Digital Signal Processor (DSP) for Portable Ultrasound

Rama Pailoor and Dev Pradhan

ABSTRACT

Ultrasound imaging is a non-invasive real-time imaging tool that is finding increased usage in hospitals and clinics. Ultrasound technology uses high frequency sound waves to reflect off changes in acoustic impedances between tissues in the body and constructs an image based upon computer's interpretation of waves' reflections.

This application report provides a high level description of the signal/image processing functional blocks in a typical portable ultrasound system. Examples of system architectures and execution benchmarks of key processing functions are given to demonstrate how the driving requirements of portable ultrasound imaging systems could be met with the use of TI DSPs and system-on-chip (SoC) devices.

Contents

List of Figures

List of Tables

1 Introduction

Portable ultrasound systems are considered to be ultrasound systems that weigh around 10 lbs or less, and can run on batteries. They began to appear in the market place in the late 90s and have seen a remarkable growth in sales in the recent years. This growth has been a direct result of their applicability in areas such as ICUs, emergency medicine, regional anesthesia and battlefield.

DSPs and SoCs are specially designed single-chip digital microcomputers that process digitized electrical signals generated by electronic sensors (e.g., cameras, transducers, microphones, etc.) that will help to revolutionize the area of diagnostic ultrasound imaging. A diagnostic ultrasound imaging system generates and transmits acoustic waves and captures reflections that are then transformed into visual images. The signal processing on the received acoustic waves include interpolation, decimation, data filtering and reconstruction. Programmable DSPs and SOCs, with architectures designed for implementing complex mathematical algorithms in real-time, can efficiently address all the processing needs of such a system.

The following information introduces the concept of a complete portable ultrasound system solution based on Texas Instruments' (TI) semiconductor components, development tools, and software solutions. Additionally, the various concepts that outline the inherent advantages of a DSP and a SoC in a portable ultrasound system - efficient signal processing, lower power consumption and lower cost, all leading to better ultrasound diagnostic imaging - will also be covered.

2 Portable Ultrasound System

2.1 Driving Requirements for a Portable System

The key driver requirements for a portable ultrasound system are the same with any portable device: size, weight, battery life, cost and performance. OEMs are making trade-offs in these areas, for example, providing just a basic imaging system with less features but with more battery life, (e.g., 8-channel black and white systems vs. more sophisticated 128 channel color systems that would need to be re-charged more often). The size of the portable system varies from laptop sized systems to handhelds. These size limitations are driving the need for more system integration on the supporting SoCs and more automatic image enhancement features due to fewer fine controls.

The requirements for portable systems is also being driven from geographies where the infrastructure is more rural and access to the larger more sophisticated imaging systems is limited, and where clinicians must now take the system to the patient. This makes cost a critical factor as well.

2.2 Complete System View

The basic functional building blocks of a diagnostic ultrasound imaging system are: the transducer, processing blocks needed for image formation, system controller and power supply module as shown in Figure 1.

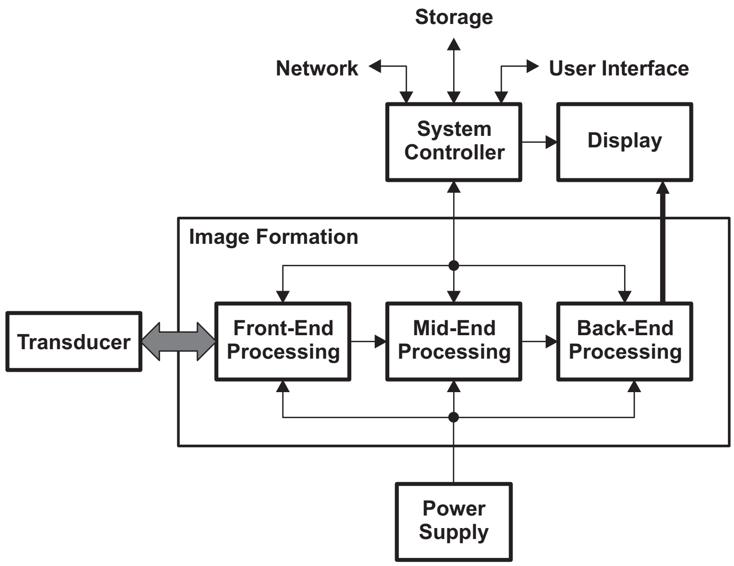

Figure 1. Ultrasound Imaging System Functional Blocks

Delineation of signal path operations as front-end, mid-end and back-end processing varies from one manufacturer to another. It also depends, to some degree, on the type of technology used for carrying out these operations: application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), DSP or PC.

