

# Real-Time Ultrasound Elastography on a Multi-core DSP

Si Luo<sup>1</sup>, Cheoljin Lee<sup>1</sup>, Kerem Karadayi<sup>1</sup> and Yongmin Kim<sup>1,2</sup>

Departments of Electrical Engineering<sup>1</sup> and Bioengineering<sup>2</sup>  
University of Washington  
Seattle, WA 98195 U.S.A.

## Abstract

Ultrasound (US) elastography measures the deformation of tissue in response to applied force to derive and display its stiffness. It has been applied to various organs and tissues, e.g., as an adjunct tool to conventional US and X-ray mammography in the detection and characterization of breast masses. Real-time US elastography processing is demanding due to several computationally-expensive modules, such as crosscorrelation. Multi-core digital signal processors (DSPs) provide increased computational performance with less power consumption and lower cost than other programmable approaches, e.g., with general-purpose CPUs and GPUs. However, optimal algorithm implementations on multi-core processors require careful algorithm/architecture analysis and prudent mapping to achieve good performance. In this paper, we present our approach and techniques used in mapping our US elastography algorithm on a multi-core DSP, i.e., Texas Instruments TMS320C6472. Our results show that a single C6472 multi-core DSP can support real-time US elastography processing.

## I. Introduction

During breast, prostate or thyroid examinations, many clinicians use finger palpation to detect abnormally hard tissues by taking advantage of the fact that tissue stiffness is correlated with pathological changes, e.g., malignant nodules are stiffer than benign nodules and normal tissues. Ultrasound (US) elastography tries to replace finger palpation while achieving higher sensitivity and specificity. It measures the deformation of tissue in response to applied force and derives and displays its stiffness. Over the last two decades, various US elastography methods have been developed, e.g., quasi-static, shear wave, and acoustic radiation force impulse (ARFI) imaging. ARFI imaging involves focusing an ultrasound beam in tissue for a short period of time ( $\sim 0.1$  ms) and measuring the resulting tissue displacements, which are correlated with the local viscoelastic properties [1]. Shear-wave elasticity imaging uses acoustic radiation force to generate shear waves [2]. Since shear waves propagate faster in stiff tissue than in soft tissue, the tissue stiffness can be inferred by measuring shear-wave velocity from US images acquired at a very high frame rate (e.g., 5000 frames/s). Quasi-static elastography is popular as it does not require a special US front-end and can be supported in traditional US systems with software modifications. It estimates tissue strain caused by compression (either transducer-induced or

naturally occurring, e.g., artery pulsation). Stiff tissue has a higher elasticity modulus and undergoes less strain (deformation) under applied force than softer tissue. Thus, by measuring tissue strain induced by compression, the stiffness of tissue can be derived.

US elastography has demonstrated potential for noninvasively differentiating benign and malignant tumors. However, real-time US elastography is computationally demanding, which was one of the reasons for its late commercial introduction. The computational burden of US elastography is mainly attributed to several modules in strain estimation and postprocessing. The most compute-intensive task in strain estimation is crosscorrelation-based search, which estimates tissue displacements between returned ultrasound echo signals before and after compression. In our previous study, Bae et al. [3] developed a new crosscorrelation-based search method, where the computational requirement does not increase with the correlation window size. For typical parameters used in US elastography, the computation time in crosscorrelation-based search was reduced by a factor of more than 30 with this algorithm. As a result, real-time US elastography was enabled with six TMS320C6416 DSPs from Texas Instruments (TI), and the first ultrasound machine (Hi Vision 8500, Hitachi Medical Corp., Tokyo, Japan) supporting real-time elastography was introduced commercially in 2003.

In this paper, we discuss the implementation of real-time US elastography on a single TI TMS320C6472 multi-core DSP. Compared to our previous implementation using multiple single-core DSPs, the new approach based on a multi-core DSP is advantageous in that it simplifies the board design, and lowers power consumption and cost. At the same time, the overall computing power has increased as well. In the following sections, we will first introduce US elastography, its processing and the TI C6472 multi-core DSP architecture briefly. Then, we present how we implemented our algorithm on this multi-core DSP and describe some of the implementation techniques.

## II. US Elastography

Figure 1(a) shows an ultrasound B-mode image of an invasive papillary thyroid carcinoma (pointed by a red arrow). Figure 1(b) shows the corresponding elastography image, which is based on strain computed between a pair of consecutively-acquired US frames (one is a precompression frame while the other is a postcompression frame). A rectangular region-of-interest (ROI) in Fig. 1(b) outlines an area where the strain is calculated. Within the ROI, each pixel is typically assigned to one of 256 colors, depending on the strain magnitude at that location and an RGB color map. One common color map uses red for high strain (i.e., softer tissues) and blue for low strain (i.e., harder tissues), while green indicates average strain. The nodule area in Fig. 1(b) is rendered in blue due to the increased stiffness of this papillary carcinoma. To provide both stiffness and anatomical information, a final elastography image is formed by blending the strain ROI with the B-mode image. Since the carotid artery (pointed by a blue arrow in Fig. 1(a)) is right next to the thyroid gland, we could use carotid artery pulsation

as a compression source (rather than external compression with transducer) to perform thyroid elastography [4].

