We incorporate 12-way sub-aperture multiplexing into our design to enable multiplexing of A/D converters and receive memory arrays, allowing a design with only a few MB of on-chip SRAM.

3. Algorithm Design

Conventional beamforming algorithms (intended for implementation in software) are too storage- and compute-intensive for hand-held 3D ultrasound. In this section, we describe our algorithmic innovations that enable a low-power hardware implementation.

3.1. Iterative Index Calculation

As discussed previously, delay calculation is enormously compute-intensive, requiring either numerous processors or a large LUT of pre-calculated values. Neither of these approaches is feasible in a small hand-held device. Prior work has reduced delay calculation computational complexity through iterative methods [5, 27], but these methods still require billions of expensive square root operations. Instead, we redesign the algorithm to require tractable storage and eliminate the trigonometric and square root operations required in a straight-forward implementation.

The key insight of our algorithm is to replace prior iterative index calculations [27] with a piecewise quadratic approximation that can be computed using only add operations. Because focal points are evenly spaced, the delta function between adjacent focal point delays form a smooth curve and indices can be approximated accurately (with error similar to that introduced by interpolation) over short intervals with quadratic approximations. We replace these exact delta curves with a per-transducer pre-computed piece-wise quadratic approximation constrained to allow an index error of at most 3 (corresponding to at most 30μm error between the estimated and exact focal point) thus resulting in negligible blur. Figure 2(a) compares our approximation to the exact difference between adjacent delays for three representative scanlines. Figure 2(b) shows the corresponding round-trip delay error. Figure 2(c) shows the root mean square (RMS) error for the full y-z slice through the middle of the image. Our approach drastically reduces storage requirements relative to pre-computing all delays because only four constants (three for the delta function; one for the section boundary) are pre-computed and stored per section. Because of its simplicity, this approximation requires only table lookups (to retrieve constants) and adds (to iteratively calculate the delay).

Via offline image quality studies, we have determined that our piece-wise quadratic estimation method requires only two sections to meet our accuracy target for the typical abdominal imaging aperture we use in our evaluation. Hence, we must store at most nine (two sections of four constants and a start value) constants per scanline. In contrast, complete delay pre-computation requires a 4096-entry LUT per scanline, over 400× more storage than our method. By exploiting symmetry in the imaging geometry, we can further reduce the required constant storage by a factor of four, for an aggregate storage requirement of 77MB. Nevertheless, 77MB exceeds the storage capacity we can provision on chip. However, our system only uses 250kB of constants at a time, allowing us to load the constants as needed from DRAM.

3.2. Narrow-Width Fixed-Point Arithmetic

We employ narrow-bit-width fixed-point arithmetic to further reduce the storage and bandwidth requirements of our design. We again performed an offline study to analyze image quality at various bit widths. Our analysis (see Section 5.3) concludes that 12-bit fixed-point precision sacrifices negligible image quality relative to double-precision floating point, but any further bit-width reduction leads to significant quality degradation. Using custom-width 12-bit ADCs and calculation pipelines substantially reduces hardware and power requirements relative to conventional 16-bit or wider DSP solutions [1, 21].

4. Hardware Architecture

We next describe the Sonic Millip3De system architecture and its key features, including the beamforming accelerator that implements our iterative delay calculation algorithm in a massively parallel array.

4.1. System Architecture

The Sonic Millip3De system (Figure 3) comprises three stacked silicon dies (transducers and analog electronics, ADC and storage, and computation) connected vertically using through-silicon vias (TSVs) and off-stack LPDDR2 memory. These components are integrated in the ultrasound scanhead, the wand-like device a radiologist manipulates to obtain ultrasound images. Our design focuses on the so-called "front-end" of an ultrasound system, which controls the transducer array and constructs a volumetric image. A separate "back-end" renders a view (either 2D slices or a 3D perspective) of the image. We envision a system where the scanhead sends image data to a tablet or other presentation system; we do not consider the design of the "back-end" in this work.

We split the design over three 3D-stacked layers for several reasons. First, the technology requirements of each layer differ substantially. The geometry of the transducer array requires a much larger die and higher voltages than the SRAM arrays or beamforming accelerator and can be economically manufactured in an older process technology. In contrast, the power hungry ADC/memory and computation layers benefit from exploiting the latest process technology.

Second, the layout of the transducer array is tightly coupled to the transmit frequency and target imaging aperture; ultrasound systems typically feature interchangeable scan heads with varying array geometries for different imaging tasks (e.g., different imaging depths). By separating the transducer array, ADC/storage, and computation engine into separate dies, a standard interface (i.e., TSV layout) between each enables dies to be reused with varying transducer array layers, reducing design costs.

Finally, as in recent 3D-stacked processor architectures where caches and cores are connected vertically [14], the face-to-face connections between SRAM arrays and corresponding computation units avoid the need for long wires.

The transducer die comprises a 128×96 grid of optimally spaced transducers whose centers are exactly $\lambda/2$ apart where λ is the wavelength of the transmit signal [25]. We assume a 4MHz transmit frequency, requiring a minimum die size of 24mm×18mm—much larger than the other layers, which are 15mm×15mm each. The area between transducers contains the analog electronics and routing to the TSV interface to the ADC/storage die. Transducers are grouped into 1024 banks of twelve (3×4) transducers each. One transducer within each bank is assigned to one of twelve receive sub-apertures. During each transmit cycle, only a single transducer among the twelve in each bank will receive data and pass it to the ADC layer. Hence, twelve consecutive transmits are required to process the entire aperture. Transducers within a bank are multiplexed onto a single signal per bank that is passed over a TSV to the ADC/storage layer for digitization.

