WHITE PAPER

Ambric Am2045

Medical Ultrasound

NETHRA® IMAGING INC.

Proprietary and Confidential

MKAN000409_v1.0
August 6, 2009

Revision History

Revision	Date	By	Description
v1.0	8/6/09		No changes to content. Document changed to reflect acquisition of Ambric by Nethra Imaging

1 Introduction

Ultrasonography has been used in medical imaging for over half a century. Clinical ultrasound is a relatively inexpensive, portable, safe, and real-time modality, making it one of the most widely used imaging modalities in medicine. Recent advances in signal acquisition and digital signal processing (DSP) technologies enable ultrasound equipment to produce high quality images for early detection and monitoring of various disease states. The performance and versatility of clinical ultrasound equipment can be generally measured by the following key factors:

- Imaging frequency
- Beamforming efficiency
- Frame rate
- Imaging intelligence (detection, classification, measurement)
- Power efficiency

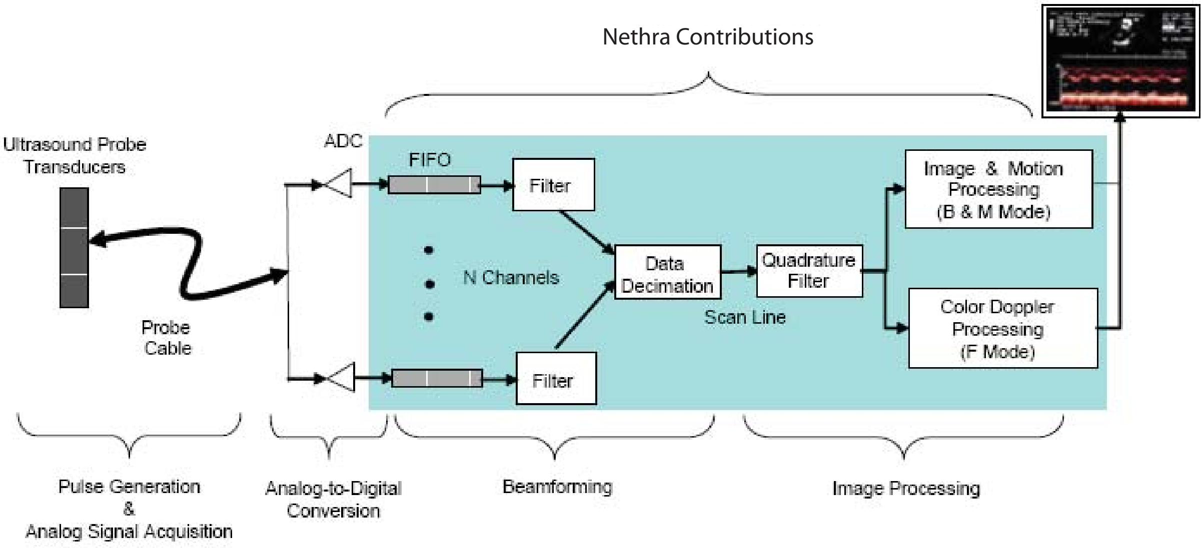
Because ultrasound requires tremendous computing power to process large amounts of data in parallel, ultrasound designers have been forced to take an ASIC/FPGA approach at the expense of design time, engineering resources, and fab cost; only to have an inflexible and non-reusable implementation in the end. Designers must be able to focus their energies on creating product value algorithms, and to spend less time in design implementation. The massively parallel programmable processing array (MPPA) technology from the Nethra® Am2045 MPPA is the enabling solution.

This paper gives the brief overview of a medical ultrasound system, describing its major functional blocks and identifying the key factors in determining an ultrasound system's quality and performance. It introduces a unique programming model that transforms the ultrasound processing design from a traditionally hardware-intensive engineering effort into a software programming task. It explains how Nethra technology meets ultrasound processing challenges. Finally, it describes the developmental tools that significantly enhance a designer's productivity.

2 *Ultrasound System Overview*

Ultrasound images are formed by the continuous emission and detection of ultrasonic pulses, which are partly transmitted and partly reflected from a boundary between two tissue structures. By detecting the amount of energy reflected from an anatomical cross-section and measuring its time delay (depth), images are formed using sophisticated digital signal and image processing algorithms. Figure 1 shows a conceptual block diagram of the key components of a medical ultrasound system. Nethra contributes significant value in the areas of beamforming and image processing.

