Synthetic Aperture Radar (SAR) Implementation on a TMS320C6678 Multicore DSP

Dan Wang
Member Group Technical Staff

Murtaza Ali
Distinguished Member of Technical Staff

Ellen Blinka
Marketing Manager

Texas Instruments

Introduction

Synthetic Aperture Radar (SAR) is a radar technique involving moving the radar platform to create the effect of a large antenna and thus achieve high-resolution remote sensing imagery. There are multiple algorithms that have been used to process received signals and form SAR images, all of which rely heavily on fundamental digital signal processing techniques such as FFTs and matrix manipulations like corner turns. For this reason, TI has implemented a SAR algorithm on the TMS320C6678 eight-core fixed- and floating-point DSP to show the full application performance and how it scales across one, two, four, and eight DSP cores.

Commercial off-the-shelf (COTS) components have been gaining popularity in Synthetic Aperture Radar (SAR) and other defense and airborne applications due to their widespread availability and relatively lower cost compared to custom-built solutions. However, the question of performance remains of critical importance to processor selection in these markets, and the compute capabilities of these COTS devices has advanced to a level where real-time processing of complex algorithms, for SAR and other applications, has become feasible. In this paper, we evaluate the capability of the TMS320C6678 processor for SAR signal processing and show its real-time processing abilities able to meet the demands of complex defense and avionic applications. Specifically, we show the modularization of the SAR algorithm and mapping the modules to C6678 architecture to achieve parallel computation. Further, the profiling results for key modules are demonstrated to evaluate the implementation quantitatively. Results indicate that a baseline SAR range-Doppler algorithm takes around 0.25 second for a 16 M (4K by 4K) image, achieving real-time performance.

TI has implemented a SAR algorithm on the TMS320C6678 eight-core fixed- and floating-point DSP to show the full application performance and how it scales across one, two, four, and eight DSP cores

SAR and the Range-Doppler Algorithm

Synthetic Aperture Radar (SAR) achieves high-resolution remote sensing imagery by moving the radar platform to create the effect of a large antenna; typically the radar is carried by a spaceborne or an airborne platform moving at known speed. A single physical antenna is used to gather signals reflected from the targets at different positions, at different times. The relative motion between the radar and the targets encodes the targets' information, which is processed to form a focused image of the surface area. At each radar position, the antenna system transmits a short chirped waveform, and then the reflected echoes from the targets are collected, digitized and stored for later processing. Advanced signal processing techniques are used to analyze the phase shift information and obtain fine resolution in the direction perpendicular to the beam direction.

Figure 1 on the following page shows the SAR geometry model of the radar location and the target surface. The pulse repetition time is the

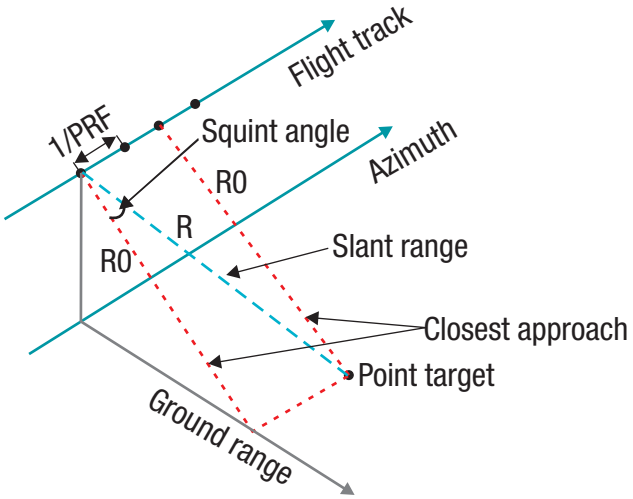

Figure 1: SAR geometry

inverse of the Pulse Repetition Frequency (PRF). The acquisition geometry makes the SAR image processing a two-dimension operation. The first dimension is called range (or cross track) that measures the "line-of-sight" distance from the radar to the target, shown as slant range along the radar sight or ground range along the ground in Figure 1. Range resolution is determined by the transmitted pulse width, so narrow pulses yield finer range resolution. The second dimension is azimuth aligned with the relative platform velocity vector. Azimuth resolution depends on the actual radar antenna length. The angle between the slant range and the closest approach is called squint angle, as shown in Figure 1.

The original data collected from the radar is unfocused, and the raw data must be processed to achieve digital focusing and generate the final SAR image. The general processing method involves weighting, phase shifting and summing the phase histories of the responses. There are multiple algorithms for achieving this; the most widely used is the Range-Doppler (RD) algorithm, with the ω–k 2D algorithm a close second. The RD algorithm features block processing efficiency and separability of processing in the two dimensions, range and azimuth, making it well-suited for parallel computing platforms. We have implemented the RD algorithm for both its efficiency and its parallel nature which can be exploited by multicore architectures, such as the TMS320C6678 eight-core DSP, to produce large improvements in processing time.