The ADC/storage layer comprises 1024 12-bit ADCs, each connected to an incoming analog signal from the transducer layer. The ADCs sample at a frequency of 40MHz, well above the Nyquist limit of even the fastest transducer arrays (15MHz). This sampling frequency balances energy efficiency and flexibility for ultrasound applications requiring varying transmit frequencies. After digitization, the received signals are stored in 1024 independent SRAM arrays, each storing 4096 12-bit samples. The SRAMs are clocked at 1GHz. Each SRAM array is connected vertically to a corresponding functional unit on the computation layer, requiring a total of 24,000 face-to-face bonded data and address signals.

The computation layer is the most complex of the three. It includes the beamforming accelerator units, a unidirectional pipelined interconnect, a control processor (e.g., an M-class ARM core), and an LPDDR2 memory controller. The die area is dominated by the beamforming accelerator array and interconnect, which are described in the following subsections. The control processor manages memory transfers from the LPDDR2 interface to the accelerator array, controls the transducer array, and performs other general purpose functions (e.g., configuration, boot). The off-stack LPDDR2 memory stores index delay constants and a frame buffer for the final volumetric image. While the control processor has a small cache, the accelerator ar-

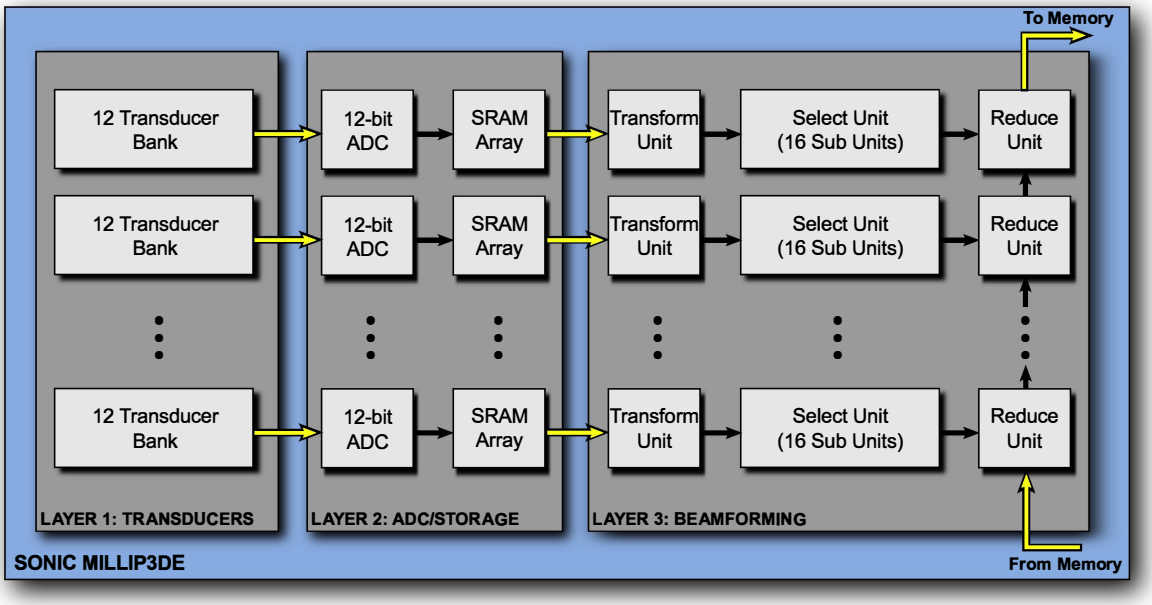


Figure 3: Sonic Millip3De Hardware Overview. Layer 1 (24×18mm) comprises 128x96 transducers grouped into banks of 3x4 transducers each. Analog transducer outputs from each bank are multiplexed and routed over TSVs to Layer 2, comprising 1024 12-bit ADC units operating at $40MHz$ and SRAMs arrays to store incoming samples. The stored data is passed via face-to-face links to Layer 3 for processing in the 3 stages of the 1024-unit beamsum accelerator. The transform stage upsamples the signal to $160MHz$. The 16 units in select stage map signal data from the receive time domain to the image space domain in parallel for 16 scanlines. The reduce stage combines previously-stored data from memory with the incoming signal from all 1024 beamsum nodes over a unidirectional pipelined interconnect, and the resulting updated image is written back to memory.

ray performs only bulk memory transfers and requires no cache hierarchy or coherence mechanism.

The Sonic Millip3De memory system comprises a 192-bit wide memory channel striped across 6 2Gb x16 LPDDR2-800 parts. This unusual arrangement matches the width of our on-chip interconnect, provides sufficient capacity (1.5 GB) and sufficient memory bandwidth (38.4 GB/sec) to load beamforming constants (requiring 6.2 GB/sec) and read/write image data (requiring 5.5 GB/sec) for our target imaging rate of one frame per second while still requiring little power (see Section 5) [28, 32].

4.2. The Beamforming Accelerator

The beamforming accelerator is the central element of Sonic Millip3De, and is the key to achieving our performance and power objectives. The accelerator relies on massive parallelism (1024 beamforming units operate in concert) and achieves energy efficiency through carefully optimized 12-bit data paths that perform only add, compare, and table lookup operations.

