Application Report

SPRAB32 – March 2009

Ultrasound Scan Conversion on TI's C64x+ DSPs

Xiaohui Li

ABSTRACT

One of the recent significant developments in ultrasound is the emergence of portable and handheld ultrasound machines and their rapid acceptance in the market place. Because of their power efficiency and high performance, digital signal processor (DSP) based devices have been increasingly used as the main processing engine in these portable and hand carried units.

This application report discusses major requirements and key challenges of implementing highly efficient scan conversion on DSP, as well as the key features and major design considerations for achieving high efficiency. It introduces TI's efficient scan conversion software implementation for TI C64x+™ based DSP devices. Performances are provided, in terms of cycles per pixel and total number of MHz consumed by the scan converter.

Contents

List of Figures

List of Tables

1 Introduction

A well designed DSP scan conversion software is efficient, accurate, and able to provide all the necessary functionalities. Because scan conversion requires a significant amount of processing power, input/output (I/O) bandwidth, and high precision to maintain accuracy, the software for core processing needs to be highly optimized, I/O needs to be carefully managed with respect to possible performance impact, and the precisions need to be maintained in order to have accurate results. If any of these are compromised, the overall performance could suffer.

Highly efficient scan conversion DSP software has been implemented on TI's C64x+-based DSP devices at Texas Instruments. The DSP software is designed to achieve maximum efficiency while maintaining accuracy. It is aimed to enable ultrasound designers to design portable and hand carried ultrasound machines using TI C64x+-based DSP devices. The entire scan conversion source code for the DSP, including detailed documentation and test benches, is available as part of the TI Embedded Processor Software Toolkit for Medical Diagnostic Ultrasound (STD-MED.)

To keep system size, cost, and power to a minimum in portable ultrasound systems, it is necessary to efficiently work within a limited amount of on-chip memory. To achieve this, it is important to have an efficient and optimized implementation of key processing blocks. Scan conversion, one of the last major signal processing blocks in an ultrasound system, is responsible for converting ultrasound images from captured coordinates to Cartesian coordinates for final display in both B-mode and color flow images. Scan conversion also combines B-mode and color flow images, if both images are present, and performs color mapping for final display.

This application report discusses the challenges of implementing scan conversion on a DSP. It introduces TI's scan conversion implementation and discusses the design considerations for key issues such as performance, I/O, and accuracy. The performance for an example configuration of the scan conversion is provided.

2 Scan Conversion Implementation on DSP

2.1 Overview of Scan Conversion

The purpose of scan conversion in a digital ultrasound application is to translate input data that are captured in different coordinates into Cartesian coordinates, which are more suitable for display. In an ultrasound system, the input to the scan converter is the scanned echo data or color flow data (velocity and turbulence). The output is typically data that needs to be displayed on a monitor, such as an LCD screen. More information on scan conversion for ultrasound application can be found in [1] [2] [3].

Figure 1 shows a functional block diagram of a scan conversion sub-system. The system consists of three major blocks: scan conversion, tissue/flow arbitration, and color mapping. The majority of processing in scan conversion is transforming data specified in input geometry (such as a phased array sector) to Cartesian coordinates of the specified display window size. The conversion can either be for B-mode scan conversion, color scan conversion, or both. In addition, aliasing detection and correction are required in color scan conversion for color flow image. If both B-mode and color flow image are present, scan conversion is also responsible for tissue/flow arbitration and color mapping. Tissue/flow arbitration is the decision to output a pixel either from B-mode image or from color flow image. Color mapping is the mapping of a pixel from either B-mode or color flow image to a particular color. Finally, the output is formulated to a certain video format such as 422 for final display.