2.3 Front-End Processing

Ultrasound image formation begins, in what is commonly referred to as the front end, using pulse-echo technique. Figure 2 shows the main components of this functional block.

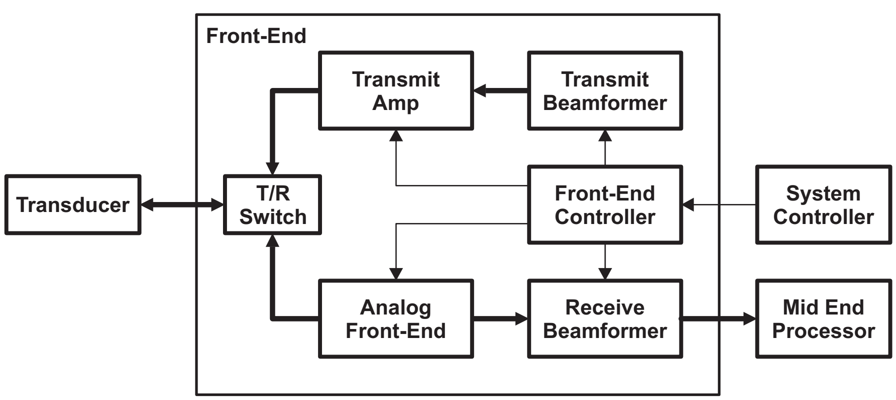

Figure 2. Components of Front-End Processing

The transmit beamformer is responsible for the orderly pulse-excitation of transducer elements, which results in emission of acoustic waves into the region of interest. Immediately following this, the transmit/receive (T/R) switch is positioned to put the front end in receive mode. Transducer elements transform the reflections or echoes into corresponding electrical signals. The analog front end (AFE) properly amplifies these signals and converts them into digital data streams for further processing. By applying dynamic delays into these data streams, the receive beamformer combines them to form a scan line, a representation of the region of interest along a given line of sight. This process is repeated either sequentially or simultaneously to form multiple scan lines to cover a region of interest. The front-end controller is responsible for controlling the timing and sequencing of transmit and receive beams. Sampling rates used for analog-to-digital (A/D) conversion in the front end can vary from 16 MHz to 50 MHz depending on system requirements.

2.4 Mid-End Processing

Most ultrasound imaging systems introduced in recent years are capable of three major modes of operation: B-mode, color-flow and Doppler. B-mode operation results in a gray scale image that is used for examining tissue structures and organs. Color-flow operation results in a color-coded display of spatial distribution of mean velocity of blood flow super-imposed on gray scale image. Doppler processing produces scrolling display of blood flow velocity distribution at a user specified location. Some systems are capable of displaying all three modes simultaneously.