Figure 1: Clinical Ultrasound System Conceptual Block Diagram



2.1 Pulse Generation

An ultrasound system employs a probe with 48 to 256 piezoelectric crystal transducers to generate and acquire ultrasound signals. The transducers, which function as both signal generators and receivers, are linked to the ultrasound processing subsystem by a cable with a micro-coaxial bundle inside. When they vibrate, transducers generate ultrasound pulses by applying alternating currents across the crystals. Lower frequencies (1 to 6 MHz) are often used to image deeper structures, like the liver and kidney, because of their better penetration capability. However, lower frequencies have poor resolution in the axial (parallel to the beam) and lateral (perpendicular to the beam) directions. Medium frequencies (7 to 15 MHz) are used for superficial structures, such as muscles, tendons, or breasts; because they provide better axial and lateral resolutions. Intravenous cardiovascular ultrasound may use frequencies as high as 40 MHz.

Image quality optimization in medical ultrasound instruments requires a systems perspective on transducers and beamforming, because there is an interaction between new developments in both fields. The following sections describe transducers, beamformers, and their relationships.

2.2 Linear Array Transducers

Depending on the application, an ultrasound probe may use either a linear array or a phased-array transducer. A linear-array transducer has a bank of crystal elements aligned in a row. At the time of pulse generation, a subset of the transducer elements fires simultaneously to form a beam. To produce each subsequent beam, a new group of elements forms by dropping and adding one element sequentially, as shown in Figure 2. The pulses must be focused to produce a cleaner image, just as a lens focuses light in a camera. The transducer focuses by firing shaped pulses from several transducers according to a delay profile. The concavely shaped pulses form a beam with a focus point, as shown in Figure 3. Continuously firing and acquiring beams produces an image frame, scan line by scan line. A linear-array transducer activates a smaller number of elements at one time. Consequently, it provides the advantage of less parallel processing and lower cost. It has the disadvantage of greater time to image a given field of view, resulting in lower frame rates.

Figure 2: Linear-array Beamforming with a Focus Point

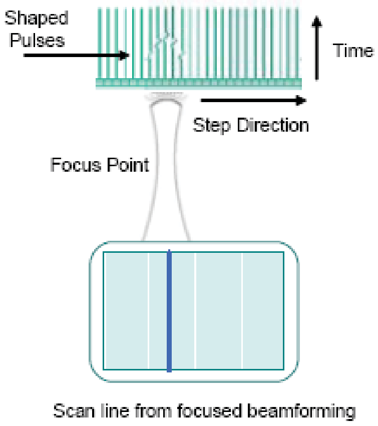
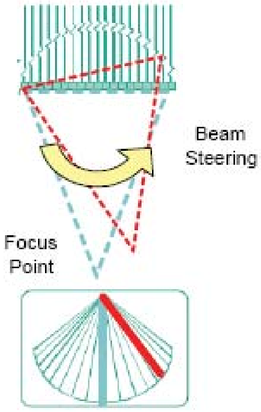


Figure 3: Phased-array Beamforming



2.3 Phased-array Transducers

For applications like cardiac ultrasound, transducers require a sufficiently small footprint to emit beams through the small spaces between bones of a rib cage, and field wide enough to image the whole heart. This task is difficult to accomplish with a linear array. To operate under such constraints, the beams must diverge from virtually the same point. This means an image must be generated by a single beam originating from the same point which is deflected in different angles to build the component sector images.

Unlike a linear-array transducer (firing only a subset of elements at a time), a phased-array transducer excites all elements at the same time to make a concave wavefront resulting in a focused beam with its narrowest part at some distance from the probe. By stimulating the transducers in a rapid sequence, the ultrasound is sent out in an interference pattern. According to Huygens' Principle, the wavefront behaves as a single beam. Thus the beam is formed by all transducers in the array, and the direction is determined by the time sequence, similar to the principle used in phased-array radar. It is possible to steer the ultrasound beam so that it effectively fires at an angle to the transducer, as shown in Figure 3. When multiple beams are fired sequentially at different angles, the beam effectively sweeps step-wise over a sector of the anatomy.