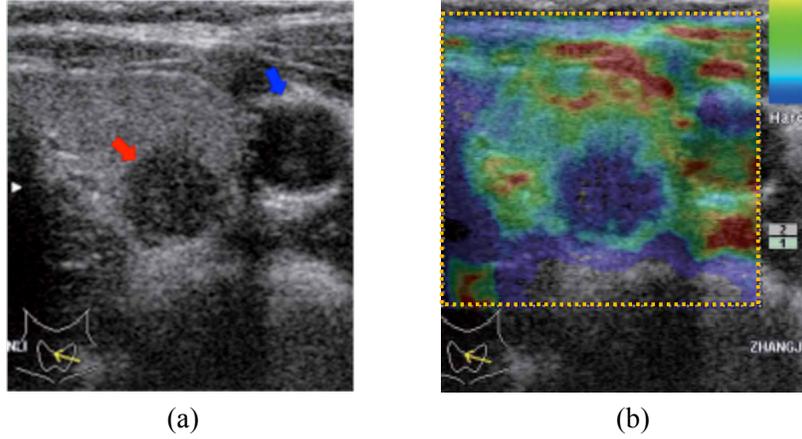


Figure 1. (a) Ultrasound B-mode image and (b) the corresponding elastography image of a papillary thyroid carcinoma.

Strain is calculated based on estimating displacement between US signals acquired before and after compression [5]. The returned US echo from a point scatterer before applying compression can be modeled as

$$x_1(t) = A(t - \tau_1) \cos[\omega_0(t - \tau_1)] \quad (1)$$

where  $\omega_0$  is the US carrier frequency,  $\tau_1$  is the round-trip time from the transducer to the scatterer,  $A$  is the envelope of the received echo. The returned echo after compression can be modeled as

$$x_2(t) = A(t - \tau_2) \cos[\omega_0(t - \tau_2)] \quad (2)$$

The displacement of the point scatterer is related to the difference in round-trip times by

$$d = c(\tau_2 - \tau_1)/2 \quad (3)$$

where  $c$  is the speed of sound in tissue (1540 m/s). Eq. (3) shows that the displacement  $d$  can be calculated by estimating the time delay between precompression and postcompression US signals, which can be obtained by performing crosscorrelation between  $x_1(t)$  and  $x_2(t)$ . Based on the calculated displacement, strain is estimated with a spatial derivative of the displacement as

$$strain = \frac{d_2 - d_1}{L} = \frac{\Delta d}{L} \quad (4)$$

where  $d_1$  and  $d_2$  are the displacement for two different points separated by a distance of  $L$ .

Alternatively, strain can be estimated in phase domain using angular strain estimation

$$strain = \frac{\phi_{\Delta d}}{\phi_L} \quad (5)$$

where  $\phi_{\Delta d}$  is the differential phase difference between two locations and  $\phi_L$  is the phase corresponding to  $L$  [6]. Either radio-frequency (RF) or demodulated baseband data can be used in Eq. (4), while demodulated baseband data are used in Eq. (5).

The simplified flowchart of our elastography processing algorithm is shown in Fig. 2. The precompression and postcompression baseband IQ data from two consecutive frames are used for strain estimation. In case of a frame rate of 30 frames/s, there would be 30 elastography images generated per second. Raw strain is estimated by using the angular strain estimator [6], including crosscorrelation-based search, after which median and boxcar filters are applied to the raw strain image to perform noise reduction and image smoothing. The filtered strain image is blended with the B-mode image during scan conversion to produce the final elastography image as shown in Fig. 1(b).

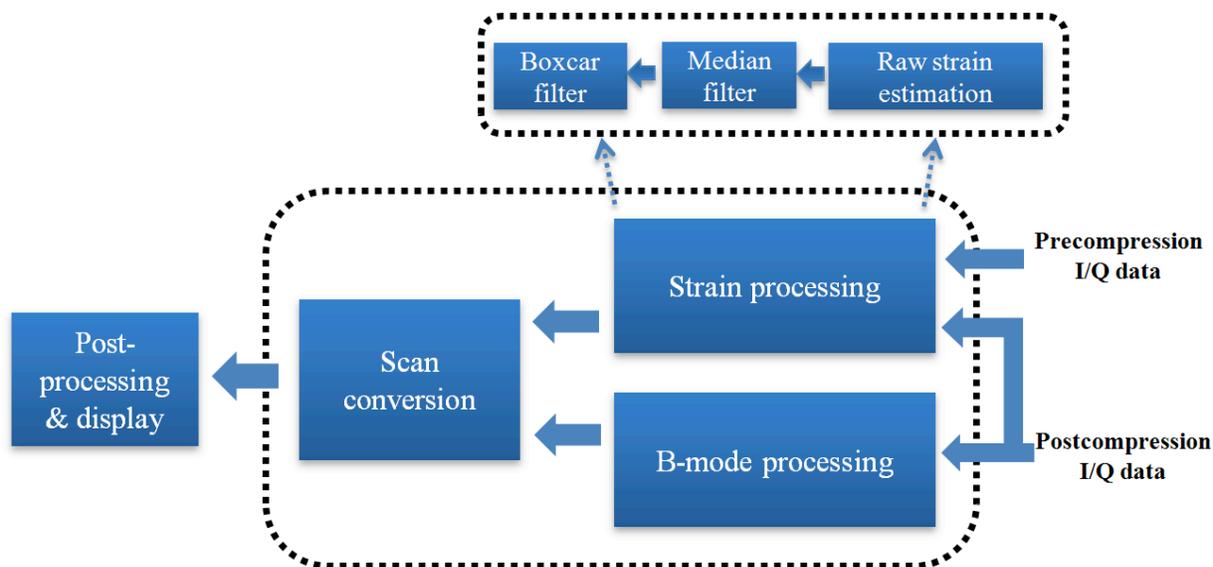


Figure 2. Flowchart of elastography processing.

