C6201 DSPs for Medical Ultrasound Beamforming

Rodger H. Hosking,
Vice-president, Pentek, Inc.

System Overview

Two primary needs for medical imaging systems are the ability to perform an enormous number of relatively simple operations and the ability to move vast amounts of data through the processing engine. By using standard, open-architecture COTS products incorporating Texas Instruments 'C6x family of DSPs and high speed analog and digital interfaces, for the first time, medical system designers can construct a system for real-time, 3-D echocardiography imaging.

System requirements for three-dimensional imaging call for sequentially scanning two-dimensional slices of the tissue to form a composite of the whole object. In order to minimize the troublesome effects of patient and transducer movement for these multiple scans, the frame acquisition and calculation time must be extremely short. Performance objectives for this system require 5000 frames per second, with 5000 pixels in each frame. Sixty-four transducers were required to adequately cover the scanning region and provide sufficient image resolution.

Principles of Beamforming

Beamforming techniques are not only used in medical systems but also in speech, sonar, radar, wireless communication, radio telescopes, and even in tactical weapons. In general, beamforming is a technique for strategically combining signals from multiple sources to create an significantly enhanced version of any single source. Beamforming is useful both for receiving signals as well as generating signals, where it is often termed beam steering.

For ultrasound medical imaging, ultrasonic pulses are sent into a patient's tissue and the resulting reflections are detected by an array of transducers. Figure 1 shows a transmitting transducer at the left emitting a pulse of energy. When the pulse encounters a feature of the tissue represented by a change in density, a signal emanates from that feature as if it were a point source.

An array of transducers picks up the reflected signal from the tissue. Each transducer receives its version of the reflection at slightly different times, due to the propagation delay through the intervening tissue. The propagation delay time from a particular point in the tissue to each transducer is designated Tn.

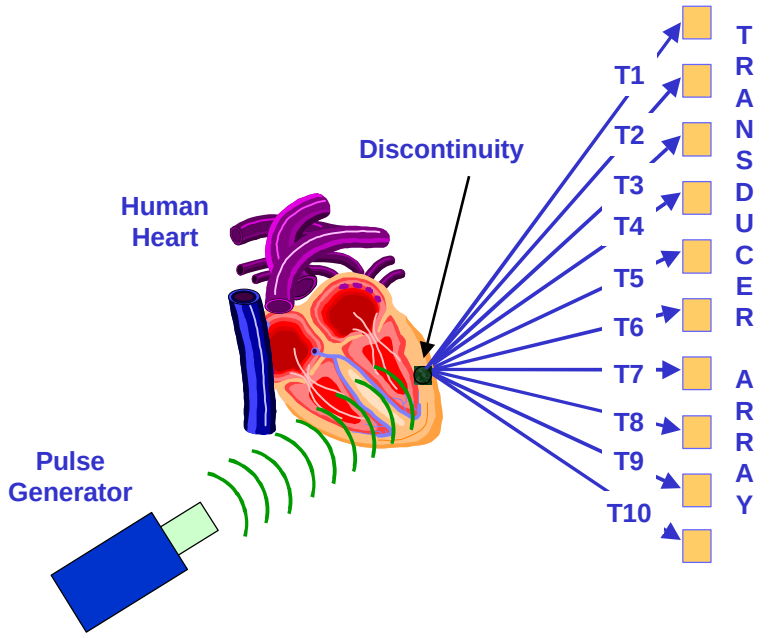

Figure 1. Pulse stimulus causes tissue discontinuity to reflect energy

At the transducer outputs, the signal waveforms from the closest transducer appear first and the other signals follow in time directly proportional to their additional distance from the tissue source point (see figure 2).