Multicore DSP Architecture

The TMS320C6678 DSP, shown below in Figure 2, is an eight-core, high-performance DSP, based on TI's fixed- and floating-point C66x DSP core. This device can run at a core speed of up to 1.25 GHz, providing up to 16 GFLOPS while consuming only 10W of power in typical use cases. The device is well suited to airborne and defense applications, such as synthetic aperture radars, as it has high performance per Watt, is available in industrial temperature ranges (–40–100°C case temp) and extended temperatures (up to 125°C), and is featured on many different COTS single-board computers in common form factors. Additionally,

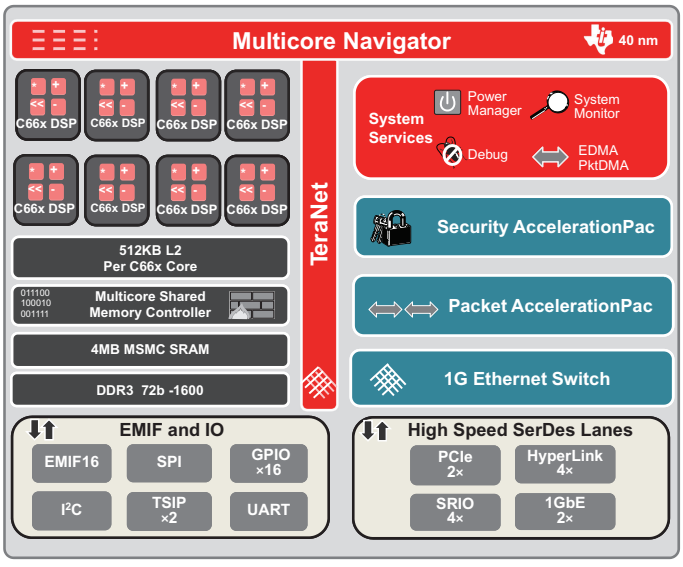

Figure 2: TMS320C6678 DSP block diagram

the device has a large amount of on-chip L2 and shared memory, all with ECC, and high-throughput peripherals such as PCIe, SRIO, Gb Ethernet, and HyperLink, which provides up to 50 Gbps connection to other KeyStone™-based devices, or to select FPGAs.

In the example code for the SAR algorithm, a 1-GHz device was used.

TI's DSPs run a lightweight real-time operating system called SYS/BIOS. Since SYS/BIOS can be used in a wide variety of processing and memory constraints, it was designed to be highly configurable. TI also provides Code Composer Studio™, an Eclipse based Integrated Development Environment (IDE) for code development, including a C/C++ compiler. The compiler is C89 compliant and virtually every C89 compliant code can be ported with no additional effort. The compiler also allows the use of pragmas and intrinsic operators to fully exploit the core architecture and extract all the potential performance without resorting to assembly programming. TI provides standard libraries which contain highly tuned, often-used signal processing and imaging functions. The SAR example code heavily uses the FFT/IFFT functions supplied with TI's DSPLib.

The compiler also supports OpenMP 3.0, which allows rapid porting of existing multi-threaded codes to the multicore DSP. TI's C66x compiler translates the OpenMP into multi-threaded code with calls to a custom runtime library. In this SAR example code we have used the OpenMP framework to instantiate individual threads across multiple cores.

Range-Doppler Algorithm Implementation on the TMS320C6678 DSP

To efficiently implement the RD algorithm on the TMS320C6678 DSP, each sequential task in the overall processing chain needs to be efficiently mapped to the DSP architecture. These main sequential tasks are: range compression, matrix transpose, range cell migration correction (RCMC) and azimuth compression. These steps are considered as modules for implementation on the TMS320C6678 DSP, and each can be broken down into even further detail, as shown at the top of Figure 3.