Recall (from Section 2) that a single ultrasound frame is obtained by summing the received data from 12 receive sub-apertures over 16 different virtual sources. For each of these 192 receive operations, the entire imaging volume is read from memory (15MB), the (single) correct sample from each transducer in the sub-aperture is added to each focal point, and the volume is stored back to DRAM. Below, we describe a single of these 192 receive operations: the data flow during each receive is identical, only the apodization and delay constants differ across receives.

The accelerator follows a streaming *transform-select-reduce* data flow paradigm. We first describe the principle of this approach and how the algorithm described in Section 3 maps to this framework.

Principle of Operation. As shown in Figure 3, the accelerator streams data in parallel from all 1024 SRAM arrays on the ADC/storage layer (corresponding to the 1024 transducer banks) to 1024 corresponding beamforming units. The data streams pass through three conceptual stages: *transform*, *select*, and *reduce*. Each stage is implemented in a separate pipelined functional unit and a unidirectional pipeline interconnect (starting and terminating at the LPDDR2 interface) links the reduce units together.

In the conceptual model, the *transform* stage performs pre-processing that must be applied to all data. For beamforming, this stage performs interpolation (which upsamples the 40MHz signal to 160MHz).

The *select* stage transforms data from the receive time domain into the image space domain. In essence, it provides a mapping from the input data streams (i.e.,

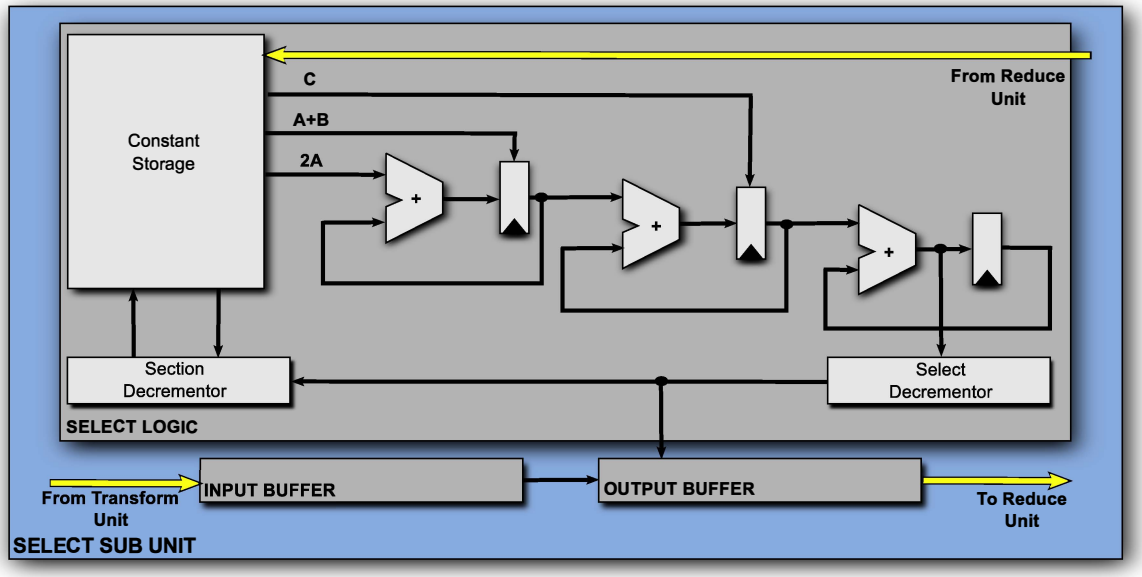


Figure 4: Select Unit Microarchitecture. Select units map incoming samples from the receive time domain to image focal points. Sample data arrives from the interpolation unit at the input buffer, and each sample is either discarded or copied to the output buffer to accumulate a particular focal point. The unit selects the correct sample for each focal point using the indexing algorithm in Section 3. The Constant Storage holds the 3 approximation constants and boundary for each approximation section. The first adder calculates $2AN + A + B$, the second adds the $N - 1$ result of the quadratic equation to create the value for N, and the final adder accumulates fractional bits from previous additions. The Select Decrementor is initialized with the integer component of the sum. Each cycle, the head of the input buffer is copied to the output if the decrementor is zero, or discarded if it is non-zero. The Section Decrementor tracks when to advance to the next piece-wise approximation section.

the signal from each transducer) into slots (i.e., focal points in the image) over which the reduce operation will be performed. The key innovation of the select stage is that we have eliminated the index-load-add-store sequence that software implementations use to map from the time domain to the image space. Rather than iterate over focal points and compute the memory addresses of the required samples, our unit iterates over the samples and either discards them or bins them to be accumulated into the correct focal points. The select units rely on pre-computed constants to indicate the mapping from samples to focal points.

Finally, the *reduce* stage performs a reduction operation across the *transform-select* pipelines. For beamsum, the reduction operation is simply an add. The main feature of the *reduce* stage is the interconnect that links the pipelines. In cases where the reduce operation is commutative, this could be an arbitrary interconnect. In our design, we use a unidirectional pipeline with wide, short links forming a 1024-stage pipeline.

Beamforming Node. A beamforming node comprises a transform unit, 16 select sub units, and a single reduce unit linked to the global interconnect. There are 1024 such nodes, each connected to a corresponding ADC/SRAM channel and transducer bank. Each node occupies roughly 400μm × 400μm (see Section 5).

Processing a Receive. Once a receive operation is complete (all SRAM buffers are filled) the control unit activates the accelerator. The entire beamforming array processes only 16 adjacent scanlines at a time, traversing the entire input data in the SRAMs for these scanlines before reprocessing the data for the next 16-scanline bundle; 157 such bundles are processed.