### III. TMS320C6472 Multi-core DSP Architecture

TMS320C6472 is one of the multi-core DSPs from TI [7]. As can be seen in Fig. 3, it has six C64x+ cores, where cores run at a clock frequency of up to 700 MHz. Each core is a very-long-instruction-word (VLIW) processor with two data paths, each with four functional units (L, S, M and D). The L, S and D units accommodate 32- or 64-bit integer and partitioned arithmetic

and logical operations. In addition, the D unit handles loads and stores of 8-, 16-, 32- or 64-bit data. The M unit supports various multiply and inner-product operations as well as some bitwise operations [7,8].

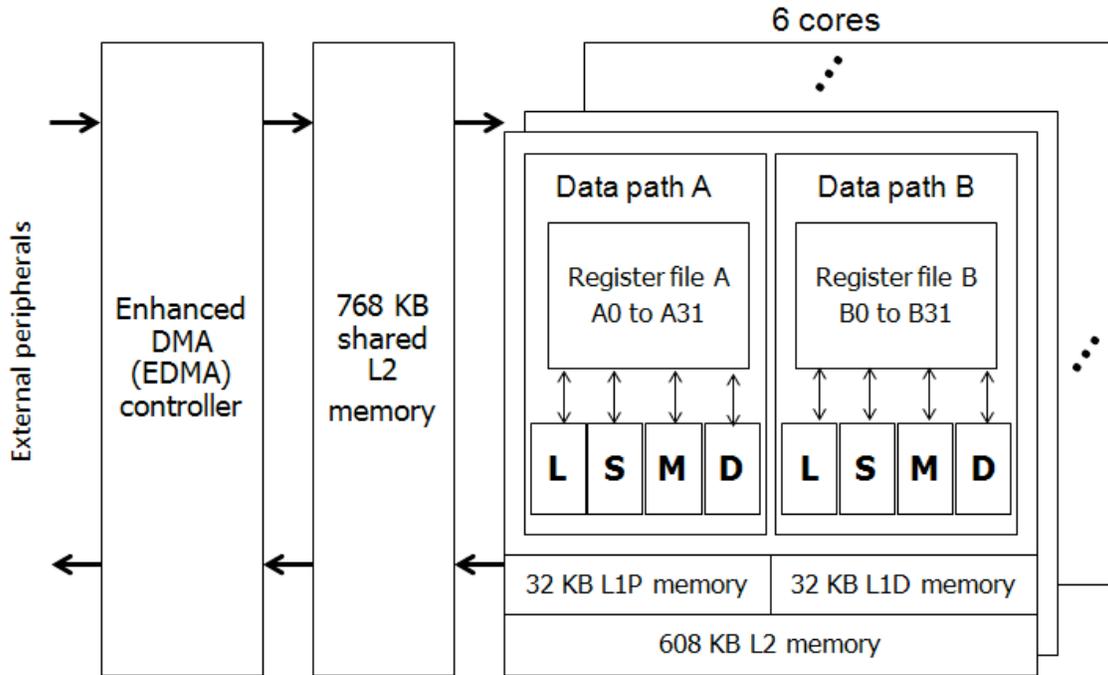


Figure 3. Simplified block diagram of Texas Instruments C6472.

Each core in C6472 has multiple levels of on-chip memory. Closest to the eight functional units in each core are the 32-kbyte Level-1 program (L1P) memory and 32-kbyte Level-1 data (L1D) memory, both of which can be configured as cache and/or addressable SRAM. The next-level memory is called local Level-2 (L2) memory. Each core of C6472 has 608 kbytes of local L2 memory. Similar to L1 memory, L2 can be configured as cache and/or addressable SRAM. In addition, C6472 has additional 768 kbytes of L2 memory shared among all 6 cores.

Furthermore, each core has a programmable internal direct memory access (IDMA) controller that transfers data between L2 and L1 memories concurrently with CPU processing. In addition, there is a separate programmable enhanced direct memory access (EDMA) controller shared among all cores, which allows data transfers across external memory, memory-mapped peripherals and on-chip memory. Programmers can use IDMA and EDMA controllers to hide data transfers behind computation via double buffering to improve performance [9].

## IV. Algorithm Mapping On Multi-core DSP

In a single-core DSP, the single core runs all the processing tasks. In case of a multi-core DSP, we need to partition the processing tasks onto multiple cores and balance the computational load among cores. A simple approach could be frame-based partitioning, where 6 cores perform elastography processing on 6 individual frames as shown in Fig. 4. Each core handles all the elastography processing for a separate frame in a circular manner. For example, core 0 will continue to process frame 6 after completing frame 0. This approach is straightforward to implement, and the load can be well-balanced among multiple cores. However, one drawback with this frame-based partitioning is the latency between the input and output frames. Because it takes a significant amount of time (100 ms or more) for each core to complete the processing of its frame, the operator could perceive a sizable lag between acquisition and display of each frame, which would make the system appear sluggish.

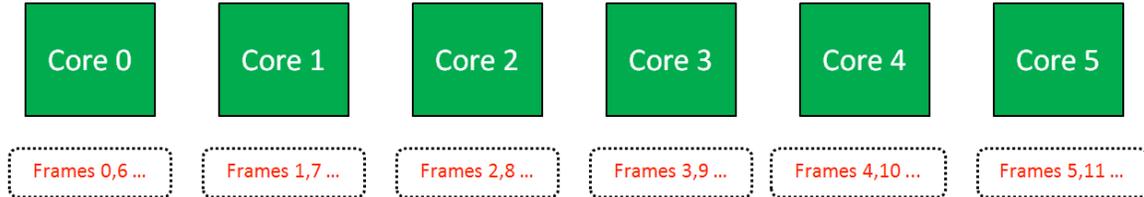


Figure 4. Frame-based partitioning.