C64x+ is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

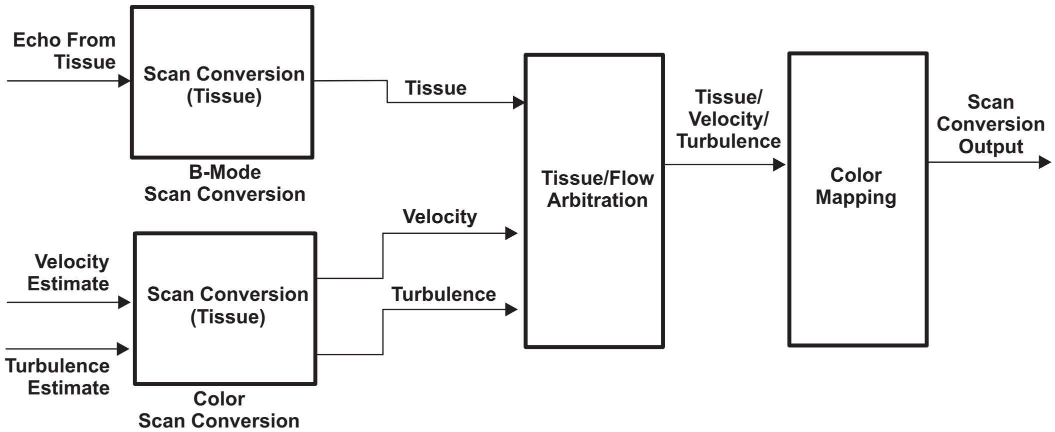

Figure 1. Scan Conversion System Block Diagram

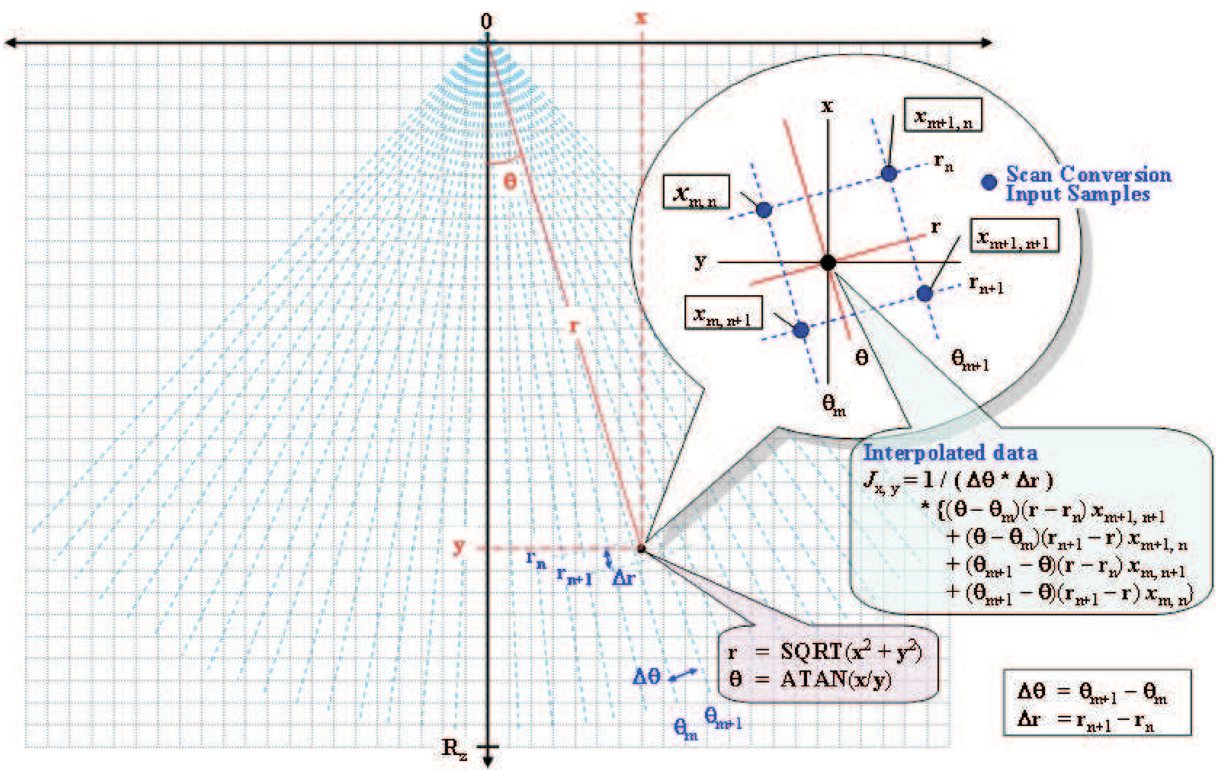

Figure 2. Scan Conversion Using Bilinear Interpolation for Sectored Image

A commonly used interpolation method for scan conversion is bilinear interpolation. Figure 2 shows scan conversion using bilinear interpolation for the sectored image. The underlining principle for scan conversion for linear scan images using bilinear interpolation is similar. The basic steps for scan conversion to generate a desired sample can be summarized in the following steps.

1. Identify the neighboring four input pixels that reside on the neighboring two scan lines.
2. Compute the four interpolation coefficients.
3. Interpolate the desired sample using the four neighboring pixels and the computed interpolation coefficients.

In addition to bilinear interpolation, color scan conversion for color flow data also needs to do aliasing detection and mitigation. In color flow imaging, the velocity data are the estimates of flow velocity as a function of position. The velocities are estimated from the correlation values between pairs of acoustic pulses reflected from the sample region. This measured velocity is limited by the pulse rate frequency in accordance with the Nyquist sampling theorem. This limitation can cause an aliasing problem in the estimated velocity. In regions of large flow velocity, the flow velocity can appear to suddenly reverse direction as a result of this aliasing problem. The purpose of the aliasing detection and compensation is to detect if any such aliasing happens in the estimated flow velocity data and to compensate for the aliasing error.

2.2 Key Requirements for DSP Based Scan Conversion

2.2.1 Application Programming Interface (API)

A DSP-based software implementation of scan conversion should have a well defined API. Through the API, you can accurately specify all the physical parameters related with the scan image, such as the number of scan lines, the number of samples per scan line, the sector angle in the case of phased array, the depth of the scan line, etc.

In addition, you should be able to specify the region of interest (ROI) for the scan conversion, such as the size of the window in terms of the physical length in both vertical and horizontal directions and the starting physical location of the window. You can specify the parameters to map the ROI to the display window for actual display.

You can also configure for different mode of operations such as B-mode conversion and/or color scan conversion.