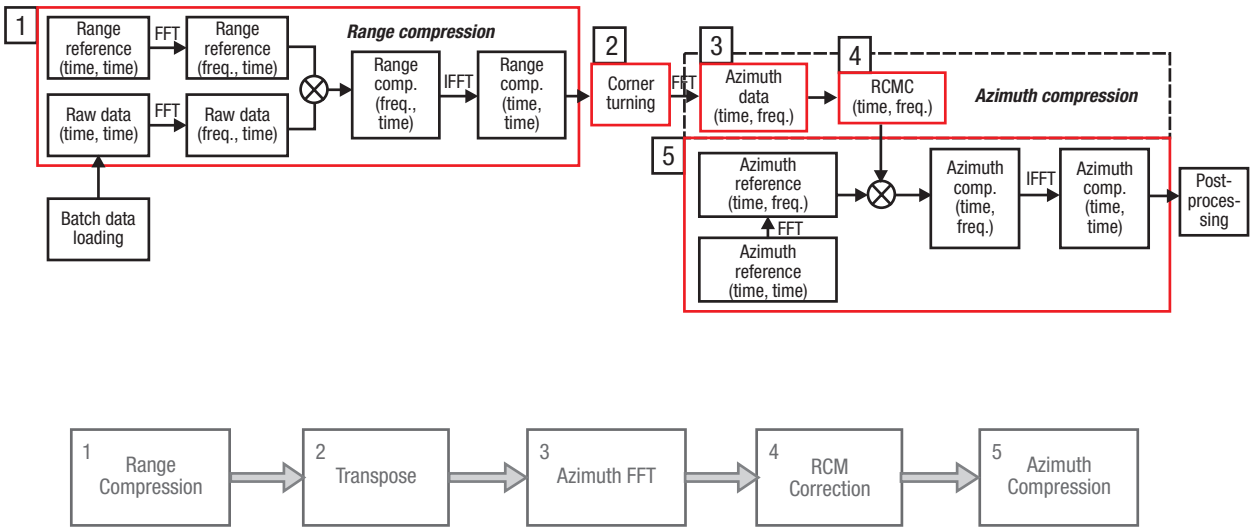

Figure 3: Range-doppler algorithm and modularization

Range Compression

Range compression is done to compress the received pulse along the range direction to concentrate the main energy into a narrower duration. It is performed with a fast convolution between the raw data and a reference signal in the frequency (range) – time (azimuth) domain. Therefore, FFT along the range direction is first performed, followed by matched filter multiplication and range IFFT. Match filtering is implemented as complex multiplication in the frequency domain. Figure 4 shows the implementation flowchart for range compression. Raw data is originally stored in external memory, and to efficiently load/write the data between the external memory and internal

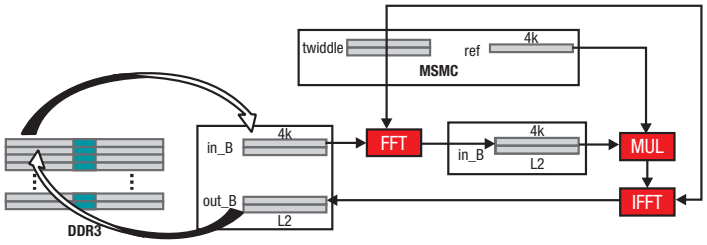

Figure 4: Steps required for range compression

memory (L2), the enhanced direct memory access (EDMA3) is used to service data transfer. EDMA3 is a unique design of TI's architecture. It features fully orthogonal transfer on three dimensions with synchronization on two dimensions, flexible transfer definitions, multiple DMA channels, and memory protection support. More detailed can be found in the TMS320C6678 DSP user guide, **"Enhanced Direct Memory Access 3 (EDMA3) For KeyStone™ Devices,"** available on **ti.com**.

The reference function and twiddle factors for the FFT and IFFT are pre-computed and stored in the shared memory of the TMS320C6678 DSP so they are accessible to all cores. A matrix transpose rearranges the range of compressed data so that it can be read and processed along the azimuth direction for the next steps. The range compressed data is stored in DDR3 grouped into square blocks to optimize DMA transfer efficiency, with block size bounded by the available L2 memory space. The ping-pong strategy is applied to improve efficiency by overlapping the DMA operation and the data processing stage.

Range Cell Migration Correction

Range migration is caused by the range variations due to the platform movement. Figure 5(a) shows the trajectories for three targets with a same closest range distance to the radar in the original time domain. Figure 5(b) shows the corresponding trajectories in the range-Doppler domain, where the three lines collapse into one trajectory. The correction is to rearrange the data in the memory to straighten the trajectory as shown in Figure 5(c), such that azimuth compression can be conducted along each parallel azimuth line. Note that the migration for the whole family of targets with a same range distance is corrected simultaneously. RCMC can be achieved by a range interpolation operation based on an interpolation kernel, such as sinc function or spline.