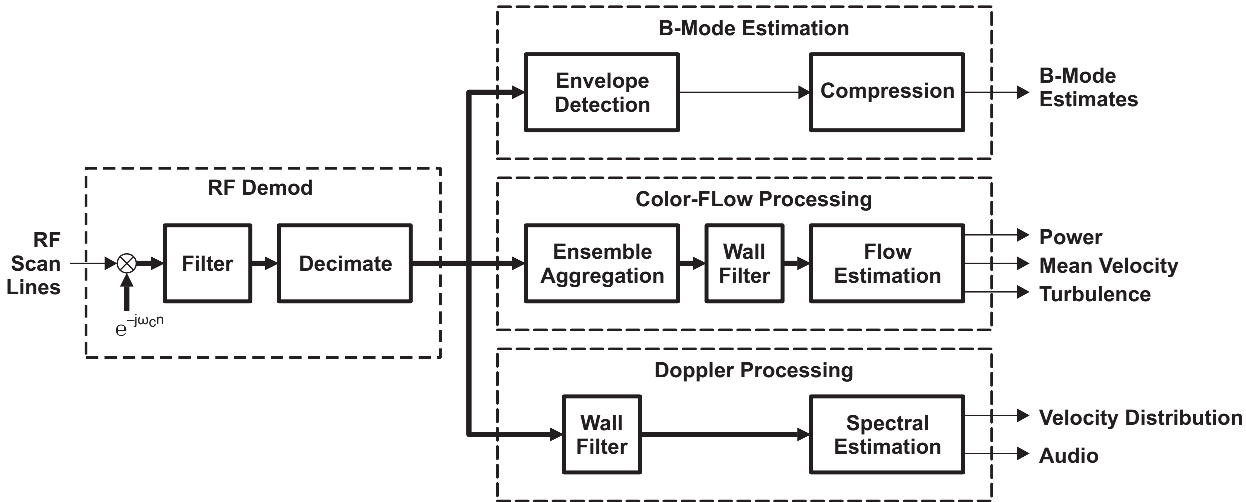

Figure 3. Mid-End Processing

Figure 3 shows major elements of the mid-processing section for the three major modes of operation. Common to all three is the initial stage where beamformed RF data gets down converted to baseband, filtered, and in most cases, decimated. What follows this depends on the mode. For B-mode, it is envelope detection and logarithmic compression. What needs to take place for color-flow is a lot more compute intensive involving high pass filtering of ensembles of scan lines to remove contributions from vessel wall or tissue motion. Output of the wall filter is then used to estimate power, mean velocity, and turbulence. Doppler processing involves a much simpler wall filter and estimation of velocity distribution using short-time Fourier transform techniques. Doppler processing also produces a stereo audio signal representing the Doppler spectrum.

2.5 Back-End Processing

Figure 4 shows an example of what takes place in the back end.

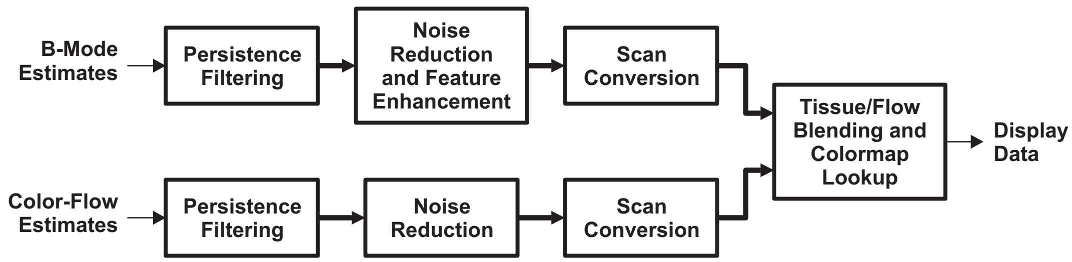

Figure 4. Back-End Processing

B-mode and color-flow estimates are subjected to temporal and spatial processing to reduce noise and enhance features of interest. Scan conversion is the operation where B-mode and color-flow estimates are converted to display raster data, pixels with 1:1 aspect ratio. When color-flow is on, B-mode and color-flow pixels have to be blended to produce a single image. This blending is typically based on application dependent thresholds.

2.6 System Controller and MMI

Just like most embedded systems, ultrasound imaging systems need a system controller to carry out functions such as:

- Configure and control the signal path
- Handle user input events and take appropriate actions
- Monitor acoustic pressure and intensity levels and ensure safety of patients
- In the case of portable systems, carry out smart power management to maximize scanning time in a single charge.
- Store and recall image clips
- Run applications to allow you to make clinically relevant measurements on acquired image sequences

3 DSP Based Portable Ultrasound Solution

3.1 Advantages of DSP and SOCs in a Portable System

Traditionally, ultrasound systems have used a combination of ASIC, FPGA, DSP and PC for capturing and processing beamformed ultrasound images and rendering them for visual display. These systems were cart-based systems that plugged into a wall for power and typically did not move from location to location. With ultrasound imaging systems now going portable, there is a need for a simpler user interface and lower power consumption. The systems also need to be rugged, upgradeable in the field, and have good connectivity (maybe even wireless). The smaller form factor on the portable systems also means that they are not expected to have features needed for detailed manual measurements and analysis, and that more must be done with the limited controls available.