2.2.2 Features

A DSP-based software scan conversion implementation can support rich features that are provided by scan converters based on higher power technologies such as a field programmable gate array (FPGA) or a printed circuit (PC). The following is a list of the features and capabilities that a scan converter should provide.

- Phased Array and Linear Array Image

The DSP-based scan conversion should be able to process images based on both a phased array and a linear array scan head. It should have the capability and flexibility in supporting different ROIs. For example, it should be able to support various sector angles in the case of a phased array, or slanted angles in the case of a linear array. It should also be flexible in terms of supporting different numbers of scan lines, different numbers of samples per scan line, and different scan line depth.

- B-Mode Scan Conversion

The DSP-based scan conversion should be able to process B-mode images.

- Color Scan Conversion

The DSP-based scan conversion should be able to operate in color scan conversion mode.

- Tissue and Flow Arbitration and Color Mapping

DSP-based scan conversion should also be capable of supporting tissue and flow arbitration and color mapping.

2.2.3 Flexibility

The DSP-based scan converter should be flexible in terms of having the ability to operate in different modes.

For example, the scan converter should be able to be configured to process B-mode only images or to process both B-mode and color images. The scan converter should be able to support arbitration between tissue and color flow and perform color mapping, but also give you the flexibility to implement your own such scheme if you choose to do so.

2.2.4 Efficiency

The DSP-based scan conversion should be highly efficient so that minimum DSP CPU bandwidth is consumed for scan conversion operation.

2.3 Key Challenges of Implementing Scan Conversion on DSP

There are several challenges in implementing highly efficient scan conversion code on a DSP as listed in the following sections.

2.3.1 High Computational Requirement:

The first major challenge of implementing highly efficient scan conversion is its high computational requirement. The processing requirement is especially high for ultrasound images based on phased array scan heads. The scan conversion is usually based on 2x2 or 4x2 bilinear interpolation. This means that for every pixel that needs to be generated by the scan converter, the scan converter needs to determine the addresses of its 4 or 8 neighboring pixels for its inputs. Then, the scan converter needs to compute the interpolation coefficients based on the distances between the pixel to be generated and its neighboring input pixels. Finally, the scan converter interpolates the pixel using the newly computed interpolation coefficients and its neighboring pixels.

The computations mentioned above, that are required for each pixel to be generated, include trigonometric and square root computations, and most of these computations require high precision to achieve high accuracy. The amount of computations for every pixel is significant; therefore, it is important to design a scan converter that can carry out all these computations accurately and efficiently.

Table 1 summarizes the DSP cycle of performances for computing arctan, sqrt, and division using Texas Instruments's IQmath library on TI C64x+ DSP. These three functions are required for determining the addresses of neighboring pixels and calculating the corresponding coefficients for each scan converter output pixel for phased array scan images. sqrt and atan are required for every output pixel. div is required for each output pixel that is inside of the sectored image.

Table 1. DSP Cycles for IQmath Functions for sqrt, atan, and div

Functions	Execution Cycles (Stand alone)	Execution Cycles (Pipelined)
IQsqrt	79	8.6
IQatan	134	32.1
IQdiv	73	11.1

Assuming 50% of the output pixels fall into the region of the sectored image, 640x480 scan conversion output size and 25 frames per sec, and assuming using pipelined operation, which is more efficient, the total percentage of 600 MHz DSP required to conduct these computations alone is about 60%, which is obviously too high for any real time computation.

In addition to performing scan conversion for a B-mode image, if color scan conversion is required, operations are required for aliasing detection and correction. Extra operations for each display pixel are also required if tissue and flow decision and color mapping need to be supported, increasing the computational requirement even further.

Table 2 summarizes the instructions required for basic B-mode scan conversion based on bilinear interpolation. It assumes that the addresses and interpolation coefficients for all the output pixels have been pre-computed.

Table 2. Summary of Required Instructions for Basic B-Mode Scan Conversion

Instruction Type	Numbers
Load, Store	6
Multiply	4
And	4
Shift	5

Table 2. Summary of Required Instructions for Basic B-Mode Scan Conversion (continued)

Instruction Type	Numbers
Addition	5
Total Instructions	24
Total % of CPU assuming: 640x480 output size 25 frames/s No optimization at all	30.72%

Obviously, if measures are not taken to optimize the DSP software, to take advantage of the highly parallel architecture of C64x+ DSP, a significant amount of DSP cycles could be consumed for scan conversion operation. If scan conversion is required for both B-mode and color, which could include flow velocity and turbulence assuming the output image sizes are the same for both B-mode and color, the total amount of cycles could triple and make the problem even worse.

2.3.2 High I/O Bandwidth Requirement

Scan conversion requires high I/O bandwidth for the CPU to access all the necessary data. Because of the non-consecutive nature of memory access to the input data, If not designed properly, the scan conversion implementation can incur significant I/O traffic and cache penalties that can have a significant negative effect on the overall system performance.

In general, the pre-scan converted image cannot fit entirely into the internal DSP memory in most portable ultrasound systems. Therefore, the entire input ultrasound image and the scan converted output image need to be stored in the external memory.