A probe with a wider transducer aperture produces better lateral resolution. A larger aperture requires more active transducer elements, which increases the signal processing path (channel) throughout the ultrasound processing subsystem. For example, a 5 MHz transducer with 64 elements generally has a maximum active aperture size around of 2 cm. This provides lateral resolution of approximately 0.6 mm at a 4 cm depth. An increase in aperture size to 128 elements improves the lateral resolution to 0.3 mm at the same depth but doubles the processing requirements.

2.4 Data Acquisition

Transducers turn ultrasonic echoes into voltages transmitted to the ultrasound processing subsystem through a probe cable. Analog-to-digital converters (ADCs) convert waveforms into digital values by rapid data sampling. The resulting signal's dynamic range depends on the number of bits per sample from the ADC, typically 12 to 14 bits. The output data from each ADC goes to a FIFO for beamforming.

Higher imaging frequencies, with shorter wavelengths and higher sampling rates, improve axial and lateral resolutions. Faster sampling rates generate more data, which requires higher computing power to complete the process within a given time interval.

2.5 Beamforming

Beamforming is the most processing-intensive part of an ultrasound system. As described earlier, the signal from each individual element is delayed to focus and steer the beam in the desired direction. In the receive beamformer this gives rise to the concept of dynamic focusing. For each pulse transmitted from the array, the receive beamformer tracks the depth and focus of the receive beam as the depth increases. Since a weighting function (apodization) is often used for sidelobe reduction, the element weights must also be dynamically updated with depth.

Each transducer element has a FIFO for collecting samples after the ADC. The samples from each FIFO must be time-aligned according to its delay profile. Further adjustments, such as focus weight factor and calibration parameters, must be taken into consideration before samples from all relevant FIFOs are summed up to produce one point in a beam. If the system clock does not match the data sampling frequency, which is typical, sample values are filtered for interpolating to produce the desired result. This process is repeated until a beam is formed for a given depth. Although the beamforming computation is not exceptionally complex, the volume of data to be processed in real time challenges the system architecture and processing power.

2.6 Frame Formation

By repeating the beamforming process, an image frame is formed scan line by scan line. However, before the data is used for higher level image processing, a quadrature filter must be applied to convert RF data into two streams of binary-baseband signals, representing cosine and sine channels. These two digital signals are used to recover both the amplitudes and phases of the signals received at each element of the array.

2.7 Frame Rate

Monitoring and measuring moving objects and structures such as blood and heart, are very important in ultrasound applications. Although the eye generally sees only 25 fps, a higher frame rate offers the possibility of replay at lower rate. For example, images acquired at 50 fps played at 25 fps double the eye's effective temporal resolution. Advanced diagnostic methods, such as speckle tracking in strain rate tracking, may require frame rates up to 70 fps.

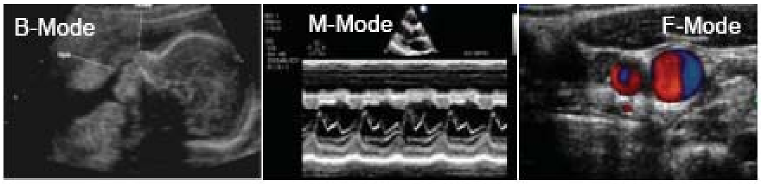
Advanced ultrasound systems may use several parallel receive beams to increase frame rate, of great importance in certain modes in cardiography (especially pediatric and color flow imaging). A 3D-imaging system requires parallel beams over an area instead of along a line, and so benefits from using four to sixteen parallel beams. This all requires a vast increase in the number of elements and processing demand.

2.8 Image Intelligence

Coherent echo signals widely undergo digital processing to extract intelligence from the images they produce prior to image presentation. Well-known methods include resolution enhancement, contrast enhancement to suppress speckle, and imaging spectral parameters. Combinations of spectral parameters and ancillary clinical data are also used with statistical classifiers to generate color-coded images to indicate tissue type (e.g. normal or cancerous) or tissue regions responding to therapy.

Several modes are available to display the final ultrasound images after processing. The mode most familiar to the general public is called the B-mode. B-mode is a two-dimensional gray-scale picture showing the cross section of a physiological structure. An M-mode (Motion) image is formed by a rapid sequence of B-mode scans. Images appear on a screen as a movie loop enabling radiologists to see and measure motion range as organ boundaries producing reflections move relative to a probe. M-mode ultrasound is widely used in studying heart motion. F-mode (Colorflow Doppler) is most useful for accurately measuring the velocity moving material, such as blood in arteries and veins. Examples of these modes are shown in Figure 4.