Another approach is task-based partitioning, where each core processes a different task and all cores work in a pipelined manner. In order to best utilize the computational power of a multi-core processor, it is important to identify and assign the tasks with a similar amount of computation time to different cores. A simple task-based partitioning example is schematically shown in Fig. 5. For this scheme, the workload for cores 0 to 5 is quite unbalanced because raw strain estimation uses the most computation in elastography processing. It would take much longer for core 0 to process raw strain estimation than other cores to complete their assigned tasks. We can try to divide raw strain estimation into several (e.g., 4) sub-tasks, where all 4 cores process their own sub-task while the remaining 2 cores handle all the other tasks. However, dividing the raw strain estimation task into multiple sub-tasks with comparable computational workload is difficult. For our US elastography algorithm, as a result, we decided to use another approach called data partitioning, where each core performs all the processing tasks on a subset of data as shown in Fig. 6.

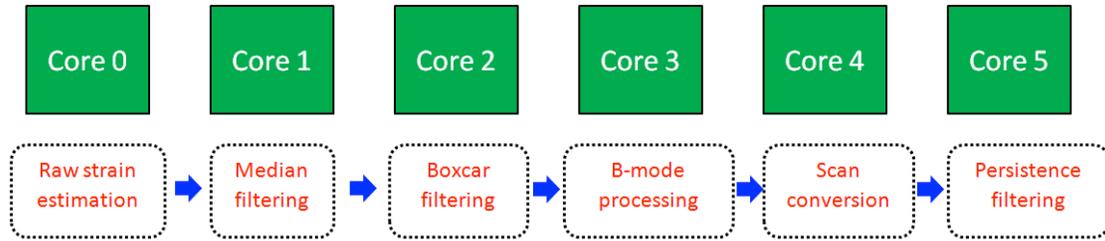


Figure 5. Task-based partitioning.

The advantage of data partitioning over task-based partitioning is that the workload on each core is well-balanced as each core runs the same processing tasks. In elastography processing, however, there are multiple modules that process the input data in a block-by-block manner, e.g., raw strain estimation, median filtering and boxcar filtering, where these modules require additional data at the boundary of each block. As a result, overhead would be incurred because additional efforts are needed to transfer the extra data at the block boundary. In spite of this, we found that data partitioning would be more efficient than task-based partitioning for elastography processing, where a few modules dominate the total computational time.

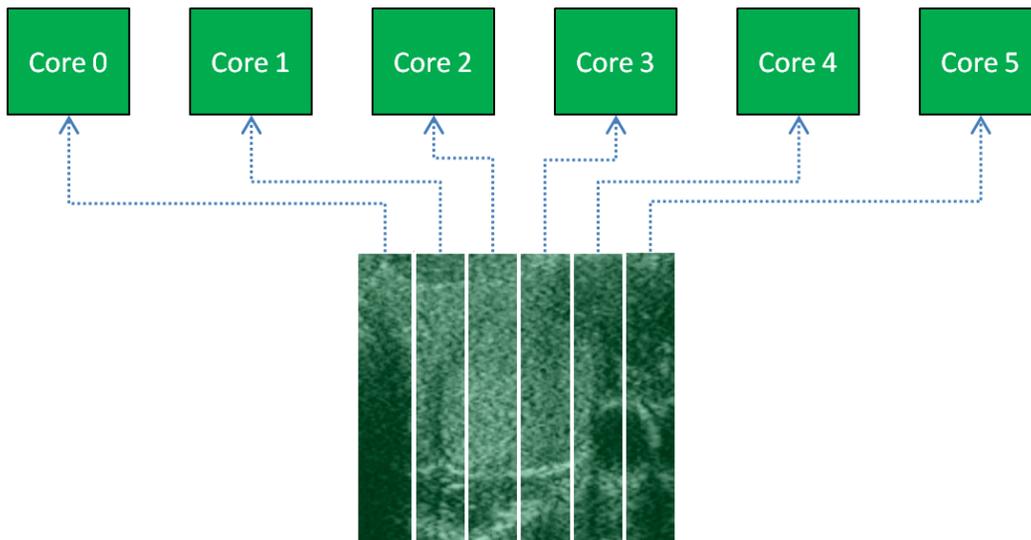


Figure 6. Data partitioning.

For data partitioning, the input image data can be partitioned either vertically or horizontally. By dividing vertically as shown in Fig. 6, all the axial samples in one scanline are transferred and processed by the same core. By dividing horizontally, on the other hand, the axial samples of a scanline would be processed by different cores and then assembled to form a single output scanline. Both vertical and horizontal divisions incur overhead. In ultrasound processing, however, dividing vertically is generally more efficient than dividing horizontally. The axial resolution in ultrasound imaging is much higher than the lateral resolution. Therefore, the kernel

size in the lateral direction is smaller than that in the axial direction, which leads to less overlapped areas in data transfers with a vertical division.