There are different methods of implementing scan conversion in terms of the sequence of the samples to be generated. One way is to do the scan conversion along the horizontal direction. As the scan conversion moves along the horizontal direction, it needs to access samples on different scan lines for its input data. As the scan conversion travels from one end of the ROI to the other end in the horizontal direction, it touches all the scan lines that are stored in the external memory. With limited size of internal cache memory in embedded processors for portable markets, cache misses can happen frequently as the scan converter scans through the entire ROI. Moreover, for significant numbers of display pixels, depending on the configuration of cache memory size, the scan converter could access cache lines that reside in the same location in the cache memory, resulting in repeated cache threshing.

If not designed properly, a significant number of cache misses and cache threshing can be incurred during the entire course of the scan conversion operation for every frame of an ultrasound image, which significantly increases the overall cache penalties and lowers the overall scan conversion performance. Benchmarks have shown that the scan conversion performance can be several times lower in terms of processing speed or several times higher in terms of total CPU cycles consumed due to the cache penalties when the processor tries to access its input data directly from external memory.

This underscores that it is very important to design an effective I/O scheme to transfer input data from external memory to internal memory and reduce the potential cache penalties.

2.3.3 High Precision Requirement

Scan conversion has significant precision requirements in order to achieve high accuracy. For example, to generate an output sample of scan conversion for ultrasound images from a phased array scan head, you need to determine the addresses of the required input pixels and their corresponding coefficients for interpolation. This requires calculating the angles between the line connecting the apex of the sector and the output sample pixel and its neighboring scan lines. To accurately compute the angles, it requires the accurate physical location of the pixel. A small amount of error in keeping track of the physical location of the display pixels could result in a big error in the angle computation. A big error in angle computation can result in significant errors in both address and interpolation computations, resulting in significant errors in the final outputs of the scan converter.

This phenomenon may not be an issue in systems based on floating-point computation, but extra considerations should be provided in order to maintain high accuracy to get the power and cost savings of a fixed-point implementation.

3 TI DSP Based Scan Conversion

Efficient scan conversion for digital ultrasound imaging has been implemented on the TI System-on-Chip (SoC) that are based on the C64x+ DSP core. This optimized scan conversion software is implemented to achieve high efficiency, high accuracy, and high flexibility. It is designed to be easy to use and most of the complexity of the software is hidden.

3.1 General Description of Implementation

To reduce run time computational load, TI's scan conversion pre-computes the addresses of four input samples and the corresponding interpolation coefficients for all the scan conversion outputs. During run time computation, the scan converter simply computes the output samples based on bilinear interpolation using the pre-computed addresses and interpolation coefficients.

TI's scan conversion consists of two parts: initialization and run time computation.

During initialization, the addresses and interpolation coefficients for all the output samples are pre-computed. A DMA table is also computed, which describes the input samples that are required to be brought into internal memory.

Run time processing of scan conversion computes output samples based on bilinear interpolation using the pre-computed addresses of input samples and interpolation coefficients. In addition to interpolation, the run time computation also manages I/O traffic to transfer the tables for all the addressed and interpolation coeffcients, input samples, and output samples in and out of the DSP internal memory.

3.2 API Definition

The API of TI's scan conversion software is designed to allow you to define the physical parameters of the input ultrasound image, mode, and the region of interest for the scan conversion. The following lists some of the main parameters that can be specified through the API.

- Input Geometry
  - Image shape: linear array or phased array
  - Sector angle for phased array or lateral angle for linear array
  - Start of the scan line
  - End of the scan line
  - Number of scan lines
  - Number of samples on a scan line
- Input Data Format
  - 8 bits signed or unsigned
  - 16 bits signed or unsigned
- Output Data Format
  - 8 bits signed or unsigned
  - 16 bits signed or unsigned
- Output Window
  - Number of pixels in horizontal direction
  - Number of pixels in vertical direction
- Mode of operation
  - B-mode or Color

Figure 3 and Figure 4 show the parameters of TI scan conversion API that specify the scan input physical characteristics and outputs parameters from linear and phased or curvilinear probes, respectively.