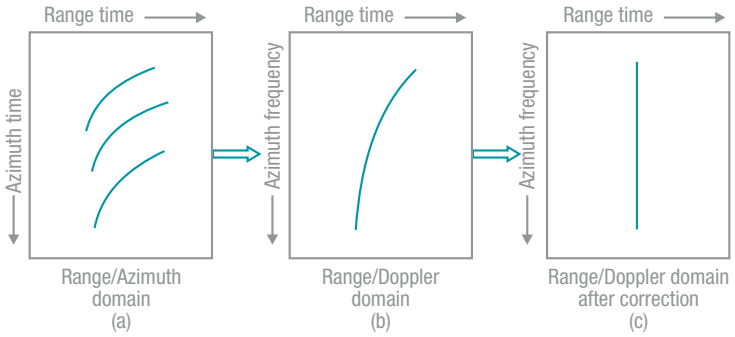

Figure 5: Range cell migration correction for a set of points at the same range distance

For RCMC, data is fetched with ping-pong DMA from the DDR3 to the L2 memory. Each block

corresponds to a certain number of range bins and Doppler frequency bins. The integer part of the migration amount is used for range sample shift, while the fractional part is used for interpolation. In this implementation, the 16-set 8-tap sinc filter is adopted as the interpolation filter. The coefficients of the 16 sets of filters are stored as a constant number. The index of which filter should be selected is determined by the fractional part. The same filter is used for each range cell. The TMS320C6678 DSP's unique double-float load, write and arithmetic instructions are utilized to improve the interpolation computation efficiency.

Azimuth Compression

The azimuth compression step compresses the spread energy in the trajectory to a single cell in the azimuth direction. This procedure is similar to range compression except that the azimuth reference function is range dependent. In other words, the azimuth reference function at each range line is different, which leads to a more complicated procedure compared to the range compression. Similar to RCMC, azimuth compression is also performed in range-Doppler domain. The final image is obtained by transforming the azimuth-compressed signal back to the time domain, followed by some post-processing steps.

Multicore Mapping

The parallel implementation using the eight cores on the TMS320C6678 DSP platform can be achieved by letting each core process a different portion of the data. Figure 6 demonstrates how the task is assigned to each core. The data in the external memory is divided into eight portions. Each core retrieves the data according to the start point of the allocated location through DMA. Within each core, the above mentioned compression and corner turning steps can be implemented accordingly using local memory devices (L2 and L1). The access to the DDR3 memory among multiple cores is scheduled by the DMA controller.

Since the DDR3 memory space is shared fully across all cores, the division of computation across cores can be easily achieved through OpenMP constructs. The OpenMP runtime on the multi-core DSP will run multiple threads (one thread per core) with their own allocated data portion.

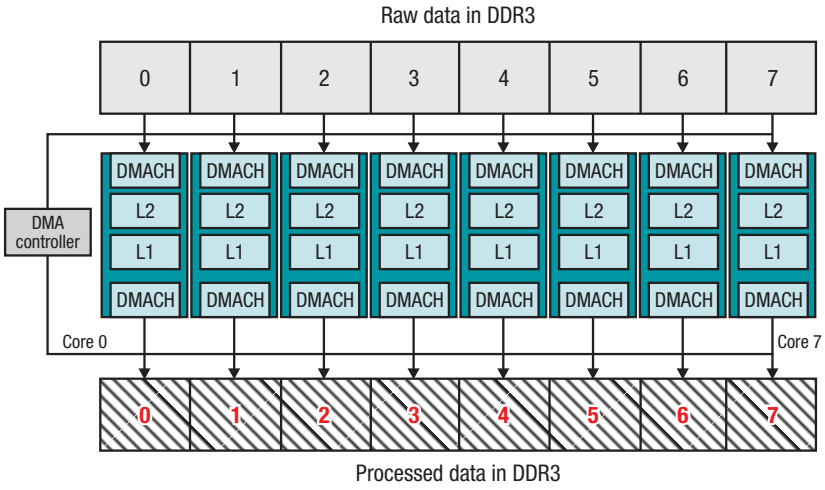

Figure 6: Mapping of data onto the 8 cores of the TMS320C6678 DSP

Evaluation

In order to evaluate the performance of the SAR implementation on the TMS320C6678 DSP, the five key portions of the algorithm (see Figure 3) are profiled to estimate the execution time of each module, and of the SAR focusing

overall. Specifically, the execution time for range compression, corner turning, azimuth FFT, RCMC and azimuth compression are evaluated. Two typical examples of image sizes, 2048 by 2048 and 4096 by 4096, are used for this demo code example. This results in 2048 (4096) FFTs and IFFTs with 2048 (4096) points in each transformation.

1MB of the shared memory is set as non-cached to avoid the cache incoherency among the eight cores. For each core, part of the L2 memory is set as L2 SRAM (384 KB) and the rest is used as cache (128 KB). Intermediate ping-pong buffers are allocated in the L2 SRAM. The batch size, i.e., the number of rows loaded from DDR3 each time is determined by the available L2 SRAM. For the 2048- and 4096-point FFT, it is set to be two and one, respectively. The squared block size during corner turning is set as 64 × 64.