Figure 4: B-mode, M-mode, F-mode Images



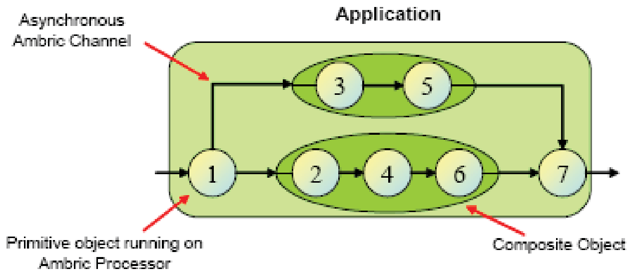
3 Answers to Challenges

3.1 Programming Model

A unique programming model is the enabling technology that transforms a traditional hardware design into a software programming task. To appreciate how using Nethra technology to meet the challenges imposed by demanding ultrasound applications, it is important to understand the programming model used with the Am2045 MPPA.

This unique programming model assumes that a modern chip integrates so many processor cores that a simple software object (a subroutine, more or less) runs exclusively on its very own processor. No other objects in the program need to contend for the same resources and more complex objects may run on multiple processors. This model allows logically divided chunks of code to run with a high degree of autonomy and independence. The MPPA processors are interconnected in hardware through Nethra's smart channels. Each channel is word-wide, unidirectional, and point-to-point. Objects running on processors are synchronized to channel activity. When an object tries to output a word to a full channel, the processor stalls until space is available. The same is true when an object tries to input from an empty channel. This concept is illustrated in Figure 5.

Figure 5: Programming Model



This programming model enables application developers to start with a system block diagram, typical of the “white board” architecture diagrams projects commonly start with. According to the algorithm chosen for the application, this system model is developed into a hierarchical structure of composite and primitive objects, connected by channels that carry structured data and control messages. Finally, ordinary sequential software is written to implement the primitive objects that are not already available in a library.

3.2 High IO and Memory Bandwidth

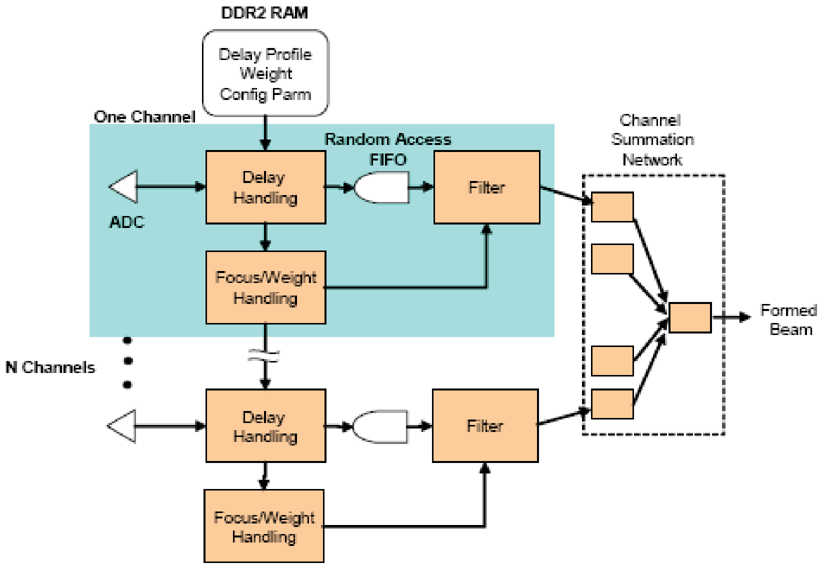
Ultrasound processing subsystems must have sufficient bandwidth to sustain high imaging frequencies. The Am2045 has an aggregate IO throughput of 28.8 Gbps and a 4-lane PCIe interface. Its 4 GPIO ports are each programmable as single 32-bit or dual 16-bit IO devices. Each Am2045 has 7.9 Mb internal SRAM and can access 512 MB of external DDR2 DRAM. The aggregate memory bandwidth is 3.2 tera-bps for SRAM and 26 Gbps for DDR2 memory.