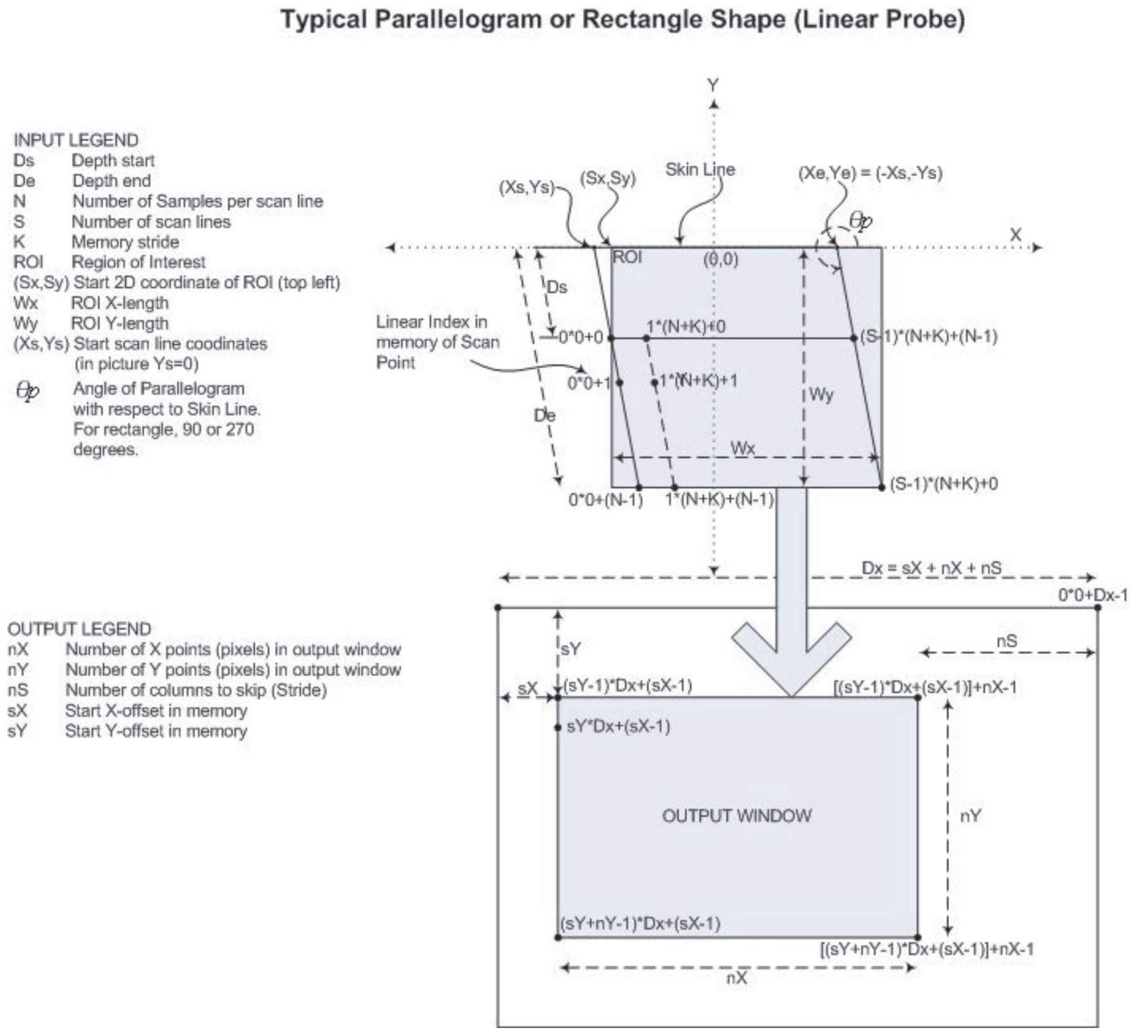

Figure 3. Input and Output Parameters of TI Scan Conversion API for Linear probe

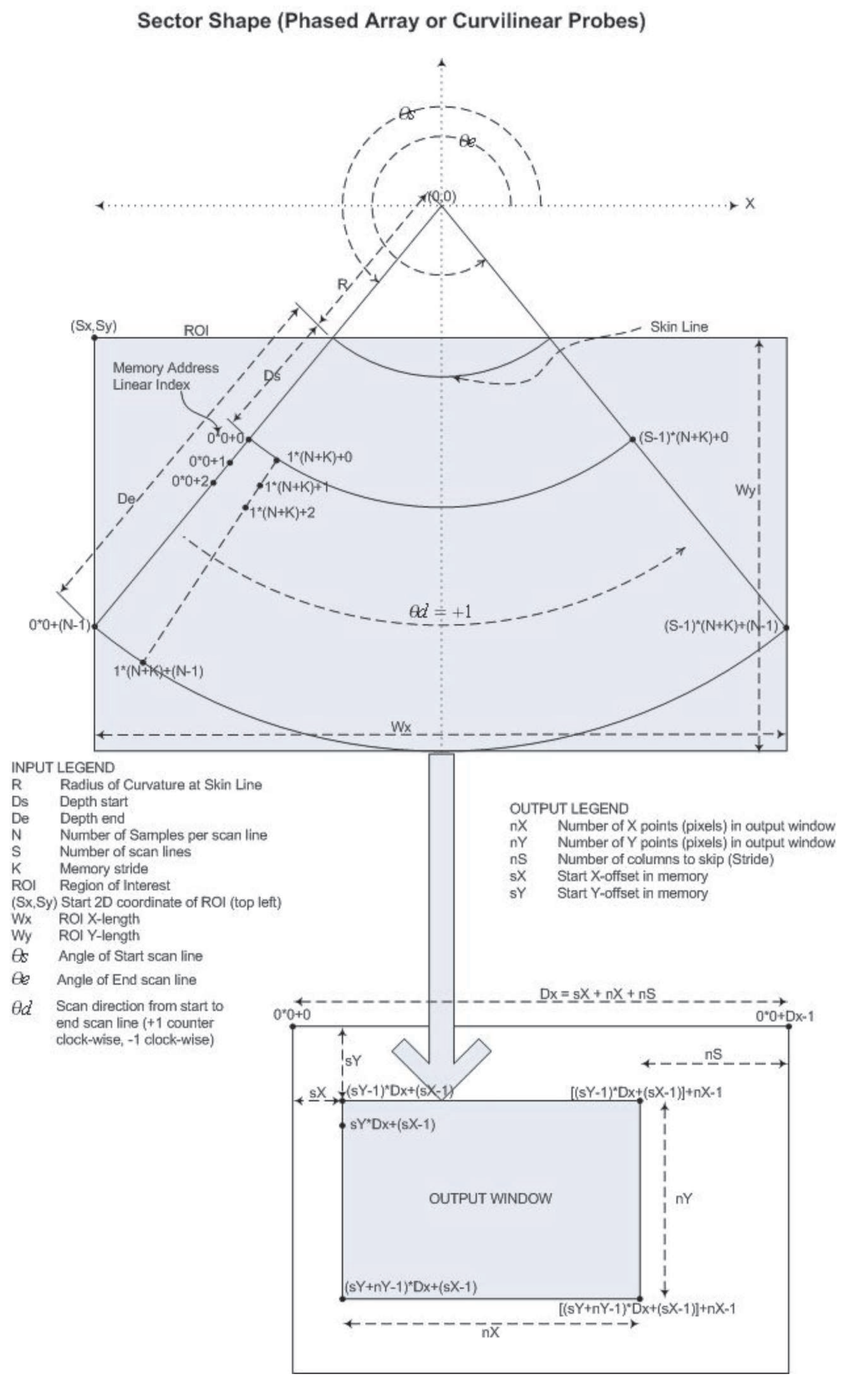

Figure 4. Input and Output Parameters of TI Scan Conversion API for Phased or Curvilinear Probe

3.3 Key Design Features

Special considerations were taken into account during design and implementation of the TI scan conversion software to overcome the challenges in achieving efficient performance. Those considerations are discussed in the following sections.

3.3.1 Achieving High Computation Efficiency

To reduce the overall computation load during scan conversion operation, the addresses of the input pixels of each display pixel and their corresponding interpolation coefficients are pre-computed and stored in external memory during initialization. For 2x2 bilinear interpolations, 32 bits are used for the addresses and 32 bits are used for the four interpolation coefficients, 8 bits each. Therefore, a total of 64 bits are required for the address and interpolation coefficients for each display pixel.

To achieve high efficiency during the scan conversion operation, highly efficient C64x+ DSP kernels are designed to implement interpolation for both B-mode and color scan conversion. For color scan conversion, the kernel not only implements interpolation, but also implements aliasing detection and correction for each display pixel. Efficient kernels that incorporate tissue and flow decision and color mapping are also implemented for the combined operation for B-mode and color scan conversion, flow and tissue decision, and color mapping. These kernels are designed to fully utilize all C64x+ DSP processing units at every DSP clock cycle and achieve maximum efficiency for scan conversion operation.