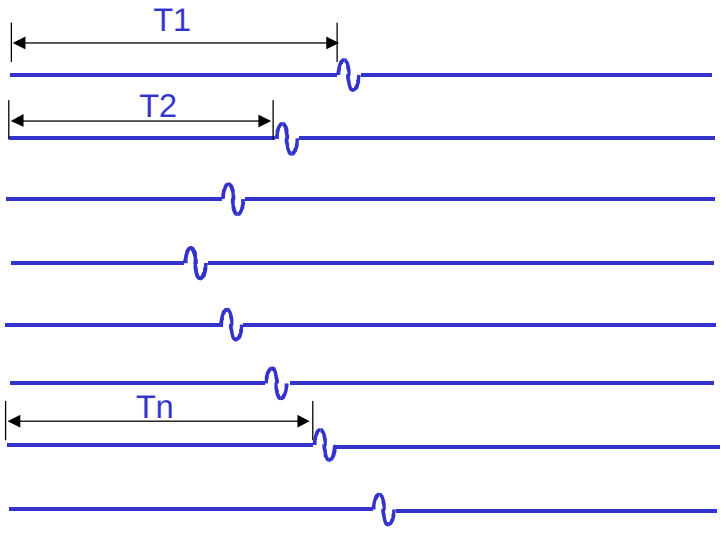

Figure 2. Transducer output pulses with delays proportional to distance

In order to apply beamforming techniques, a delay block is inserted in the path of the signal from each transducer as shown in figure 3. Then the delayed outputs signals are summed to form a single combined output.

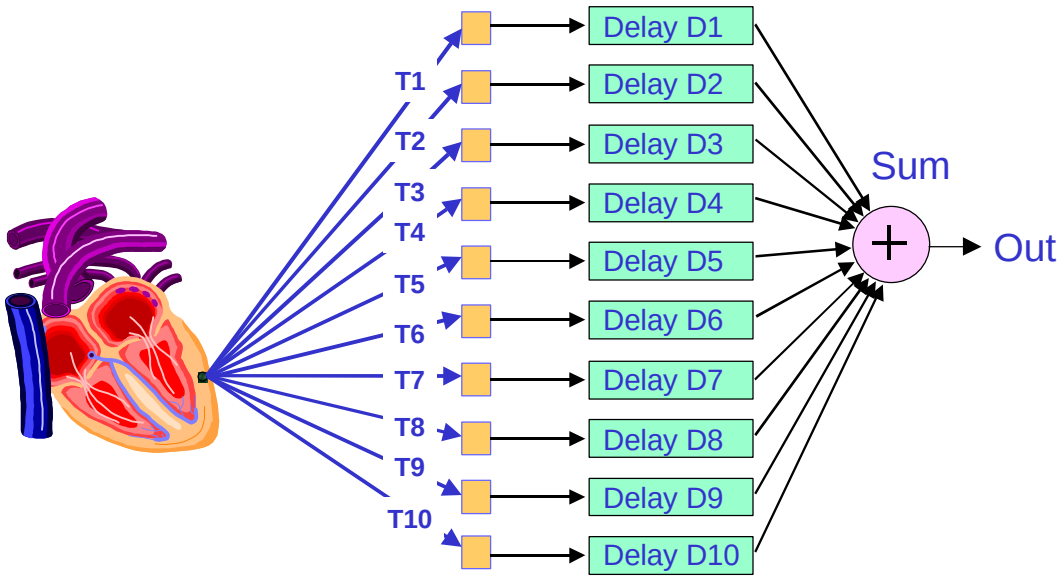

Figure 3. Beamforming system with variable delay generators and summation block

By carefully choosing specific delay coefficients for each of the delay blocks, the delayed pulses can be made to arrive at the sum block simultaneously, adding constructively to form a large output signal as shown in figure 4.