Table 1and Table 2 show the results of profiling the five modules with different image sizes on a TMS320C6678 DSP-based evaluation module (EVM). Results are presented with 1, 4 and 8 cores of the TMS320C6678 DSP used. When comparing the results from the two different image sizes, the computation time scales well with the image size. As expected, the timing required for range compression and azimuth compression also scales very well with the increase of the number of operational cores. On the other hand, the corner turning timing, RCMC and azimuth FFT saturate at around 4 cores due to the fact that these steps are memory I/O bound. Once the DDR3 bandwidth is saturated, increasing the number of operational cores does not improve the computation speed. For the total execution time, the acceleration factor with 8 cores relative to a single core is around 6.

Figure 7 on the following page shows the percentage of required processing time for each step in the case of the 4096 by 4096 data set. The range compression and RCMC are the most computationally intensive steps, and take the longest portions of the overall execution time. The sum of portions for azimuth FFT and azimuth compression is similar to that of the range compression step on its own. Overall, it takes around 0.25 second to process the whole 4096 by 4096 image using all 8 cores of the TMS320C6678 DSP in parallel, making it useful for achieving real-time operation, which is critical in many avionic and defense applications. Given that this processing is done with a 10-W device, the above demonstrated result makes the TMS320C6678 DSP very competitive among other alternatives, such as general-purpose GPUs (GPGPUs) and CPUs, which

Number of active cores	Range compression (ms)	Corner turn (ms)	Azimuth FFT (ms)	RCMC (ms)	Azimuth compression (ms)	Total (ms)
1	142	17	34	46	44	283
4	36	6	9	13	11	74
8	18	6	7	12	6	50

Table 1: Execution time for modules of the SAR algorithm with image size of 2K × 2K

Number of active cores	Range compression (ms)	Corner turn (ms)	Azimuth FFT (ms)	RCMC (ms)	Azimuth compression (ms)	Total (ms)
1	573	100	199	269	263	1404
4	144	33	50	71	66	363
8	72	34	45	66	33	251

Table 2: Execution time for modules of the SAR algorithm with image size of 4K × 4K

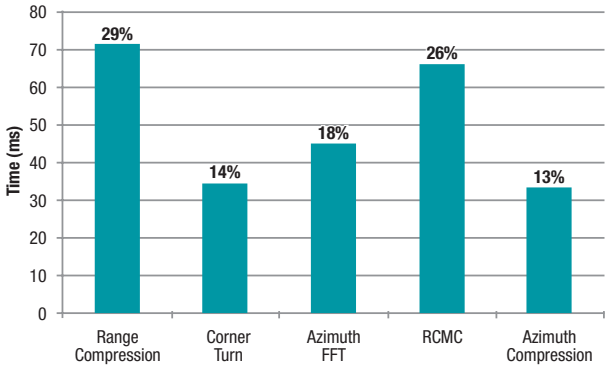

Figure 7: Time consumption of each module as percentage of total time for 4K × 4K image processing

commonly achieve similar processing speeds, but with greatly increased power consumption. Additionally, since the SAR processing is embarrassingly parallel, multiple TMS320C6678 devices can be employed to further improve the throughput, connected via HyperLink, or other high-throughput connections.

Conclusions

The performance per power efficiency of the multi-core DSP, TMS320C6678, makes it a good choice for various computationally intensive applications, such as synthetic aperture radars and others that use similar signal processing techniques, such as FFTs and matrix manipulations and transposes.

This device is widely used in various embedded applications including radio controllers, military and civilian radars, and industrial and medical imaging devices. The benchmarking results of the Range-Doppler SAR processing algorithm shows that real-time SAR processing is feasible at high power efficiency using this multi-core DSP device, TMS320C6678. Additionally, industry-standard software frameworks such as OpenMP can be leveraged to take advantage of the multi-core DSP by parallelizing algorithms quickly. The scalability of SAR operations across multiple devices makes the algorithm even more well-suited for DSPs to provide highly-efficient high-performance processing for the wide variety of SAR, and other, applications.

See the paper "Synthetic Aperture Radar on Low Power Multi-Core Digital Signal Processor" from the IEEE High Performance Extreme Computing conference for more details on this implementation[1].

Resource

[1] D. Wang, **"Synthetic Aperture Radar on Low Power Multi-Core Digital Signal Processor,"** in *IEEE High Performance Extreme Computing*, Boston, 2012.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265