3.3.2 Reducing I/O Bandwidth Requirement and Cache Penalty

The TI DSP-based scan conversion software uses DMA extensively to transfer various required data in and out of DSP internal memory to reduce cache penalties. Figure 5 shows the simplified TI SoC device demonstrating the I/O scheme of TI's scan conversion implementation.

The scan conversion software operates in the horizontal direction. For every line, the software needs to transfer necessary inputs, the addresses, and the coefficients for all the pixels along the horizontal line to the internal memory. After processing each line, it also needs to transfer its outputs and the scan converted samples to the external memory. The scan conversion software allocates a small buffer in L1 SRAM for storing each line of the addresses and coefficients and scan conversion outputs.

In portable systems, it is very difficult to use a direct ping-pong buffer scheme for transferring input data because the pre-scan converted ultrasound image cannot usually fit entirely into DSP internal memory, the access pattern to the input or pre-scan converted image is not along scan lines, and the locations on every scan line varies. In TI's scan conversion implementation, a small image buffer is allocated in L2 SRAM and an intelligent DMA scheme is developed to keep track of the input data that is required for each line of output samples and transfer the required input samples into the small image buffer in L2 SRAM. This scheme allows the scan converter to localize the input samples that are required and transfer a minimum amount of the required input samples to the internal memory.