Between bundles, receive delay and apodization constants are loaded from memory and fed into all the beamforming units by sending control packets addressed to each unit around the interconnect. About 250kB of constants must be loaded per bundle.

The select units within each beamforming node all operate independently—each selecting focal point data for one of the 16 scanlines. The select units arbitrate for access to the transform unit and request the next 4 transducer samples, which the transform unit interpolates and apodizes to produce 16 properly weighted samples. Select units continue requesting input data and outputting focal point data until they fill their 16-entry output buffer, at which point they block. The select unit microarchitecture is detailed in Figure 4.

In the mean time, image data from earlier receives is loaded into the network's ingress node. Data is read from each of the 16 active scanlines in a round-robin fashion, 16 focal points (192 bits) at a time, and

injected into the network.

For a particular scanline, when both the select unit's output and the data arriving via the interconnect are available, the two are added and propagated to the next beamforming node. Thus, each 16-point bundle from each scanline will visit every beamforming node, accumulating the appropriate incoming sample. The scanline flows circle the network independently. When the last data return to the control processor at the egress node, the next bundle of scanlines is processed. When all bundles have been processed, the SRAM buffers on the ADC/storage layer are cleared and the control processor triggers the next transmit.

Transform Unit. The transform unit includes a linear interpolation unit, which upsamples the transducer data upon request from a select unit. Upsampling is a standard technique in ultrasound to improve image resolution without the power overhead of faster ADCs [8].

Select Units. Select units map the interpolated receive data to focal points for a single scanline using the algorithm described in Section 3. Because the unit processes focal points in order of increasing distance from the scanhead, the round-trip delays increase monotonically (Figure 2(a)). Hence, the unit can select the correct sample for each focal point in a single pass over the receive data.

The select unit block diagram is shown in Figure 4. The *Constant Storage* block stores the delay constants and section boundaries used in the index approximation algorithm. Constants are loaded between each scanline bundle. The *Input Buffer* and *Output Buffer* are FIFO queues. Also shown are three adders, which calculate the next delay delta via our quadratic estimation, and two decrementors, which orchestrate input data selection and the piece-wise quadratic sectioning.

Whenever the Input Buffer is empty, the sampling unit requests the next 16 12-bit samples from the interpolation unit. The select unit then generates the index of the first focal point on the scanline by adding the transmit (Tx) and receive (Rx) constants, placing the sum in the *Select Decrementor*. Each cycle, the Select Decrementor decrements. If the value is non-zero, then the head of the input buffer is discarded—that input sample does not correspond to the next focal point. However, when the value becomes zero, the head of the input buffer is appended to the output buffer—this input sample will be added to the next focal point.

The Section Decrementor counts down the remaining focal points in the current section. When it reaches zero, the constants and boundary for the next approximation section are loaded from the Constant Storage.

Through this simple use of decrementors, adders, and a few pre-computed constants, the sampling unit completely avoids the need for complex delay calculations. It is this simple design that enables the enormous energy efficiency gains of the Sonic Millip3De.

Reduce Unit & Interconnect. The reduce unit contains an array of 16 192-bit buffers (one per scanline) and 16 12-bit adders. Whenever both the buffer and select unit for a particular scanline are ready, the values are added and passed to the next beamforming node.

The reduce units are connected via the unidirectional pipeline interconnect. Each link is 192 bits wide, and clocked at 1GHz. The network provides a peak bandwidth of 22.3GB/s between neighboring nodes, comfortably exceeding the minimum requirements to achieve 1 frame per second. The links between beamforming units are nearly 400μm long, and are routed on a quad-spaced metal layer. The wires are not repeated, as an entire clock cycle is available to traverse between units. Because of the sheer number of wires (192 links each between 1024 beamforming units), the interconnect accounts for a substantial fraction of the overall power of the Sonic Millip3De system (see Section 5).

5. Methodology and Results

We evaluate Sonic Millip3De in two ways. First, we validate that algorithmic approximations and fixed-point rounding errors do not compromise image quality. Second, we report full-system power requirements in 45nm, and project when technology scaling will enable our 5W objective for safe human skin contact.

5.1. Why a specialized accelerator?

To motivate the need for a specialized accelerator, we consider the power requirements of a strawman 3D ultrasound design built with stock, high-end DSPs typically used in 2D systems. We do not consider GPUs or other high-performance processors because even one such chip requires orders of magnitude more power than the 5W objective. In particular, we consider the TI C6678, TI's recommended processor for portable ultrasound imaging [4]. This 8-core processor features optimized fixed- and floating-point functional units. Karadayi and co-authors report that this chip computes one 4096-point scanline for a 512-transducer array (64 per core) in 740μs using state-of-the-art algorithms [21]. As our receive sub-apertures comprise 1024 transducers, we scale Karadyi's result to 1480μs per scanline per transmit.

Our design assumes a 3D aperture with 2500 (50×50) scanlines, and must process 192 total transmits per frame (16 virtual sources & 12 sub-apertures) as shown in Table 1. From these back-of-the-envelope values we estimate that a 6678-based design requires 710 chips (drawing nearly 7kW [1]) to meet

Parameter	Value
Sub-apertures	12
Virtual Sources	16
Total Transmits per Image	192
Total Transducers	12,288
Receive Transducers per Sub-aperture	1024
Storage per Receive Transducer	4096 x 12-bits
Focal Points per Scanline	4096
Image Depth	6cm
Image Total Angular Width	$\pi/2$
Sampling Frequency	40MHz
Interpolation Factor	4x
Interpolated Sampling Frequency	160MHz
Speed of Sound (tissue)	1540m/s
Target Frame Rate	1fps

Table 1: 3D ultrasound system parameters.