DSPs and SOCs today are pervasive in hand held portable devices; their strength is their high performance per mW of power and the high level of system integration for support for nearly all standard peripheral devices (e.g., USB storage, LCD display, Ethernet, 802.11, etc.). Couple that with the flexibility they provide via programmability to support more intelligent processing and image enhancement analytics and it is easy to see how DSPs and SOCs enable a very powerful solution in a portable ultrasound system.

Most DSPs and SOCs are also well supported by real-time operating systems that abstract out some of the complexities of the hardware, and provide a more familiar software development environment that have been tuned to support applications with real-time deterministic requirements and limited memory resources.

The following examples show how Texas Instruments' DSP and SoC devices can be used in portable ultrasound systems.

3.2 Example System Block Diagrams

Figure 5 shows system block diagram highlighting the use of TMS320DM648 and TMS320DM6446 for carrying out mid-end, back-end, and system controller functions.

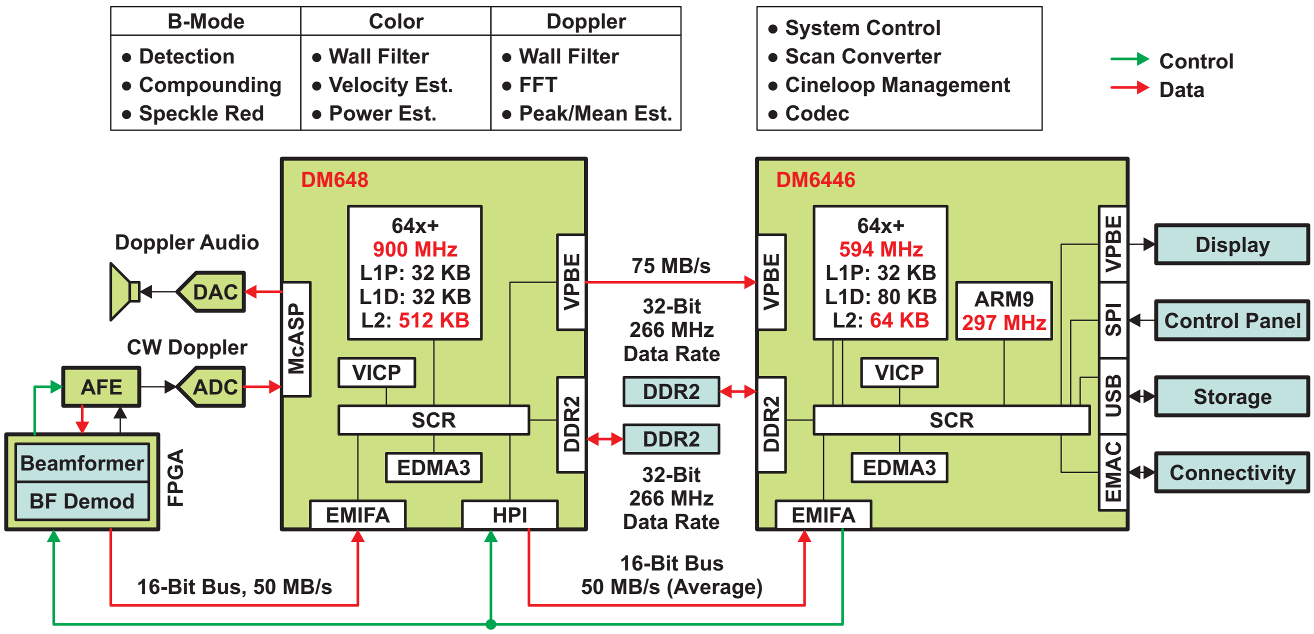

Figure 5. DM648 + DM6446 Block Diagram

The combination of two C64x+™ cores and video/imaging coprocessors (VICP) offer compelling compute capability to address the needs of the three modes of ultrasound operation. The primary role of the ARM9 is to run the operating system upon which the system controller functions and user interface can be implemented.

Figure 6 shows an example of a system requiring higher input/output (I/O) bandwidth and computations than the system shown in Figure 5. A TMS320C6455 DSP is used here, with a wider EMIFA bus, which allows higher input data rates. Larger L2 memory and higher operating clock frequency are the major contributors to increased compute capability. In this example, the OMAP3530 plays the dual role of system controller and back-end processor.

