

**Product Bulletin**

# TMS320<sup>TM</sup> DSP Algorithm Standard

**Key Features and Benefits**

**New**

- **Algorithm Developer's Kit v2.51 Available Now** – Enables quicker and easier algorithm development and facilitates usage and testing
- Allows mixing of interoperable components
- Lowers support and development costs by eliminating custom coding
- Reduces system integration time
- Enables creativity and innovation
- Choose from hundreds of compliant algorithms available today

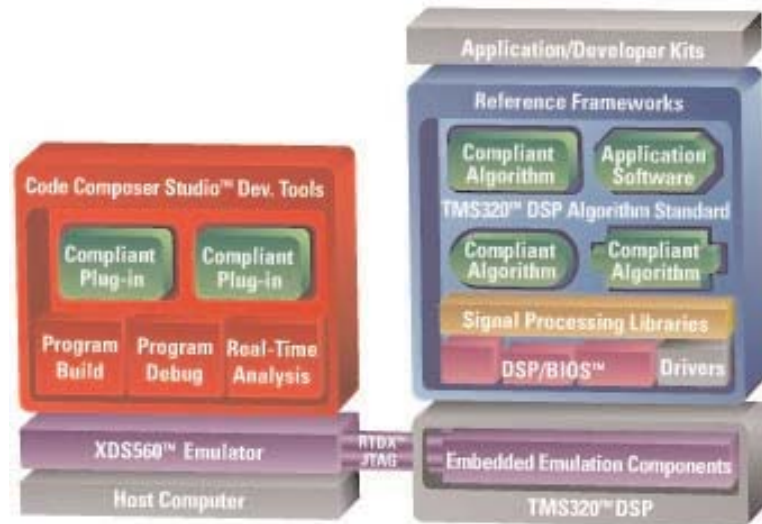
Texas Instruments' real-time eXpressDSP<sup>TM</sup> Software and Development Tools is a premier open software environment for TI DSPs. With eXpressDSP, a simplified and streamlined development infrastructure helps reduce development time. Programmers spend more time creating innovative applications rather than working on cumbersome and repetitive development and integration tasks.

A key component of eXpressDSP Software is the TMS320<sup>TM</sup> DSP Algorithm Standard, also known as XDAIS. This standard frees DSP designers from an exclusive "custom-crafted" approach to a new paradigm of interoperable components. A significant base of system-ready algorithms allows real "make vs. buy" decisions.

**Where DSP Comes Together**

The TMS320 DSP Algorithm Standard is part of the overall eXpressDSP Software technology initiative. eXpressDSP Software includes four major elements: DSP/BIOS<sup>TM</sup>, a scalable real-time kernel; eXpressDSP Reference Frameworks, the industry's first getting started software for DSP

**eXpressDSP Software and Development Tools**



*The TMS320 DSP Algorithm Standard, part of the eXpressDSP Software and Development Tool offering, enables system-ready algorithms for DSP designs.*

applications; TMS320 DSP Algorithm Standard, a set of rules and guidelines for developers to create standard algorithms, and TI's DSP Third-Party Network with its large base of TI DSP-based software modules.

**TMS320 DSP Algorithm Standard**

The standard is a set of coding conventions for algorithm writers

that reduces time-consuming system integration for anyone trying to put algorithms into their system. This is achieved by defining common programming rules and guidelines with a set of programming interfaces that are consistently used by algorithms across a wide variety of applications.