Architecture	Energy/Scanline (1 fps)	Single Core Time/Scanline
Intel Core i7-2670	25.08*J*	4.46*s*
ARM Cortex-A8	33.04*J*	132.18*s*
TI C6678 DSP	2.84*J*	2.27*s*

Table 2: Efficiency of conventional DSPs and CPUs.

the 1 frame-per-second goal. We further compare to other energy-efficient processors running the algorithm from [21]; however, these designs are even less efficient than the DSP solution (Table 2). Without orders-of-magnitude innovation in algorithms, implementing 3D ultrasound with conventional DSPs is infeasible in the foreseeable future.

The orders-of-magnitude power gap between Sonic Millip3De and the conventional processors arises from eliminating the "von Neumann" penalty of a general purpose processor (we implement the innermost loop of beamforming entirely in hardware), our indexing algorithm that requires only adds (eliminating multiples and square roots), and careful minimization of data movement and off-chip DRAM accesses (each focal point is loaded/stored only once per transmit).

5.2. Evaluation Methodology

To evaluate our own design, we use the ultrasound parameters shown in Table 1. We analyze image quality using Field II [19, 20] (a widely-used simulation framework for ultrasound imaging built on top of Matlab) to simulate echo signals of a 5mm cyst in tissue. We visualize the resulting images using Matlab.

The objective of our evaluation is to measure the image quality loss that arises due to our iterative delay calculation and 12-bit fixed point functional units. Hence, we contrast our algorithm against a double-precision floating point simulation using precise delays (calculated via Eq. 1).

We quantify image quality by measuring contrast-to-noise ratios (CNR) of imaging the 5mm cyst. CNR is a standard measure of the accuracy of ultrasound

Bits	10	11	12	13	14	Ideal
CNR	2.233	2.536	2.942	2.960	2.942	2.972

Table 3: CNR vs. precision. Ideal indicates double precision floating point and exact delay calculations.

imaging [9, 39]; it measures the contrast resolution of echoic regions (tissue) and anechoic regions (cyst) under the effects of receiver noise and clutter (reflections from outside the imaging aperture that produce artifacts within the image). CNR is defined as:

$$CNR = \frac{|\langle L_{cyst}\rangle - \langle L_{background}\rangle|}{\sqrt{\sigma^2_{cyst} + \sigma^2_{background}}} \quad (2)$$

where $\langle L_{cyst}\rangle$ and $\langle L_{background}\rangle$ are the mean signal within the cyst and background areas, respectively, and σ^2_{cyst} and $\sigma^2_{background}$ are the corresponding variances.

To measure hardware power and performance, we synthesize an RTL-level specification in Verilog using an industrial 45nm standard cell library and SRAM compiler. We report power, timing, and area estimates from synthesis. We model interconnect power and performance using SPICE. We estimate ADC power from recently published designs in 40nm [30, 40]. The cited ADC designs provide 11-bit precision and operate at 6× higher frequency than we require. We scale down operating frequency by 6×, but conservatively estimate that increasing precision to 12 bits quadruples power requirements, for a net power scaling factor of 2/3. We estimate DRAM power from reported efficiency of LPDDR2 designs [28], assuming 12 x16 2Gb parts.

5.3. Image Quality

We contrast images of our cyst model using precise index calculations (i.e., Equation 1) and double-precision floating-point against the same image reconstructed via our iterative delay method and 12-bit fixed-point precision as implemented in the Sonic Millip3De hardware. Figure 5 (a) shows a slice from the baseline double-precision simulation, while (b) shows the same slice using our methods and fixed-point beamsum. The baseline algorithm achieves a CNR of 2.972, while our design produces a nearly indistinguishable image with CNR of 2.942 (higher values indicate better contrast). Reducing precision to 11 bits (c), however, results in noticeable artifacts and CNR of only 2.536. Table 3 shows CNR for a range of precision.

5.4. Full-System Power

We next report Sonic Millip3De's power requirements in 45nm technology and project requirements in future nodes; scaling trends are shown in Figure 6 while a detailed breakdown appears in Table 4. Using a combinations of synthesis results and quoted estimates

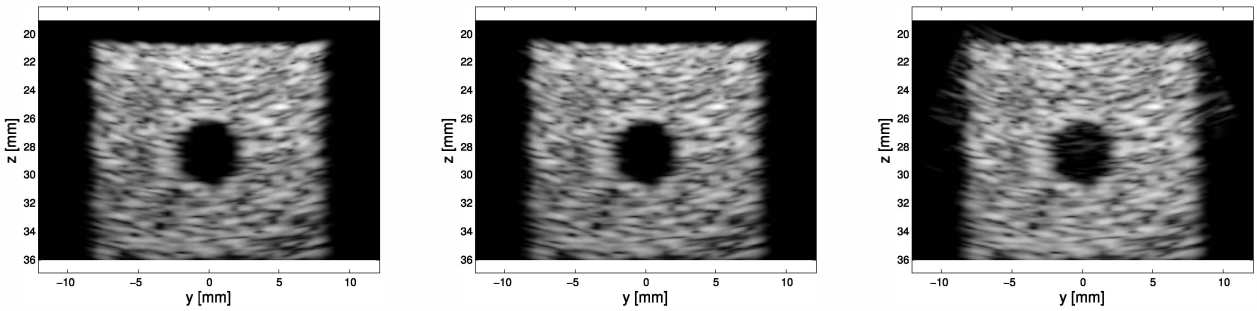


Figure 5: Image Quality Comparison. (a) Y-Z (vertical) slice through cyst from a 3D simulation using Field II [19, 20], generated with double-precision floating point and exact delay calculation (Eq. 1). CNR is 2.972. (b) The same slice generated via our delay algorithm and 12-bit fixed-point precision. CNR is 2.942. (c) Same as (b), with 11-bit precision. CNR is 2.536.