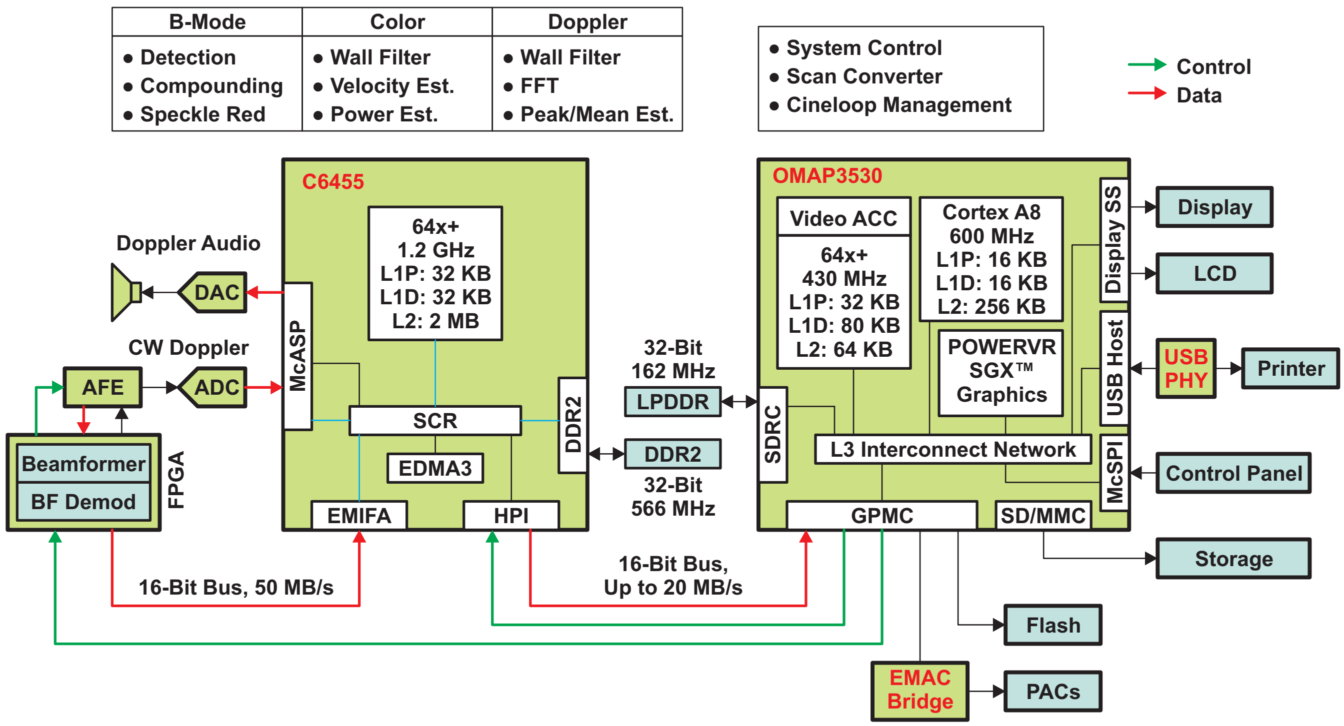

Figure 6. C6455 + OMAP3530 Block Diagram

3.3 Real-Time Operating Systems (RTOS)

To make application development easier, more portable from one hardware platform to another, and faster to market, embedded systems today are gravitating more and more to off-the-shelf embedded operating systems. One can leverage the many man years of effort that RTOS developers have put in to make this hardware abstraction layer more feature rich, more robust, and more efficient in the support functions of the application.

Besides standard operating system features like multi-tasking, memory management, interrupt, and event handling, embedded operating systems have some special characteristics that differentiate them from non-embedded OSes (like Windows® XP.) For example, embedded OSes are usually configurable to allow you to add or remove features as needed. They normally have a small memory footprint, higher predictability (real time) with a pre-emptive priority-based scheduling scheme, efficient task switching latency, light weight, predictable inter-process communication mechanisms, power management support, etc. These are key needs for embedded systems that are often real time in nature, cost conscious, and power efficient.