3.3 MPP Ideal for Beamforming

Until now, the need to process massive amounts of data in parallel meant most beamforming logic was implemented using ASICs or FGPAs. This conventional approach incurs the need for even more logic, and adds overhead for global flow control or explicit pipeline scheduling to synchronize computing elements at run-time. The Am2045, with its massively parallel processing (MPP) architecture, naturally handles such processing challenges.

The Am2045 has over 300 DSP/RISC processors with grouping flexibility to process data streams from each transducer. Processors are interconnected by smart channels, with topology driven by the application design. Synchronization among processors is handled by the smart channels with handshake mechanisms implemented in silicon. A processor suspends or resumes execution by itself at the channel, according to the channel's input or output availability, without needing explicit checking by the application. Figure 6 shows a beamforming example based on Nethra's Ambric architecture.

Figure 6: **Beamforming**



For the sake of simplicity, assume each rectangular block in Figure 6 represents one processor. (In actual implementation, one or more processors may be involved depending on processing performance requirements.) The delay profile, focus weight factor, and other configuration parameters are stored in the external DDR2 memory. For each beamforming process, the characteristics of the beam data are read from the DDR2 memory and streamed through the front end processors. Each processor takes a chunk of data for a particular transducer channel and passes the rest of the data to the downstream processor, one 32-bit word per clock cycle. Each delay handling processor receives data from the ADC and the upstream processor asynchronously. The smart channels handle the flow control naturally: the processor wakes up when data arrives, and is blocked if the channel is empty.

The delay handler outputs time-aligned samples into a random access FIFO (RAFIFO) so the filter can perform interpolation or other processing. The RAFIFO enables the MPPA processors to compute data while they are still in the channel. This is a major performance advantage over DSPs, which incur significant overhead just by moving data in and out of memory before any computation starts. A small set of processors are used at the end for data decimation by summing all the channels to produce a formed beam.

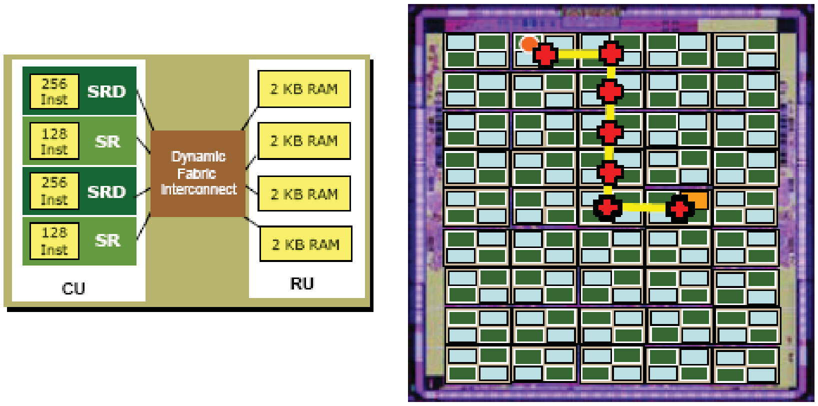
3.4 Deterministic Frame Rate

Any implementation can achieve high frame rates, given sufficient processing power. However, predictable performance is as important as high frame rates for real-time applications. This proves a challenge for any computing architectures using deep pipeline and multi-level cache. Most DSPs require a significant amount of code to set up and tear down the processor so as to take advantage of pipelines and cache. Because the performance behavior is determined at run time and is often data-dependent, it is difficult to predict and plan necessary resources to meet performance requirements. The efforts spent on performance tweaking may easily be wasted once data invalidates prior assumptions.

By contrast, the Am2045's 1 tera-OPS performance is deterministic and predictable in the design phase. The Am2045 processors do not use cache. Processor cores and internal memory modules are highly configurable, and their performance behavior is well-defined. Resources (processors or memory modules) that border each other within a cluster are connected point-to-point by a local fabric (crossbar). Clusters are connected by a global fabric for remote resource access. The latency to "hop" from one cluster to another is well defined, and the communication over the global fabric does not interfere with the local fabric. This communication model is illustrated in Figure 7.

The deterministic nature of the Ambric architecture enables design engineers to take full advantage of Am2045 processing power to achieve predictably high frame rates.