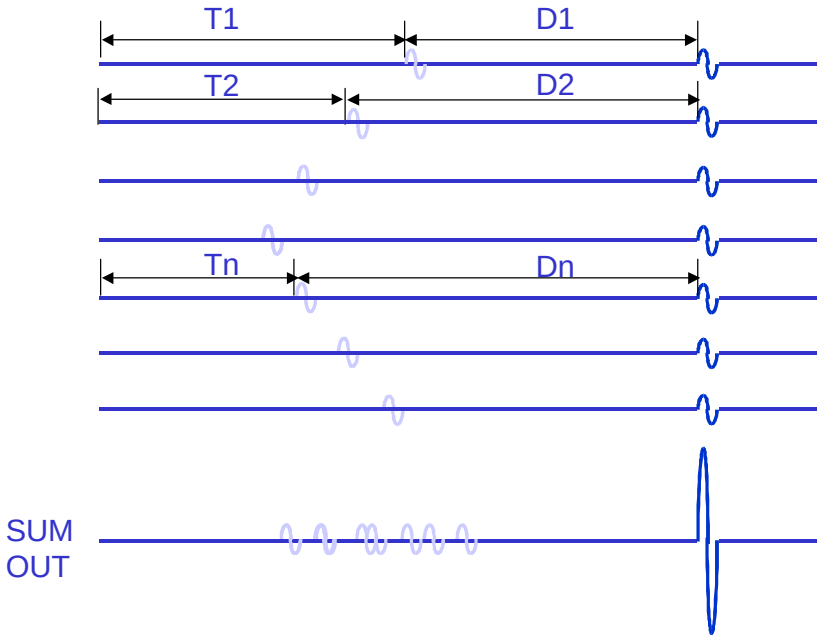

Figure 4. Beamformed (delayed) transducer outputs sum constructively

Note that for each point in the tissue, there is a unique set of Dn delay coefficients for maximizing signal contributions from that one point. Signals originating from other points in the tissue will not be aligned in time and will interfere destructively at the sum output.

To produce a two-dimensional, bit-mapped image of the tissue sample, a stimulus pulse is issued with a particular set of delay coefficients in place for each pixel. The resulting value of each pixel is derived from the energy received at the sum output, thus forming the image by the successive coefficient sets effectively "scanning" the tissue electronically, pixel by pixel.

Mapping these requirements into a DSP-based beamforming system, the block diagram shown in figure 5 evolved. In each of the 64 channels, the transducer signal is first digitized with a high-speed A/D converter. The signal is then sent into a DSP for generating the delay function. Digitized transducer data must be delivered to the DSP in blocks long enough to capture the earliest and the latest reflected pulses from the entire tissue.

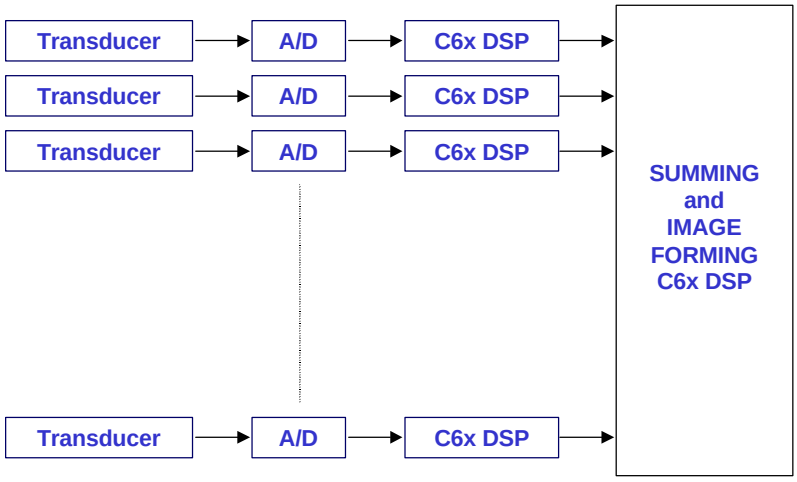

Figure 5. Medical beamforming system using C6x DSPs for delay

Each transducer channel computes its own contribution to each of the 5000 pixels in the image. In order to compute its contribution for a single pixel, the channel processor uses a unique delay coefficient assigned to that channel for that pixel. The pixel delay coefficients between channels need to be different since they are based on the distance between each transducer and the point in the tissue corresponding to that specific pixel.

Each DSP utilizes the stored transducer data table to create a delay generator. Based on the particular delay coefficient for a given pixel, the DSP selects an earlier or later sample in the table, the table location being proportional to the coefficient value. The chosen samples, one from each of the 64 channel processors, are then sent out into the summing DSP for pixel formation.

The same delay calculation and summation process repeats for the each of the 5000 pixels in the image using the **same** transducer samples already stored in memory, but with different delay coefficients. After all pixels have been computed to produce a complete image frame, the system becomes ready for a new pulse and a new image processing cycle. Thus, the data selection and summation time for all pixels in the image defines the frame rate.