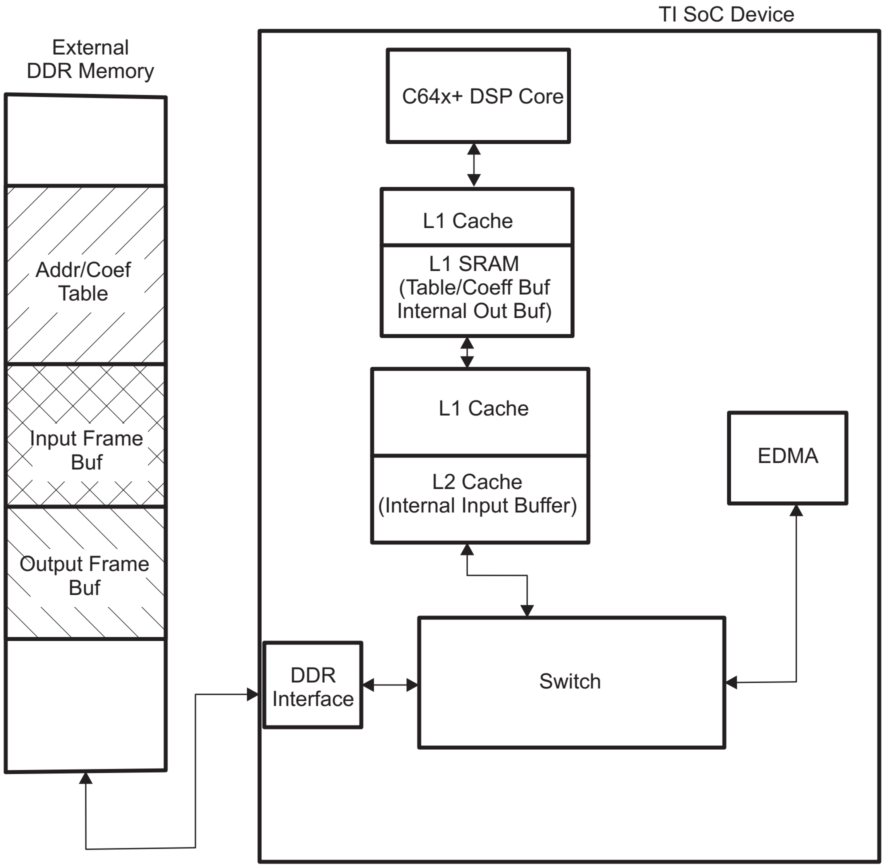

Figure 5. Block Diagram for TI Scan Conversion I/O Scheme

3.3.3 Maintaining Accuracy

High precision is required during address and coefficient calculations to insure high accuracy of scan conversion. Careful analysis has been done to identify the area and variables that are required for higher precision. For example, the coordinates, x and y, of each scan conversion output pixel need to be in high precision to ensure accurate angle calculation for computing addresses and neighboring pixels and their interpolation coefficients. Higher precisions are used for all those variables that are necessary to insure high accuracy while maintaining efficiency at the same time.

3.4 Modes of Operations

This section lists the modes of operation that the TI scan conversion software can operate on. The necessary API function calls for each mode are also listed here for demonstration purposes. Some other API functions are omitted here for reason of brevity.

3.4.1 B-Mode Scan Conversion

The TI scan conversion DSP software can be configured to perform scan conversion for B-mode picture. It takes unsigned 8 bit inputs and generates scan converted outputs in 8 bits unsigned format. The scan conversion is currently based on 2x2 bilinear interpolations. The B-mode scan conversion operation can be achieved through the following API function calls.

```
// configure scan converter
setScuConfig( &scuConfig_d );

// call scan converter if a frame of input is available
scuProcess_BMode( &pInFrame, &pOutFrame );
```

Here scuConfig_d is a structure that is defined by the API, specifying all the necessary parameters for the software scan converter for B-mode operation.

3.4.2 Color Scan Conversion

The TI scan conversion DSP software can be configured to perform color scan conversion. It takes a frame of signed 8 bit inputs and generates scan converted outputs in 8 bits signed format. This is mainly used for processing velocity image of blood flow. The scan conversion performs aliasing detection and correction and generates new samples based on 2x2 bilinear interpolations. The color scan conversion operation can be achieved through the following API function calls.

```
// configure scan converter
setScuConfig( &scuConfigColor_d );

// call scan converter if a frame of input is available
scuProcess_Color( &pInFrame, &pOutFrame );
```

Through separate API calls for B-mode and color, you can generate scan converted B-mode and color flow image. You can implement your own scheme for tissue and flow decision and color mapping to combine B-mode and color images and produce final ultrasound image for display.

3.4.3 B-Mode Scan Conversion With Color Mapping and 422 Video Output Format

The TI scan conversion DSP software can be configured to perform B-mode scan conversion, color mapping and produce outputs in 422 video format. It takes a frame of unsigned 8 bit inputs and generates scan converted outputs in 16 bits 422 video format for each display pixel. The scan conversion uses 2x2 bilinear interpolations. The scan conversion can be achieved through the following API function calls.

```
// configure scan converter
setScuConfig( &scuConfig_d );

// call scan converter if a frame of input is available
scuProcess_BMode_422( &pInFrame, &pOutFrame, &pBmodeColorMapTbl );
```

3.4.4 B-Mode and Color Scan Conversion, Tissue/Flow Arbitration, Color Mapping With 422 Video Output Format

The TI scan conversion DSP software can be configured to perform B-mode scan conversion, color scan conversion, tissue and flow decision, color mapping and produce outputs in 422 video format. It takes a frame of unsigned 8 bit inputs for B mode and 16 bits input for color generates scan converted outputs in 16 bits 422 video format for each display pixel. Each 8 bits unsigned B-mode input is for regular tissue image. Each 16 bits input for color consists of 8 bits signed velocity and 8 bits unsigned variance of blood flow image. The scan conversion is based on 2x2 bilinear interpolations. The scan conversion can be achieved through the following API function calls.

```
// configure scan converter
setScuConfig( &scuConfig_d );
setScuConfig( &scuConfigColor_d );

// call scan converter if a frame of B-mode and a frame of color image are
// available
scuProcess_BModeColor_ArbColorMapping_422(
                                            &pInFrameBmode,
                                            &pInFrameColor,
                                            &pOutFrame,
                                            &pArbTbl,
                                            &pColorMapTbl_Bmode,
                                            &pColorMapTbl_Color);
```

This mode is designed to achieve maximum efficiency by combining B-mode and color scan conversion, tissue and flow arbitration, and color mapping for both tissue and flow images. It still gives you the freedom of designing your own schemes for tissue and flow arbitration and color mapping by providing your own tissue/flow arbitration table and color mapping tables for both tissue and color flow images.