Figure 7: Intracluster and Intercluster Communication



3.5 Empowering Imaging Processing

DSPs are often used to perform the high-level image processing functions necessary after a frame is composed. Differences in computing characteristics require engineers to use different design approaches in solving front-end (ASIC/FPGA) and back-end (DSP) processing problems. This traditional approach requires an engineering resource pool of multiple skill sets, uses multiple tools that are costly to purchase and maintain, and increases the chances of mistakes due to differences in the design methodologies.

Nethra's unique programming model, the high IO bandwidth, and the 50 GMAC DSP processing power make the Am2045 highly effective for both front and back-end ultrasound processing. This unified approach not only enhances design efficiency, it increases design density by enabling the integration of functional blocks that formerly were artificially separated by differences in implementation methodologies.

Compared with a 1 GHz TI C641x DSP, AM2045's throughput is 5 to 25 times higher than TI-published benchmarks, including matrix and FFT operations, with only 1/3 of the code size. The massive number of DSP cores gives Am2045 the ability to handle very sophisticated algorithms. Algorithms that process tiled images can really take advantage of the Am2045's bric architecture by processing multiple tiles in parallel to achieve higher performance.

The Am2045 is highly scalable for future growth. Multiple chips can be connected through the GPIO interface. The programming model can be applied to multiple processors whether they are on the same chip or across several chips. For example, to increase a system from 64 to 128 channels, most of the design effort consists of resource re-partitioning, rather than starting a complete new design as an ASIC/FPGA implementation would require.

3.6 Power Management Efficiency

Efficient power management is very important for portable ultrasound systems. The Am2045's GALS (Globally Asynchronous, Locally Synchronous) architecture gives it a unique efficient power management advantage. GALS enables individual processor clusters to run at widely different clock frequencies, from less than 1.0 MHz to 300 MHz. This enables applications to reduce power consumption by throttling processor speeds according to functional performance profiles.

3.7 Design Tools

IDE

Nethra's Integrated Design Environment (IDE) is a comprehensive software-development environment based on the industry-standard Eclipse environment. Developers describe design structures, implement algorithms in Java or assembly language, and re-use already validated objects from the library.

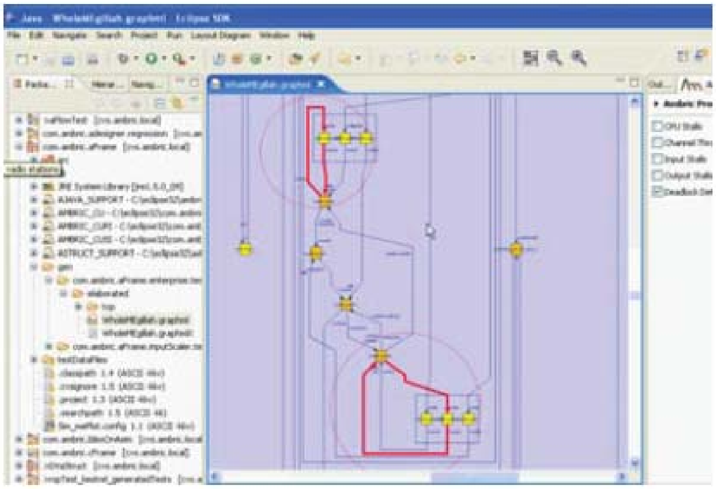
Simulator

The functional simulator verifies a design's functional correctness and its expected execution behavior. Developers have the option of fine-tuning placement and routing constraints for the most efficient use of device resources. Once attributes have been defined, generate the binary file to be downloaded to the Am2045.

Debugger

The visualizer includes a multi-processor debugger and a performance profiler, shown in Figure 8. Set breakpoints, single step, get processor status, and peek or poke data in memory or in FIFOs. The profiler is used to analyze processor usage and communication fabric load. It also detects processor deadlock and starvation. Information is presented in either text or graphical format. Developers can easily navigate the source code throughout the whole debugging and profiling process.

Figure 8: **Multiprocessor Debugger**



4 Conclusion

The Am2045 is a programmable, massively parallel silicon solution that has decided advantages over ASICs, FPGAs, and DSPs; especially in sensor applications involving beamforming and image processing such as medical ultrasound, radar, and sonar.

Notice

NETHRA IMAGING INC.
2855 Bowers Ave.
Santa Clara, CA 95051
Phone: (408) 257-5880
Fax: (408) 855-8601
Email: sales@nethra.us.com
www.nethra.us.com