Improving Resolution

Not surprisingly, as the time resolution of the delay coefficients increases, so does the optical resolution of the image. Since delay resolution equals the time between A/D samples, one way to improve the image is simply to increase the A/D sampling rate. This impacts three system parameters, all with real-world limits: the signal-to-noise performance of the A/D, the data storage rate, and the size of the sample data table which must be stored in the DSP. As is turns out, attacking the image resolution problem with higher sampling rates is impractical, and DSP's are enlisted to solve the problem.

So far, the DSP has merely served the role of an (expensive) sample selector. By taking advantage of the DSP's computational ability, finely-resolved delays can be implemented using a well known signal processing technique. The complex transducer inputs can be phase-shifted through mathematical angle rotation,

$$I' = I \bullet \cos \Phi - Q \bullet \sin \Phi$$

$$Q' = I \bullet \sin \Phi + Q \bullet \cos \Phi$$

where I and Q are the components of closest "delayed" sample from the stored transducer data table and I' and Q' are the rotated components, with a phase shift by angle Φ. This phase shift calculation is equivalent to interpolating between transducer input samples, producing extremely fine delay resolution.

Now there are two delay coefficients for each pixel in each channel: one for the coarse delay which defines which sample to pick, and the second for the angle of rotation for the fine delay rotation. To save processing time, instead of storing the angle itself, the fine delay coefficient tables store the pre-computed sines and cosines of the phase angles.

Hardware Implementation

Choosing a specific DSP processor and hardware platform requires a rough calculation of the processing load for each DSP. Again, each image frame requires 5000 pixels and the DSP must calculate 5000 frames per second to minimize motion effects. Assuming that one DSP is assigned to each transducer, the DSP must accept the input data samples, calculate that transducer's contribution to each of the 5000 pixels, and deliver the output samples all within one frame period, or 200 μsec.

Each pixel requires a coarse delay table lookup, an input data sample fetch, two fine delay sine/cosine table lookups, four angle multiplications, two additions, and a complex

data store. With approximately 13 operations per pixel, the DSP must handle a minimum of 13 • 5000 = 65000 operations in 200 µsec, or roughly 325 MIPs, not counting any overhead for inefficiencies or managing DMA channels!

In order to meet the resolution requirements, an A/D converter with a minimum of 11-bit accuracy and sampling rate of at least 50 MHz is required. Assuming two bytes per sample, this dictates an input data rate to the DSP of 100 MB/sec plus an output rate of 200 MB/sec for the complex product terms.

Based on the rough calculations above, the Texas Instruments TMS320C6201 DSP chip with 1600 MIPS was selected both for its computational speed and its ability to move data quickly. The Model 4290 Quad 'C6201 processor board family was chosen for its data flow architecture, providing private 400 MB/sec data channels for each of the four processors. These transfer rates are available for both interprocessor communication as well as high-speed I/O, using the Velocity Interface Mezzanine (VIM) modules. Three products in this family comprise the beamforming "node" shown in figure 6, which readily solved two major data flow problems.