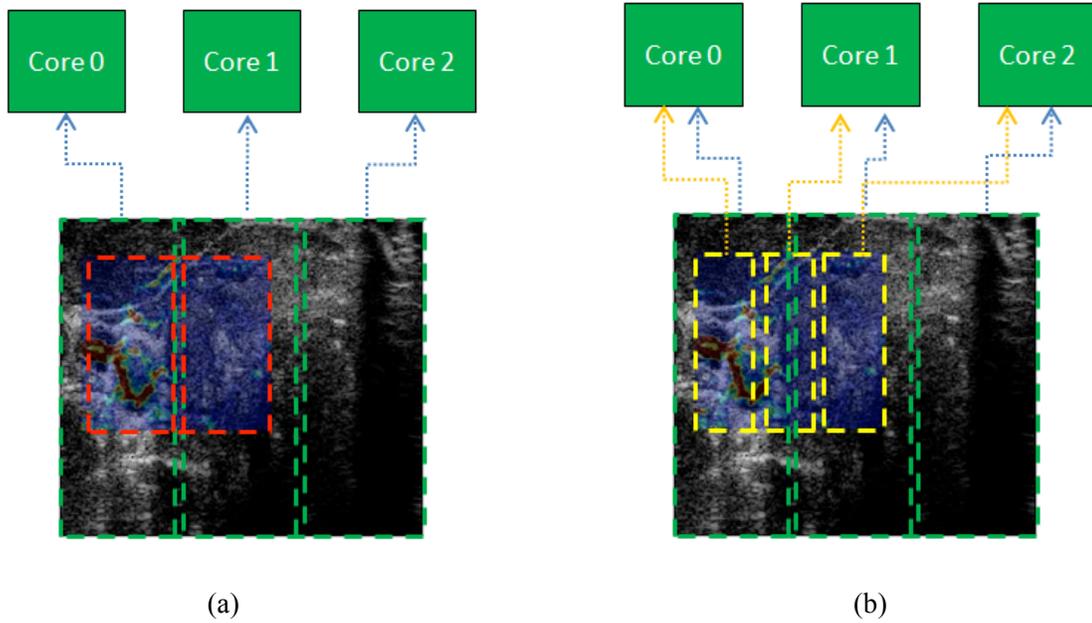


Figure 7. (a) Single-pass and (b) two-pass data transfer. Only 3 cores are shown for simplicity.

Another consideration is how data are transferred from external memory to on-chip memory for processing. Since strain and B-mode processing utilize the same IQ data in our implementation, it is possible to bring the data once and perform both strain and B-mode processing on the same data. Figure 7(a) schematically illustrates this approach. The input data are divided vertically into three data stripes of equal size. Since the strain ROI is smaller than the B-mode image, core 2 would be idle when the other two cores process their portions. For better utilization of computational resources, we decided to bring in the input data for strain and B-mode processing separately. This is shown in Fig. 7(b), where the input data for the strain ROI are brought into on-chip memory for strain processing first. After strain computation, the strain image stripes are transferred to external memory. Then, the input data for B-mode processing are transferred into on-chip memory. Even though this approach requires additional data transfers to move the postcompression data within the strain ROI, the computational efficiency gain from increased processing concurrency by the equal distribution of workload to multiple cores outweighs the cost for these additional data transfers.