4 Performance and Memory Requirement of TI DSP Based Scan Conversion

Performances in terms of cycles per pixel, percentage of DSP MHz consumption, and memory requirement of TI DSP-based software scan conversion is listed in this section. Because both performance and memory requirements are directly related with the physical parameters of the input ultrasound images and output window size, the performance data are provided for an example with the major parameters shown below with output window size of 640x480.

Parameters for B mode and color input images and output window:

- Phased array
- B-mode sector angle: 90 degree
- B-mode number of scan lines: 256
- B-mode number of samples per scan line: 512
- Color sector angle: 70 degree
- Color number of scan lines: 64
- Color number of samples per scan line: 128
- Output window size for B-mode: 640x480
- Output window size for color: 640x480

Table 3. Internal Memory Requirement for B-Mode Scan Conversion for 640x480 Output Window Size

	B-Mode, 8 Bits Unsigned Output	
	L1D SRAM	**L2 SRAM**
Input Buffer	0	16 Kbytes
Output Buffer	1280 bytes	0
Addr/Coeff Table	10240 bytes	0
B-Mode Color Mapping Table	0	0
Total Memory	11.25 Kbytes	16 Kbytes

Table 4. Internal Memory Requirement for B-Mode Scan Conversion for 422 Output Format for 640x480 Output Window Size

	B-Mode, 422 Video Format Output	
	L1D SRAM	**L2 SRAM**
Input Buffer	0	16 Kbytes
Output Buffer	2560 bytes	0
Addr/Coeff Table	10240 bytes	0
B-Mode Color Mapping Table	1024 bytes	0
Total Memory	13.5 Kbytes	16 Kbytes

Table 5. Internal Memory Requirement for Color Scan Conversion for 640x480 Output Window Size

	Color, 8 Bits Signed Output	
	L1D SRAM	**L2 SRAM**
Input Buffer	0	16 Kbytes
Output Buffer	2560 bytes	0
Addr/Coeff Table	10240 bytes	0
B-Mode Color Mapping Table	0	0
Total Memory	12.5 Kbytes	16 Kbytes

Table 6. Internal Memory Requirement for B-Mode and Color Scan Conversion, Arbitration, Color Mapping With 422 Output Format for 640x480 Output Window Size

	B-Mode, Color, Arbitration, Color Mapping	
	L1D SRAM	**L2 SRAM**
Input Buffer	0	16 Kbytes
Input Buffer	1280	0
Output Buffer	2560 bytes	0
Addr/Coeff Table	10240 bytes	0
Arbitration Table	2048 bytes	0
Color Mapping Table - B Mode	1024 bytes	0
Color Mapping Table – Color	8192 bytes	
Total Memory	24.75 Kbytes	16 Kbytes

Table 7. Performance for Various Modes of Scan Conversion Operation for Sectored Image With 640x480 Output Window Size

	DM6437, 16K L1D Cache, 64K L1D SRAM, 64K L2 Cache, 64K L2 SRAM	
	Average Cycles/Pixel	**% of 600 MHz C64x+DSP (25 frames/s)**
B-Mode	7.97	10.20%
B-Mode With 422	8.65	11.07%
Color	5.58	7.14%
B-Mode, Color, Arb, Mapping, 422	18.80	24.07%

5 Summary

This application report discusses implementing scan conversion on TI C64x+ DSP core-based devices for digital ultrasound applications. TI's efficient scan conversion implementation was introduced. The basic B-mode of this scan conversion consumes less than 9% of the cycles of a 600 Mhz C64x+ DSP core for a 640x480 output size. This performance reflects all the potential overheads such as DMA transfer and any penalties due to cache misses and memory bank conflicts. By consuming such a small portion of the DSP MHz, TI's scan conversion leaves enough headroom for other important and processing-intensive algorithms used in ultrasound image processing. This efficient scan conversion makes it possible to implement portable and hand held digital ultrasound machines using TI DSP devices such as the DM6446 and OMAP3x family of SOCs.

TI's scan conversion implementation, as well as many other ultrasound algorithmic blocks, is available as part of the TI Embedded Processor Software Toolkit for Medical Diagnostic Ultrasound.

6 References

1. Schueler, H. L., Lee, H., and Wade, G., Fundamentals of digital ultrasonic imaging, *IEEE Trans. Sonics Ultrasound*, SU-31, 195-217 (1984)
2. Robinson, D.E., and Knight, P.C., Interpolation scan conversion in pulse-echo ultrasound, *Ultrasonic Imaging*, 4, 297-310 (1982)
3. Park, S.B., and Lee, M.H., A new scan conversion algorithm for real time sector scanner, *1984 IEEE Ultrasounics Symposium*, 723-727 (1984)