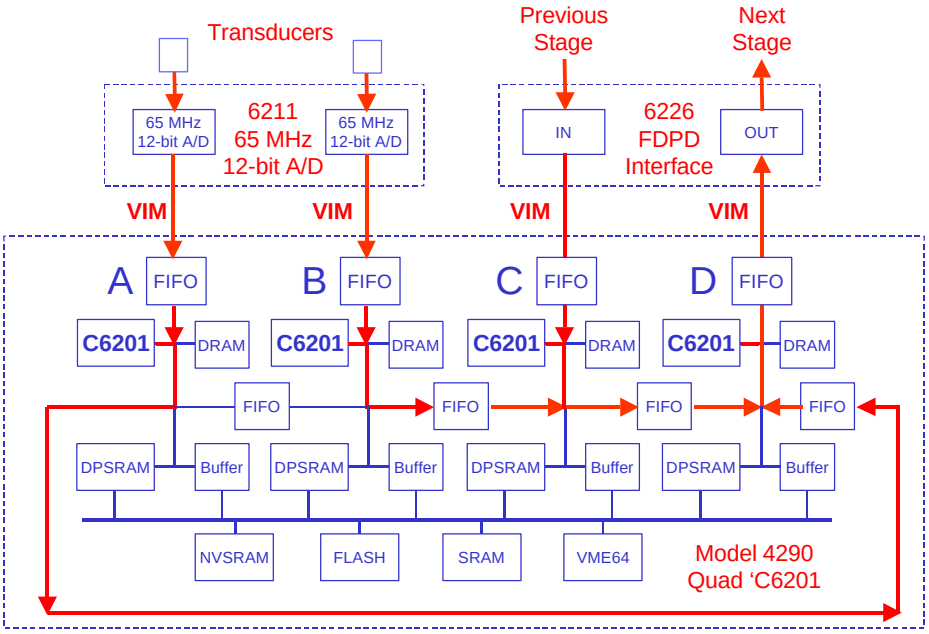

Figure 6. Single-slot quad 'C6201 DSP beamforming node with A/D and FPDP mezzanines. Multiple FIFOs support 1600 MB/sec data transfers.

The first problem was getting a fast enough connection between the A/D converters and the DSPs. This was easily handled with the Model 6211 Dual 65 MHz A/D VIM module. Using an integral data packing engine which arranges two 12-bit samples across a 32-bit word, it supports word transfers at rates as high as 100 MHz for 400 MB/sec maximum throughput. Since the VIM interface includes a synchronous FIFO buffer between the mezzanine and the DSP, the A/D operating at 50 MHz can write to its side of the FIFO at a constant rate of 100 MB/sec. The DSP can unload the other side of the FIFO in

efficient block transfers at 400 MB/sec, for a 25% duty cycle utilization of the processor data bus.

The second data flow problem was sending the output data to a summing processor to combine the contributions from each sensor. As a very effective solution, the other two ‘C6201 processors were harnessed to implement a pipelined summation path using two front panel data ports (FPDPs) for a high-speed I/O chain. The Model 6226 Dual FPDP Interface VIM module attaches to two processors, equipping each of them with a private 160 MB/sec interface to front panel ribbon cable connectors.

In this arrangement, the previous node sends a data block containing the 5000-pixel image with contributions from all of the previous upstream transducers. As this image flows into each processor node, the 5000 pixel contributions from the two transducer channels on that node are added to the image. The image is then forwarded to the next two-channel node for its contributions. The required data rate for the input and output image streams at 5000 frames per second, 5000 pixels per frame and 2 bytes per pixel becomes 50 MB/sec each, well within the 160 MB/sec

Processing Tasks

Within the node, data movement and data processing tasks must be coordinated on a strict time schedule for maximum utilization of the DSP processors. The task list shown in figure 7 shows the major DSP operations and the data movement for each image processing block as a series of numbered time slots.

	Proc A	Proc B	Proc C	Proc D
1	**•A into Mezz FIFO**	**•B into Mezz FIFO**		
2	**•Mezz FIFO to ‘C6x**	**•Mezz FIFO to ‘C6x**		
3	**•Table Look Up**	**•Table Look Up**		
4	**•Angle Lookup**	**•Angle Lookup**	**•Prev into Mezz FIFO**	
5	**•Phase Rotation**	**•Phase Rotation**	**•Read Prev Data**	
6	**•Output to D IP FIFO**	**•Output to C IP FIFO**	**•Wait**	
7	*•A into Mezz FIFO*	*•B into Mezz FIFO*	**•Read From B IP FIFO**	**•Read From A IP FIFO**
8	*•Mezz FIFO to ‘C6x*	*•Mezz FIFO to ‘C6x*	**•Sum Beam B + Prev**	**•Wait**
9	*•Table Look Up*	*•Table Look Up*	**•Sum to D IP FIFO**	**•Wait**
10	*•Angle Lookup*	*•Angle Lookup*	*•Prev into Mezz FIFO*	**•Read From C IP FIFO**
11	*•Phase Rotation*	*•Phase Rotation*	*•Read Prev Data*	**•Sum C + A Beam**
12				**•Output to Mezz FIFO**

Figure 7. Processor task list for basic block computation (bold) with next block shown nested (italics)

Processors A and B operate in parallel, first sending the input block of 10000 data points from the A/D through the VIM mezzanine to the mezzanine FIFOs and then on into internal memory (steps 1 & 2). Data flows into both processors simultaneously over dual, private, dedicated paths with no contention for a shared bus. Refer to figure 6 for these data flow paths.