In a portable ultrasound system many of these features come into play. Specifically, the real-time nature of an ultrasound system makes it important to be able to capture ultrasound images, process them, and render them to be seen on a screen instantly. System startup (boot time) is expected to be fast (< 10 seconds) and the latency, when switching from B-mode estimation to color flow and Doppler processing, is expected to be minimal (< 1 second). Long battery life (8 hours) and power savings mode, with quick wake-up capability, is an important requirement. Support for standard peripherals like USB storage, printers, LCD displays, etc., is a must. All of these needs are well supported in most off the shelf RTOS available on TI SOC devices.

4 System Characteristics

4.1 Benchmarks

The following sections provide C64x+ cycle count estimates for several key functions needed in mid-end and back-end processing. These estimates are based either on stand-alone benchmarking or benchmarks from TI's dsplib and imglib. For more detailed analysis of several functions used for signal path implementation, see *Efficient Implementation of Ultrasound Color Doppler Algorithms on Texas Instruments' C64x+TM Platforms* (SPRAB11). The numbers provided here give you an idea about compute power needed for each mode of operation. Actual performance numbers may vary depending on system design and usage of shared memory resources.

Table 1 gives a summary of CPU usage estimates for various processing blocks described in previous sections. The numbers are for a C64x+ core running at 1 GHz..

Table 1. Key Benchmarks

Function	CPU Usage	Assumptions
RF Demod	≤ 16%	RF sampling rate = 50 MHz Imaging depth ~= 25 cm (scan line interval = 325 μs) Filter Length (L) = 16 (# of taps per output sample) Number of output samples = 512 (decimation factor (S)=32) Number of cycles per output sample= S + (0.625L + S) = 74
Color-Flow Wall Filter	≤ 5%	Ensemble length (N) = 10 Number of scan lines per frame = 128 Number of color flow samples per scan line = 256 Frame rate = 30 Hz Number of cycles per output sample = $N^2/2$ CPU Usage = < 5%
Color-Flow Parameter Estimation	≤ 6%	Ensemble length (N) = 10 Number of scan lines per frame = 128 Number of color flow samples per scan line = 256 Frame rate = 30 Hz Number of cycles/output for correlation computation = (N+1)/2 Number of cycles/output for velocity estimation = 37 (approximate) Number of cycles/output for turbulence estimation = 16 (approximate)
Spectral Doppler Wall Filter	≤ 1%	Pulse Repetition Frequency (P) = 20 kHz Wall Filter Length (L) = 64 Number of cycles per output = 2*((L/4 + 4) + 15) (dsplib benchmark)
Spectral Doppler Estimation	≤ 1%	Estimation of velocity distribution is typically done using Short Time Fourier Transform techniques. FFT Size = 256 Rate of FFT computation = 1 kHz Benchmark for 256 point 16x32 complex FFT = 1827 cycles (intrinsic C version) Number of cycles needed for windowing and power estimation = 512 per spectral column.
Scan Conversion	≤ 10%	Size of output raster used for displaying ultrasound image: 512 x 512 Acquisition Frame Rate: 30 Hz

5 References

- Thomas L. Szabo, *Diagnostic Ultrasound Imaging*, ISBN-13: 978-0-12-680145-3
- *TMS320C64x+ DSP Little-Endian DSP Library Programmer's Reference* (SPRUEB8)
- *TMS320C64x+ DSP Image/Video Processing Library (v2.0) Programmer's Reference* (SPRUF30)
- *Signal Processing Overview of Ultrasound Systems for Medical Imaging* (SPRAB12)
- *Efficient Implementation of Ultrasound Color Doppler Algorithms on Texas Instruments' C64x+™ Platforms* (SPRAB11)

Appendix A TMS320 Digital Signal Processor Platform

The TMS320 DSP family from TI offers the widest selection of signal processors available anywhere, with a balance of general-purpose and application-specific processors to suit your needs. There are three distinct instruction set architectures that are completely code-compatible within the following platforms:

- High performance multi-core DSP platforms
- High performance single core DSP platforms
- Power efficient DSP and SOC platforms