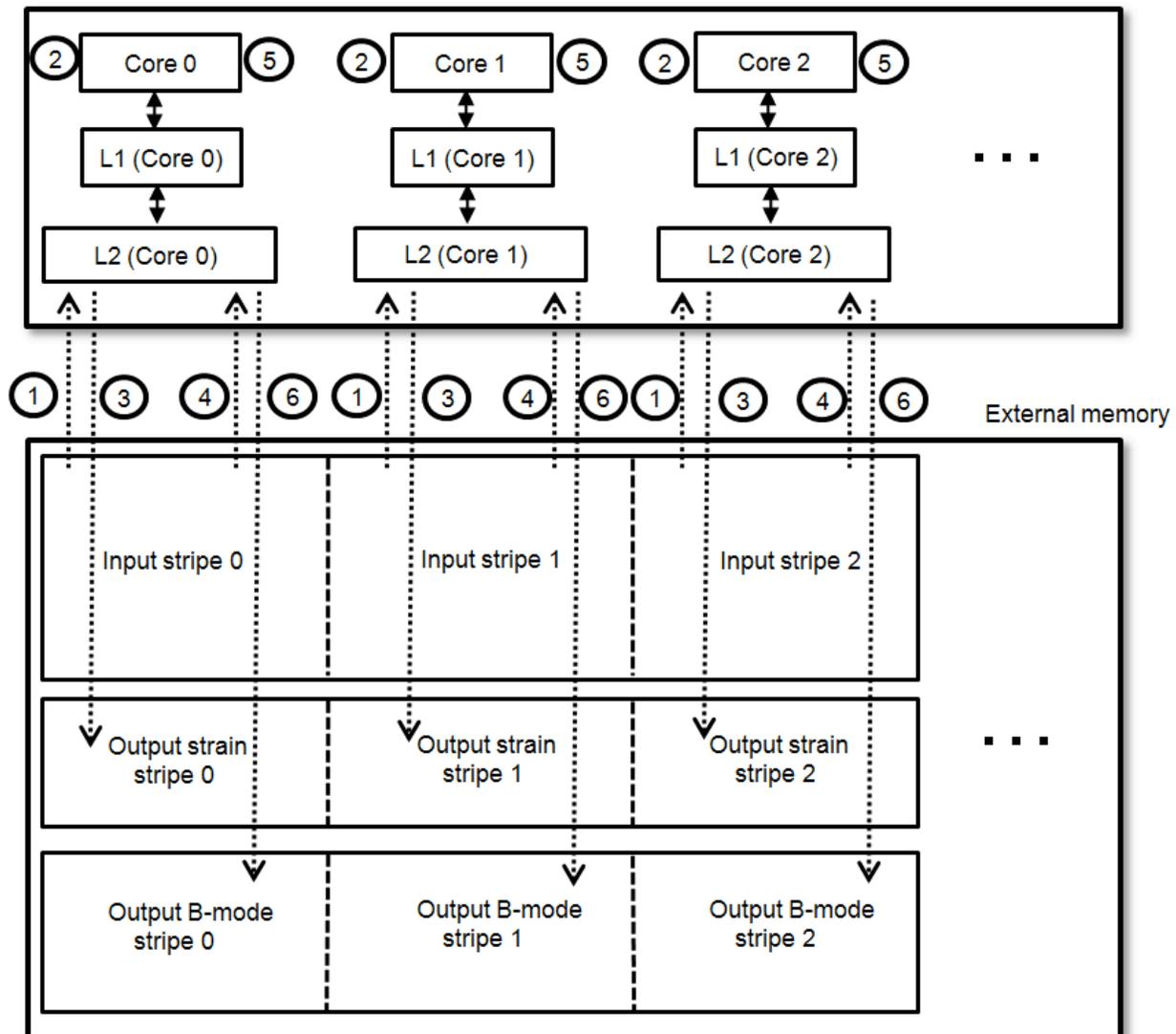


Figure 8. Steps in elastography processing.

Figure 8 summarizes the steps in elastography processing. In step 1, the input data (both precompression and postcompression) for the strain image ROI are brought into the local L2 memory of each core from external memory using EDMA. The L1 memory of each core is configured as cache. After strain processing in each core (step 2), the processed strain stripes are transferred back to external memory to form a strain output image (step 3). In step 4, the postcompression data for B-mode processing (step 5) are transferred into local L2 memory using EDMA, after which the B-mode stripes are assembled to produce a B-mode output image (step 6). As shown in Fig. 2, the strain and B-mode output images are further processed and blended together to produce an elastography image similar to the one in Fig. 1(b).

One of the key techniques used for optimally mapping raw strain estimation on C6472 is the moving correlation method. For strain estimation, 2D crosscorrelation  $R_{xy_{2d}}(i, j, m, n)$  is used to measure the displacement between precompression and postcompression US data

$$\begin{aligned}
R_{xy_{2d}}(i, j, m, n) &= \sum_{h=-\frac{W_2-1}{2}}^{\frac{W_2-1}{2}} \sum_{k=-\frac{W_1-1}{2}}^{\frac{W_1-1}{2}} x(i+h, j+k) \times y^*(i+h+m, j+k+n) \\
&= \sum_{h=-\frac{W_2-1}{2}}^{\frac{W_2-1}{2}} \sum_{k=-\frac{W_1-1}{2}}^{\frac{W_1-1}{2}} r_{xy}(i+h, j+k, m, n) \\
&= \sum_{h=-\frac{W_2-1}{2}}^{\frac{W_2-1}{2}} R_{xy_{1d}}(i+h, j, m, n)
\end{aligned} \tag{6}$$

where  $x$  and  $y$  are the precompression and postcompression US data, respectively,  $i$  is the column index (along the lateral direction),  $j$  is the row index (along the axial direction),  $m$  is the column-wise search index,  $n$  is the row-wise search index,  $W_1$  and  $W_2$  are the row-wise and column-wise correlation window size in terms of the number of samples and they are odd in Eq. (6),  $r_{xy}$  is the point correlation between  $x$  and  $y$ ,  $R_{xy_{1d}}$  is the row-wise 1D crosscorrelation. Eq. (6) indicates that  $R_{xy_{2d}}$  can be computed by first performing 1D row-wise (along the axial direction) crosscorrelation  $R_{xy_{1d}}$  followed by column-wise (along the lateral direction) crosscorrelation. To reduce the number of cycles needed for calculating 2D crosscorrelation, we used the moving correlation method, which is similar to the moving average method [10]. In the moving correlation method, row-wise 1D crosscorrelation  $R_{xy_{1d}}(i, j+1, m, n)$  is computed from the previously-computed row-wise 1D crosscorrelation  $R_{xy_{1d}}(i, j, m, n)$  by adding the new point correlation value entering the correlation window ( $p$ ) and subtracting the point correlation value leaving the correlation window ( $q$ ) as shown in Fig. 9.  $p$  is computed with a complex multiplication between precompression and postcompression data as indicated in Eq. (6), and  $q$  is available since it was previously computed. Using this approach, once the first row-wise 1D crosscorrelation  $R_{xy_{1d}}(i, 0, m, n)$  is computed,  $R_{xy_{1d}}(i, j+1, m, n)$  can be computed with one complex multiplication, one addition and one subtraction.

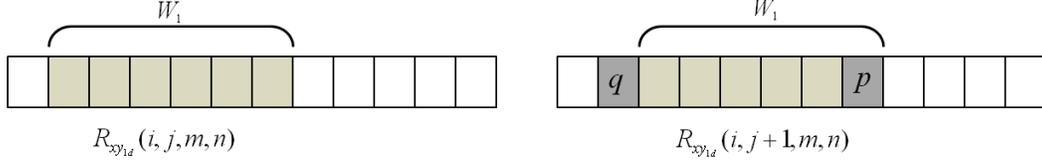


Figure 9. Computation of row-wise 1D crosscorrelation using moving correlation.