Processors A and B then use the coarse delay table to locate the correct input sample for the first pixel. The fine delay table is accessed for sine and cosine coefficients used to phase-shift the sample. This process repeats 5000 times, once for each pixel (steps 3, 4, & 5). The coarse and fine delay tables were previously loaded into internal processor memory as part of initialization.

Processors A and B are equipped with private interprocessor FIFOs connecting them directly to processors D and C, respectively. Output data is conveniently buffered in these FIFO's ready to be loaded and unloaded in blocks when convenient (steps 6 & 7). Each output data block consists of 5000 complex, 32-bit samples.

Prior to receiving data from processor B, processor C had retrieved the image data from the previous node over the VIM mezzanine FPDP input port (steps 4 & 5) and now sums the transducer pixels from processor B with the received image (step 8). This combined image is then sent through the interprocessor FIFO to processor D, in another block of 5000 complex samples (steps 9 & 10).

Processor D had previously read the computed pixel contribution from processor A (step 7), and it now adds this to the propagated sum from processor C to form the final output image (step 11). The 5000 pixel output image is delivered to the VIM mezzanine, FIFO where it can propagate to the next stage over the FPDP output port.

Notice that by nesting the task list shown in figure 7, the next processing block can start at time slot 7 instead of time slot 13. This pipelining of tasks within the processing node improves the overall performance of the node by nearly a factor of two, fully exploiting the four simultaneous 400 MB/sec data paths of the Model 4290.

Data Transfer Loading

In this application, data moving occupies a significant fraction of the available bus bandwidth of each processor. The chart below summaries the data transfers required for each 200 μsec processing cycle. The worst case occurs for processors C and D, involved in three data transfers utilizing 62.5% of the bus bandwidth.

	Processor A Bus	**Processor B Bus**	**Processor C Bus**	**Processor D Bus**
VIM Mezzanine FIFO Transfers	Transducer A Data A/D A → Proc A 12- bits (packed 32) 10,000 samples 50 μsec	Transducer B Data A/D B → Proc A 12- bits (packed 32) 10,000 samples 50 μsec	Prev Image Data In FPDP C → Proc C 16-bits (packed 32) 5,000 pixels 25 μsec	Next Image Data Out Proc D → FPDP D 16-bits (packed 32) 5,000 pixels 25 μsec
Interprocessor FIFO Transfers	Transducer A Pixels Proc A → Proc D 32-bits complex 5,000 samples 50 μsec	Transducer B Pixels Proc B → Proc C 32-bits complex 5,000 samples 50 μsec	Transducer B Pixels Proc B → Proc C 32-bits complex 5,000 samples 50 μsec	Transducer A Pixels Proc A → Proc D 32-bits complex 5,000 samples 50 μsec
			Partial Sum Pixels Proc C → Proc D 32-bits complex 5,000 samples 50 μsec	Partial Sum Pixels Proc C → Proc D 32-bits complex 5,000 samples 50 μsec
Totals	100 μsec	100 μsec	125 μsec	125 μsec
Bus Duty Cycle	50%	50%	62.5 %	62.5%

Putting It All Together

The resulting 64-channel system requires 32 identical two-channel nodes connected in a pipelined fashion as shown in figure 8. Each node occupies a single VMEbus slot containing the quad ‘C6201 processor plus the VIM mezzanine modules for the dual A/D and dual FPDP interface, all standard COTS products. Two 21-slot VMEbus card cages each hold 16 processor node assemblies. Connections between adjacent FPDP ports within each cage is made across the front panels with standard flat ribbon cable.