	Transducer	ADC [40]	SRAM	Beamsum	Interconnect	Mem. Interface [6]	DRAM [28]	Total
Sonic Millip3De (45*nm*)	0.3W	1.2*W*	0.197*W*	4.8*W*	5.04*W*	0.007*W*	3.8*W*	15.3*W*
Sonic Millip3De (11*nm*)	0.3W	0.146*W*	0.049*W*	1.2*W*	1.23*W*	0.002*W*	0.461*W*	3.39*W*

Table 4: Power Breakdown. SRAM power from an industrial 45*nm* SRAM compiler. Beamsum unit power from synthesized RTL using 45*nm* standard cells. Interconnect power from 45*nm* SPICE simulations. Scaling prediction trends from [12, 30].

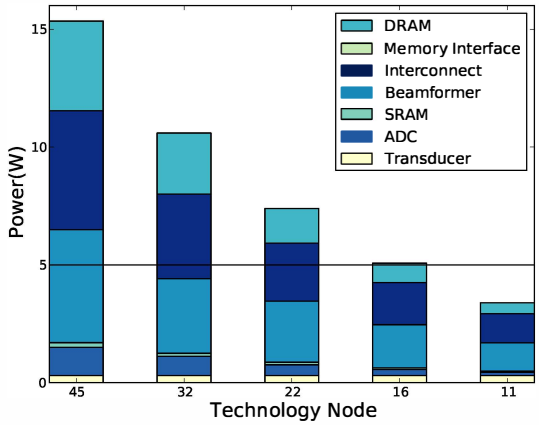


Figure 6: Power Breakdown Across Technology Nodes. Scaling projections based on trends reported in [12, 30]. We project meeting the 5*W* power budget at the 11*nm* node.

([28, 40]), we determine that the Sonic Millip3De system requires a total of 15.3W in 45nm, falling short of the target 5W power budget for safe use on humans. However, we note that over 60% of the power is dissipated in the compute layer. Hence, further architectural and circuit innovation and technology scaling can close the power gap.

We project when Sonic Millip3De might meet our power target using published scaling trends. For this analysis, we use ADC scaling trends from [30], technology scaling from [12], and assume that wire power does not scale (though our wires get shorter from transistor shrinking). We project that Sonic Millip3De will fall just short of our goal in 16nm, and meet the 5W target by the 11nm node.

6. Related Work

Several ultrasound system vendors have recently released 2.5D (3D reconstruction by assembling 2D slices) and 3D capable systems [2, 34]. To achieve sufficient frame rates with tractable hardware requirements, these systems: (1) use much smaller transducer arrays, (2) sample only a subset of transducers upon each receive, or (3) mechanically or electrically sweep a linear array to collect elevational data. All of these simplifications greatly reduce computational complexity at the cost of aperture restrictions and reduced spatial image resolution.

Numerous hand-held 2D devices are available [3, 29]. These devices have demonstrated the potential impact hand-held ultrasound can have on medicine in the developing world. However, all current hand-held units have considerably lower image quality and resolution than bench units, and none are 3D capable.

The academic ultrasound community has made substantial efforts to further the state-of-the-art in 2D hand-held systems [15, 36], for example, using FPGA-based designs. The success of these efforts has accelerated the commercial availability of 2D hand-held ultrasound. However, to conserve power these systems rely on a number of algorithmic trade-offs that sacrifice quality or lead to visual artifacts when compared to bench systems. Our design focuses exclusively on high-resolution 3D image formation to achieve comparable image quality to non-portable commercial 3D designs. Academic work on 2.5D and 3D systems has focused mostly on using clusters of CPUs [42] or GPUs [10, 37] to perform beamsum, both of which are too large and power hungry for a hand-held device.

Most of these efforts have focused on 2.5D systems.

In the computer systems community, researchers have proposed custom processor architectures for portable medical imaging for other modalities. For example, the MEDICS [11] architecture proposes a number of innovations to reduce the power requirements of X-ray CT image reconstruction.

Finally, the Sonic Millip3De system architecture builds on a growing body of work on 3D die stacked computer architectures [7, 14, 24, 26]. These works have not only demonstrated that 3D die stacking is possible (through fabricated test chips), but also show how to exploit the benefits of 3D stacked design for massively-parallel low-power systems (e.g., reducing wire lengths by stacking functional units and caches).

7. Discussion

Sonic Millip3De is an example of the application-specific accelerator design that we believe will become increasingly common as architects seek to deliver ever higher computing performance within the limited energy and thermal budgets of hand-held platforms. A key take-away from this effort is the necessity of co-designing algorithms and hardware when order-of-magnitude efficiency gains are needed: to make beamforming amenable to an efficient hardware implementation, we had to risk degrading image quality through new algorithmic approximations. We subsequently tuned these optimizations via image quality analysis to achieve acceptable error.