Once row-wise 1D crosscorrelation has been completed, column-wise crosscorrelation can begin. Similar to row-wise crosscorrelation, we can compute the next column-wise crosscorrelation based on the previously-computed column-wise crosscorrelation as shown in Fig. 10 with one addition and one subtraction.

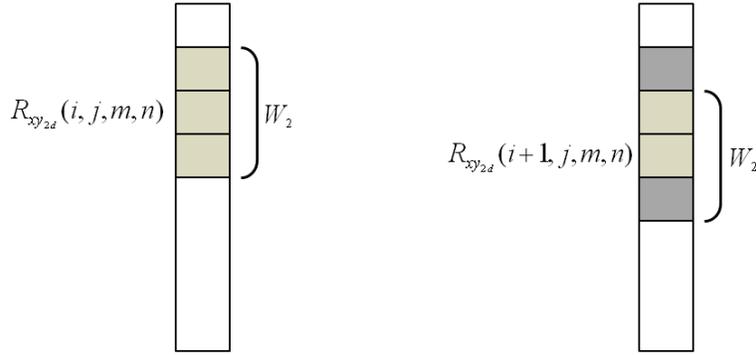


Figure 10. Computation of column-wise crosscorrelation using moving correlation.

Utilizing the moving correlation method in both row and column directions, the computational requirement for calculating one 2D crosscorrelation is reduced from  $W_1 \times W_2$  complex multiplications and  $W_1 \times W_2 - 1$  additions/subtractions to 1 complex multiplication and 4 additions/subtractions. In addition, the computational load of the moving correlation method does not depend on the correlation window size,  $W_1$  and  $W_2$ . For a strain image size of  $100 \times 100$ , a correlation window size of 5 ( $W_2$ : lateral)  $\times$  21 ( $W_1$ : axial) and a search range of 3 (lateral)  $\times$  40 (axial), the computational requirement for 2D crosscorrelation was reduced from 126M complex multiplications ( $100 \times 100 \times 5 \times 21 \times 3 \times 40$ ) and 125M ( $100 \times 100 \times (5 \times 21 - 1) \times 3 \times 40$ ) additions/subtractions to 1.2M ( $100 \times 100 \times 3 \times 40$ ) complex multiplications and 4.8M ( $100 \times 100 \times 3 \times 40 \times 4$ ) additions/subtractions with the moving correlation method. More detailed description of these low-level optimization techniques can be found in [3].

## V. Results

Table 1 lists the execution time of individual modules for an input size of 358 (scanlines)  $\times$  384 (samples) and a strain ROI size of  $224 \times 224$ . The correlation window size was 3 ( $W_2$ )  $\times$  13

( $W_1$ ), while the crosscorrelation search range was  $3 \times 18$ . The kernel sizes of 7 (lateral)  $\times$  7 (axial) and 5 (lateral)  $\times$  9 (axial) were used for boxcar and median filtering, respectively. With a  $224 \times 224$  strain ROI, the total execution time is 16.9 ms/frame on a C6472 running at 700 MHz, which corresponds to 59 frames/s. The total execution time is generally proportional to the strain ROI size. With a strain ROI size of  $288 \times 288$ , the total execution time increases to 26.3 ms/frame as shown in Table 2, corresponding to 38 frames/s. These results show that a single 700-MHz C6472 multi-core DSP can support real-time US elastography processing including B-mode processing and scan conversion. As can be seen in Tables 1 and 2, the strain processing module is most computationally-expensive. According to Fig. 2, this strain processing module consists of three components, i.e., raw strain estimation, median and boxcar filtering. Table 3 shows the breakdown of 17.5 ms strain processing time from Table 2. The raw strain estimation occupies 80% (14.0 ms/17.5 ms) of the total execution time for strain processing.

Table 1. Performance on a C6472@700 MHz with  $358 \times 384$  B-mode image and  $224 \times 224$  strain ROI sizes.

Module	Execution time (ms)
Strain processing	9.5
B-mode processing	1.8
Persistence filtering	0.7
Scan conversion (strain and B-mode images)	4.9
Total	16.9

Table 2. Performance on a C6472@700 MHz with  $358 \times 384$  B-mode image and  $288 \times 288$  strain ROI sizes.

Module	Execution time (ms)
Strain processing	17.5
B-mode processing	2.0
Persistence filtering	1.9
Scan conversion (strain and B-mode images)	4.9
Total	26.3

Table 3. Strain processing performance on a C6472@700 MHz with a  $288 \times 288$  strain ROI size.

Module	Execution time (ms)
Raw strain estimation	14.0
Median filtering	3.4
Boxcar filtering	0.1
Total	17.5

## VI. Discussion

In this white paper, we presented how we mapped our US elastography algorithm on a TMS320C6472 multi-core DSP. Because strain processing is the most computationally-expensive task, it is essential that multiple cores work collaboratively on strain processing for efficient computation. For better utilization of computational resources, we used a data partitioning approach rather than task-based partitioning. Also, we used a vertical division strategy to minimize overhead incurred by data partitioning. Finally, we used a two-pass data transfer scheme so that each core participates in strain and B-mode processing equally although it requires additional data movement between on-chip and external memory.