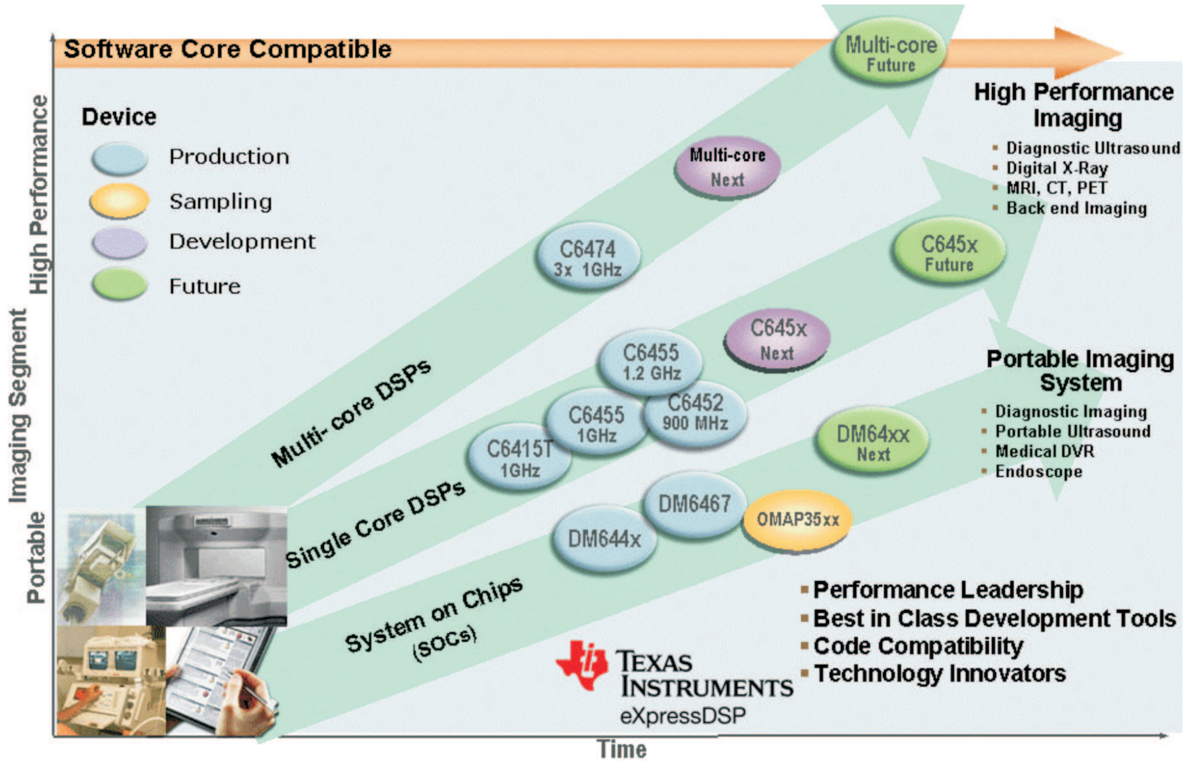

Figure A-1. Roadmap for DSP Architectures

Appendix B eXpressDSP™ Software and Development Tools

TI's real-time eXpressDSP software and development tool strategy includes three tightly knit ingredients that empower developers to tap the full potential of TMS320™ DSPs:

- The world's most powerful DSP integrated development tools: Code Composer Studio™ IDE
- eXpressDSP Software including:
  - Scalable, real-time software foundation: DSP/BIOS Real-Time Operating System (OS) Software Kernel Foundation
  - Standards for application interoperability and reuse: TMS320 DSP Algorithm Standard – xDAIS and XDM
  - Design ready code that is common to many applications to get you started quickly on DSP design: eXpressDSP Framework Software
- A growing base of TI DSP-based products from TI's DSP Third Party Program including eXpressDSP compliant products that can be easily integrated into systems.

Each element is designed to simplify DSP programming and move development from a custom crafted approach to a new paradigm of interoperable software from multiple vendors supported by a worldwide infrastructure.

There has been an explosive growth in real-time applications demanding the real-time processing power of TI DSPs. eXpressDSP enables innovators and inventors to speed new products to market and turn ideas into reality. Previously unimagined applications, including virtual reality, medical imaging, auto navigation, digital audio, and Internet telephony now rely on the crucial real-time computing power that can only be found in a DSP.

DSK and EVM

DSP starter kits (DSK) and evaluation modules (EVM) are tools that allow developers to test their code in hardware before actually building prototype designs. DSKs are available for TMS320C6455 and TMS320C6713 DSP. EVMs are available for the TMS320DM6446, TMS320C6474, TMS320DM648 and OMAP35xx SOCs.