We began our design process by investigating where software implementations of ultrasound beamforming spend time. To our surprise, computation was dominated by the indexing operations that calculate memory addresses rather than computations on the received signals themselves, which lead us to investigate less compute-intensive indexing approximations. The piecewise quadratic fit presented in Section 3 was the end result of several approximation attempts; for each, we studied image quality impact using Field II first for 2D and then for 3D imaging.

Following our redesign of the indexing algorithm, a further key design realization came when we observed that indices increase monotonically along a scanline, enabling a single-pass streaming approach to align the received samples to focal points. This realization led to the genesis of the transform-select-reduce architecture and the Sonic Millip3De data path. Moreover, by accessing memory through streams, our design could manage the high memory bandwidth requirements and naturally exploit locality in a way that traditional von Neumann designs could not.

Given the transform-select-reduce architecture, the remaining parameters of our design then followed from balancing resources to achieve the one frame-per-second design target, which is sufficient for a variety of ultrasound applications [13, 16]. Simple back-of-the-envelope calculations indicated the design required roughly 1024 parallel channels to achieve this framerate, which in turn drove other design decisions (e.g., memory bandwidth, storage).

We hope that Sonic Millip3De might serve as a case study that demonstrates the potential of 3D die stacking technology and that our approach will prove valuable to those designing similar accelerators for entirely different computing domains.

8. Conclusions

Ultrasound imaging already provides doctors and patients access to safe internal imaging, and advances in existing 3D designs have demonstrated improved ease of use, image quality, and accuracy for volumetric measurements. Although 2D hand-held ultrasound systems are available today, high-resolution 3D imaging remains infeasible in hand-held platforms due to the massive computational requirements of 3D beamsum.

In this work, we have described Sonic Millip3De, a new 3D stacked accelerator unit for hand-held 3D ultrasound. Our design combines a streaming *transform-select-reduce* accelerator architecture with our newly developed iterative delay calculation algorithm to minimize power requirements and exploit data locality. Using synthesis of RTL-level hardware design for an industrial 45nm standard cell process, we have shown that our design can enable volumetric imaging with a fully sampled (128×96) transducer array within a 16W full-system power budget (over 400× less power than a DSP-based solution). Based on current scaling trends, we project that the Sonic Millip3De will meet the 5W target power budget for safe use on humans by the 11nm technology node.

Acknowledgements

This work was partially supported by NSF CCF-0815457, CSR-0910699 and grants from ARM, Inc. The authors wish to thank J. Brian Fowlkes, Oliver Kripfgans, and Paul Carson for feedback and assistance with image quality analysis and Ron Dreslinski for assistance with SPICE.

References

[1] TMS320C66x Multicore DSPs for High-performance Computing. http://www.ti.com/lit/ml/sprt619/sprt619.pdf.

[2] Voluson S8. http://www.gehealthcare.com/euen/ultrasound/products/general-imaging/voluson-s8/VolusonS8_v16.pdf.

[3] Vscan Pocket Ultrasound. http://www3.gehealthcare.com/en/Products/Categories/Ultrasound/Vscan.

[4] TMS320C6678 Data Manual, Feb. 2012. http://www.ti.com/lit/ds/symlink/tms320c6678.pdf.

[5] M. Ali, D. Magee, and U. Dasgupta. Signal Processing Overview of Ultrasound Systems for Medical Imaging. Technical report, Texas Instraments, Nov 2008.

[6] ARM. Cortex-M3 40G Specifications. http://www.arm.com/products/processors/cortex-m/cortex-m3.php.

[7] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCaule, P. Morrow, D. W. Nelson, *et al.* Die Stacking (3D) Microarchitecture. In *Proc. of the 39th International Symp. on Microarchitecture*, 2006.

[8] R. S. C. Cobbold. *Foundations of Biomedical Ultrasound*. Oxford University Press, 2007.

[9] J. Dahl, G. Trahey, and G. Pinton. The effects of image degradation on ultrasound-guided HIFU. In *Proc. of IEEE International Ultrasonics Symp.*, Oct. 2010.

[10] Y. Dai, J. Tian, D. Dong, G. Yan, and H. Zheng. Real-Time Visualized Freehand 3D Ultrasound Reconstruction Based on GPU. *IEEE Trans. on Information Technology in Biomedicine*, 14(6):1338–1345, Nov. 2010.

[11] G. Dasika, A. Sethia, V. Robby, T. Mudge, and S. Mahlke. MEDICS: ultra-portable processing for medical image reconstruction. In *Proc. of the 19th International Conf. on Parallel Architectures and Compilation Techniques*, 2010.

[12] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Proc. of the 38th International Symp. on Computer Architecture*, 2011.

[13] P. Falkensammer and H. Brandl. Ultrasound Technology Update:4D Fetal Echocardiography. http://www3.gehealthcare.com/en/Products/Categories/Ultrasound/Vscan.

[14] D. Fick, R. Dreslinski, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu, *et al.* Centip3De: a 3930 DMIPS/W configurable near-threshold 3D stacked system with 64 ARM cortex-M3 cores. In *Proc. of International Solid-State Circuits Conference*, Feb. 2012.

[15] M. Fuller, K. Owen, T. Blalock, J. Hossack, and W. Walker. Real time imaging with the Sonic Window: A pocket-sized, C-scan, medical ultrasound device. In *Proc. of IEEE International Ultrasonics Symposium*, Sept. 2009.