One longer FPDP ribbon cable joins the two cages. The last node delivers completed image frames to a storage and display sub-system, part of the host workstation. Note that no high-speed data flows across the VMEbus backplane, which is reserved

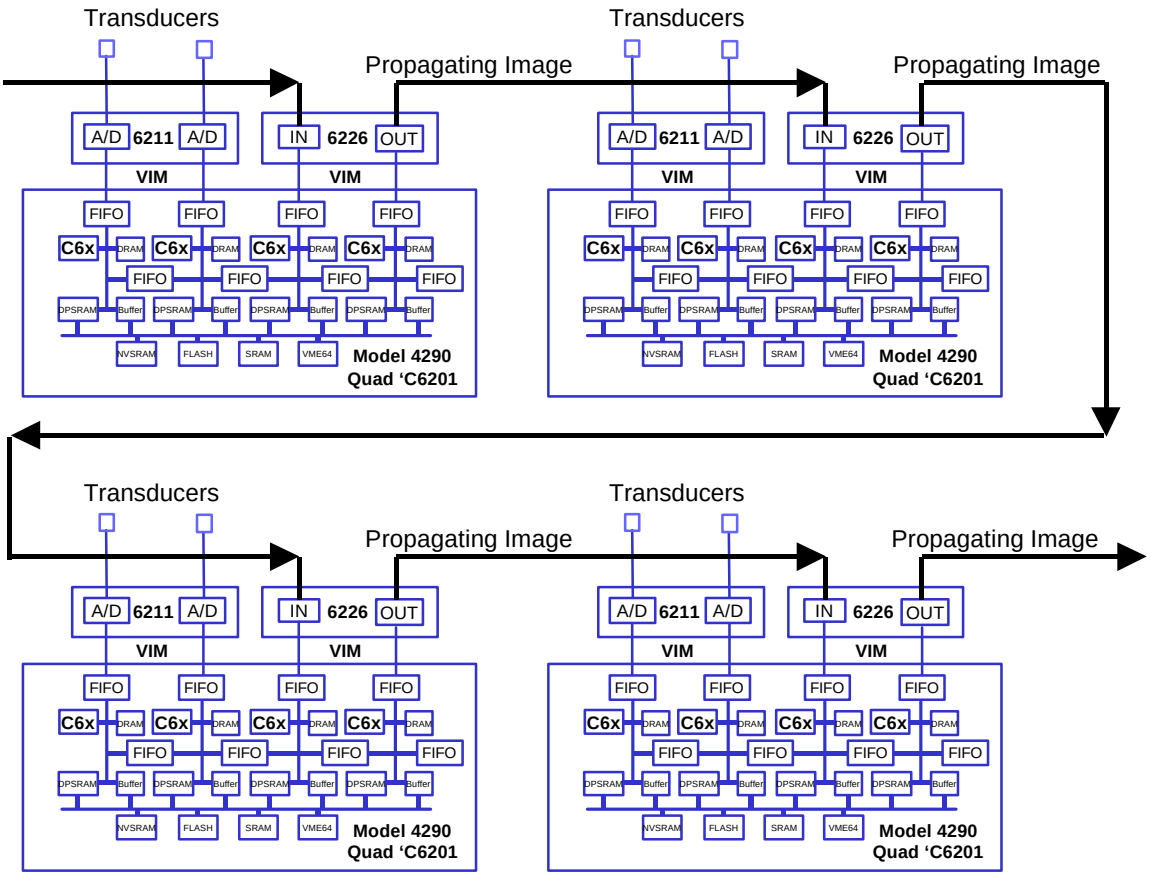

Figure 8. Each beamforming node contributes pixel data from two transducers to the image as it propagates down the pipeline

The highly modular arrangement of performing the beamforming and pixel summation within each node, yields great flexibility in changing the number of transducers in increments of two, simply by adding or removing node processor boards. This allows an easy tradeoff between system cost and image resolution, both of which are directly affected by the number of transducers.

Enabled by the exceptional computational and data moving capabilities of the 'C6x family of processors and the high speed peripheral interfaces afforded by VIM, medical imaging systems will become increasingly more powerful. For the first time, researchers will be able extract previously unattainable and vital information, critical to diagnosing and treating a wide range of conditions.