The raw strain estimation is the most time-consuming task in strain processing. We can evaluate our previous single-core and new multi-core implementations for raw strain estimation. The execution time for raw strain estimation was 94.2 ms/frame on a single-core C6416 as listed in Table 4, compared to 14 ms/frame on a C6472 in Table 3. The multi-core implementation has a larger output strain ROI than that of the single-core implementation. On the other hand, C6472 is running at 700 MHz while C6416 was running at 500 MHz. It is difficult to directly compare these two implementations and performance due to differences in strain ROI size, window and kernel sizes, and clock frequency. Nonetheless, our experience indicates that a significant performance improvement was obtained in C6472 mainly due to the presence of six cores in a single chip.

Table 4. Strain processing performance on a C6416@500 MHz with a  $108 \times 154$  strain ROI size.

Module	Execution time (ms)
Raw strain estimation	94.2
Median filtering	18.1
Boxcar filtering	0.3
Total	112.6

US elastography has shown potential for various clinical applications, e.g., noninvasively differentiating benign and malignant breast tumors, visualizing thermal lesions after radio-frequency ablation on tumor, or detecting vulnerable plaques in arteries. In our previous studies, we demonstrated that US elastography could potentially reduce the number of fine needle aspiration (FNA) biopsies performed on thyroid nodules by about 60% [11, 12]. Thyroid nodules are a common medical problem, with studies reporting as high as 50% of the population having a thyroid nodule at autopsy. Currently, ultrasound is used to detect the presence of thyroid nodules, assess their characteristics for suspicious features, obtain measurements and

guide the FNA biopsies. However, no single US feature is highly predictive of malignancy, and ultrasound imaging tends to give a low specificity value, resulting in many benign nodules being referred to an FNA biopsy. Thus, the majority of FNA procedures (250,000 annually in the U.S. each costing about \$1,500) are performed on benign nodules. If we can detect many benign nodules and remove them from an FNA biopsy altogether, the number of FNA biopsies performed on benign nodules will be significantly reduced, thus improving the quality of patient care. Also, costs associated with FNA biopsies on patients with benign nodules could decrease substantially. US elastography could provide more diagnostically useful information to physicians and increase their level of confidence in deciding which nodules should be referred for an FNA biopsy. With the availability of lower-cost and easy to use US machines (e.g., low-to-mid-end or portable), US elastography could be used by endocrinologists or primary care doctors in addition to radiologists for managing numerous thyroid nodules that are being detected around the world.

Currently, real-time US elastography is available in high-end US machines, e.g., Philips iU22, GE LOGIQ E9, and Hitachi Preirus. It is not available in low- and mid-end or portable US systems. Our results show that a single TMS320C6472 multi-core DSP can support real-time US elastography. It should be possible to support B-mode, spectral Doppler, color Doppler and elastography processing with a single multi-core DSP. In the future, it might be possible to support even front-end beamforming in the same multi-core DSP. Such an US system could be low-cost, compact, efficient in power use and flexible for supporting new clinical applications and algorithms.

## References

- [1] K. Nightingale, M. S. Soo, R. Nightingale and G. Trahey, "Acoustic radiation force impulse imaging: in vivo demonstration of clinical feasibility," *Ultrasound Med Biol*, vol. 28, pp. 227-235, 2002.
- [2] J. Bercoff, M. Tanter and M. Fink, "Supersonic shear imaging: a new technique for soft tissue elasticity mapping," *IEEE Trans Ultrason Ferroelectr Freq Contr*, vol. 51, pp. 396-409, 2004.
- [3] U. Bae and Y. Kim, "Real-time ultrasound elastography," *Proc SPIE*, vol. 6513, 65130J; doi:10.1117/12.710130, 2007.
- [4] U. Bae, M. Dighe, T. Dubinsky, S. Minoshima, V. Shamdasani and Y. Kim, "Ultrasound thyroid elastography using carotid artery pulsation: Preliminary study," *J Ultrasound Med*, vol. 26, pp. 797-805, 2007.

- [5] M. O'Donnell, A. R. Skovoroda, B. M. Shapo and S. Y. Emelianov, "Internal displacement and strain imaging using ultrasonic speckle tracking," *IEEE Trans Ultrason Ferroelectr Freq Contr*, vol. 41, pp. 314-325, 1994.
- [6] U. Bae and Y. Kim, "Angular strain estimation method for elastography," *IEEE Trans Ultrason Ferroelectr Freq Contr*, vol. 54, pp. 2653-2661, 2007.
- [7] TMS320C6472 Fixed-Point Digital Signal Processor. Texas Instruments, Literature Number SPRS612F, February 2011.
- [8] K. Karadayi, V. Markandey, R. J. Gove and Y. Kim, "Strategies for mapping algorithms to mediaprocessors for high performance," *IEEE Micro*, vol. 23, no. 4, pp. 58-70, 2003.
- [9] D. Kim, R. Managuli and Y. Kim, "Data cache and direct memory access in programming mediaprocessors," *IEEE Micro*, vol. 21, no. 4, pp. 33-42, 2001.
- [10] U. Bae, V. Shamdasani, R. Managuli and Y. Kim, "Fast adaptive unsharp masking with programmable mediaprocessors," *J Digital Imaging*, vol. 16, pp. 230-239, 2003.
- [11] M. Dighe, J. Kim, S. Luo and Y. Kim, "Utility of the Ultrasound elastographic systolic thyroid stiffness index in reducing fine-needle aspirations," *J Ultrasound Med*, vol. 29, pp. 565-574, 2010.
- [12] S. Luo, E. H. Kim, M. Dighe and Y. Kim, "Thyroid nodule classification using ultrasound elastography via linear discriminant analysis," *Ultrasonics*, vol. 51, pp. 425-431, 2011.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Transportation and Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2011, Texas Instruments Incorporated