[16] F. Gaufillet, H. Liegbott, M. Uhercik, F. Cervenansky, J. Kybic, and C. Cachard. 3D Ultrasound real-time monitoring of surgical tools. In *Proc. of IEEE International Ultrasonics Symp.*, Oct. 2010.

[17] T. Gunnarsson, A. Theodorsson, P. Karlsson, S. Fridriksson, S. Boström, J. Persliden, I. Johansson, and J. Hillman. Mobile computerized tomography scanning in the neurosurgery intensive care unit: increase in patient safety and reduction of staff workload. *J. of Neurosurgery*, 93(3):432–436, Sept. 2000.

[18] G. U. Haugen, K. Kristoffersen, and D. G. Wildes. Ultrasound probe sub-aperture processing, 2009.

[19] J. Jensen. FIELD: A Program for Simulating Ultrasound Systems. In *Nordicbaltic Conf. on Biomedical Imaging*, 1996.

[20] J. Jensen and N. Svendsen. Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 39(2):262 –267, March 1992.

[21] K. Karadayi, C. Lee, and Y. Kim. Software-based Ultrasound Beamforming on Multi-core DSPs. Technical report, University of Washington, March 2011. http://www.ti.com/lit/wp/sprabo0/sprabo0.pdf.

[22] M. Karaman, P.-C. Li, and M. ODonnell. Synthetic aperture imaging for small scale systems. *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 42(3):429–442, 1995.

[23] M. Karaman and M. O'Donnell. Subaperture processing for ultrasonic imaging. *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 45(1):126–135, Jan. 1998.

[24] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, and K. Flautner. PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor. In *Proc. of the 12th Conf. on Arch. Support for Programming Languages and Operating Systems*, 2006.

[25] G. Lockwood, J. Talman, and S. Brunke. Real-time 3-D ultrasound imaging using sparse synthetic aperture beamforming. *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 45(4):980–988, July 1998.

[26] G. H. Loh. 3D-Stacked Memory Architectures for Multi-core Processors. In *Proc. of the 35th International Symp. on Computer Architecture*, 2008.

[27] D. P. Magee. Iterative Time Delay Values for Ultrasound Beamforming, 2010.

[28] K. Malladi, F. Nothaft, K. Periyathambi, B. Lee, C. Kozyrakis, and M. Horowitz. Towards energy-proportional datacenter memory with mobile DRAM. In *Proc. of 39th International Symp. on Computer Architecture*, June 2012.

[29] Mobisante. MobiUS SP1 Product Overview. http://www.mobisante.com/product-overview/.

[30] B. Murmann. "ADC Performance Survey 1997-2012". http://www.stanford.edu/~murmann/adcsurvey.html.

[31] T. Nelson, J. Fowlkes, J. Abramowicz, and C. Church. Ultrasound biosafety considerations for the practicing sonographer and sonologist. *J. of Ultrasound in Medicine*, 28(2):139, 2010.

[32] R. Palmer, J. Poulton, B. Leibowitz, Y. Frans, S. Li, A. Fuller, J. Eyles, J. Wilson, M. Aleksic, T. Greer, *et al.* A 4.3 GB/s mobile memory interface with power-efficient bandwidth scaling. In *Symp. on VLSI Circuits*, 2009.

[33] C. Passmann and H. Eermert. A 100-MHz ultrasound imaging system for dermatologic and ophthalmologic diagnostics. *IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control*, 43(4):545–552, July 1996.

[34] Phillips. iU22 xMatrix Ultrasound System. http://www.healthcare.philips.com/main/products/ultrasound/systems/iu22/index.wpd.

[35] J. L. Prince and J. M. Links. *Medical Imaging: Signals and Systems.* Pearson Prentice Hall, 2006.

[36] K. Ranganathan, M. K. Santy, J. A. Hossack, T. N. Blalock, and W. F. Walker. Direct Sampled I/Q Beamforming: A Strategy for Very Low Cost Ultrasound. In *Proc. of IEEE International Ultrasonics Symposium*, 2003.

[37] T. Reichl, J. Passenger, O. Acosta, and O. Salvado. Ultrasound goes GPU: real-time simulation using CUDA. In *Proc. of SPIE 7261*, 2009.

[38] S. Stergiopoulos, editor. *Advanced Signal Processing: Theory and Implementation for Sonar, Radar, and Non-Invasive Medical Diagnostic Systems.* The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2009.

[39] K. F. Üstüner and G. L. Holley. Ultrasound Imaging System Performance Assessment. American Assocation of Physicists in Medicine Annu. Meeting, 2003.

[40] B. Verbruggen, M. Iriguchi, and J. Craninckx. A 1.7mW 11b 250MS/s 2x interleaved fully dynamic pipelined SAR ADC in 40nm digital CMOS. In *Proc. of 2012 IEEE International Solid-State Circuits Conf.*, Feb. 2012.

[41] D. Weinreb and J. Stahl. The introduction of a portable head/neck CT scanner may be associated with an 86% increase in the predicted percent of acute stroke patients treatable with thrombolytic therapy. *Radiological Soc. of N. America*, 2008.

[42] F. Zhang, A. Bilas, A. Dhanantwari, K. N. Plataniotis, R. Abiprojo, and S. Stergiopoulos. Parallelization and performance of 3D ultrasound imaging beamforming algorithms on modern clusters. In *Proc. of the International Conf. on Supercomputing*, 2002.