## A DSP First: Interface Standards to Enable System-Ready Software Algorithms

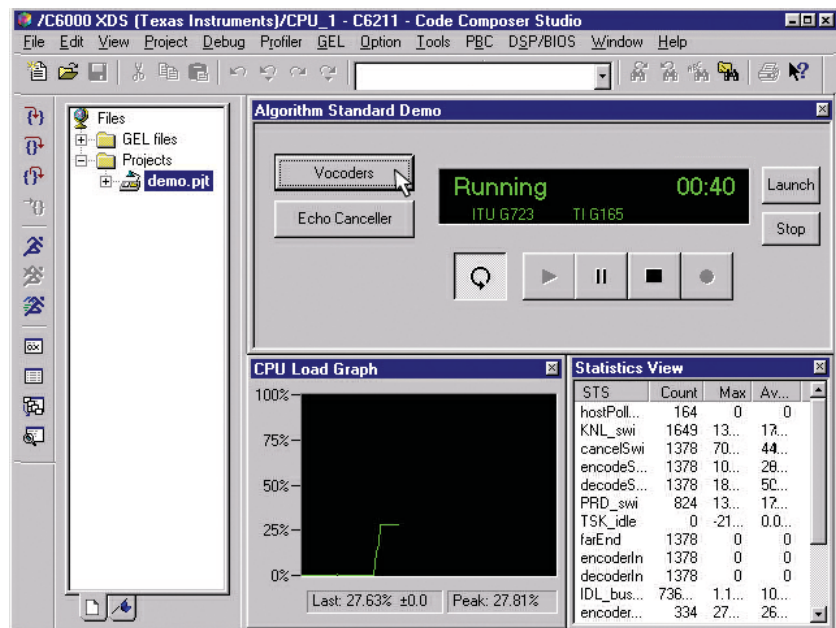
Previously, the DSP market suffered from a lack of standardization in software algorithms, which enable DSPs to perform complex signal-processing functions. Because there were no standards to ensure a consistent programming interface, DSP design engineers faced time-consuming problems when integrating existing algorithms into new or different applications. Frequently, they focused valuable resources on re-engineering code. The lack of standards made the integration of algorithms from more than a single source very difficult. As a result, integration times are extended, debugging is tricky, and it is difficult or impossible to compare two similar algorithms. By publishing and supporting the TMS320™ DSP Algorithm Standard, TI aids in decreasing software system integration and debug time.

### The TMS320 DSP Algorithm Standard Solves Integration Issues

The TMS320 DSP Algorithm Standard Developer's Kit provides all of the information necessary to enable application developers and system integrators to understand and utilize algorithms that are compliant to the standard. Information and tools are also provided for generations of new algorithms that are compliant to the standard.

### The Result of the Standard: Hundreds of Compliant Algorithms Available Today

More than 100 third-party algorithm developers have been working with the TMS320 Algorithm Standard for more than three years and have developed over 650 algorithms that comply with the standard. These algorithms are designed for ease of integration and to reduce the amount of



The Developer's Kit demo shows how easily eXpressDSP-compliant algorithms interoperate.

time spent debugging the system as algorithms are added. Some examples of algorithms available today are video, imaging, voice, speech, hearing, biometrics, telephony, wireless and control. Algorithms are available for all three TI DSP architectures: TMS320C2000™, TMS320C5000™, and TMS320C6000™ DSPs. For the most up-to-date listing of eXpressDSP™-compliant algorithms, check TI's DSP Village at [www.dspvillage.com](http://www.dspvillage.com)

### Compliance Program

TI offers an eXpressDSP-compliance testing program. Algorithm developers submit algorithms to TI for test. Algorithms that pass the test suite and meet the requirements stated in the standard specification are allowed to display the eXpressDSP-compliant mark. Only eXpressDSP-compliant



## Scope of the Standard

### Mandatory Rules

The standard consists of the following:

- 46 basic "common sense" rules for all algorithms
- IALG APIs abstract DSP memory management away from algorithms to the controlling framework.
- IDMA2 APIs abstract DSP DMA management away from algorithms to the controlling framework. IDMA2 replaces the previous IDMA APIs provided in earlier versions of the standard.
- Instruction Set Architecture (ISA) rules for all families of TMS320 DSPs
- Naming conventions to reduce name space pollution

### Optional Guidelines

- 19 additional guidelines that algorithm developers are strongly encouraged to follow. The implementation of these guidelines provide additional assistance to the system integrator.
- Methods for extending baseline API for added functionality – allowing for vendor differentiation.

algorithms are available for sale through the eStore. With the introduction of Algorithm Developer's Kit version 2.51, the testing program, called QualiTI, is also available to any Code Composer Studio™ (CCStudio) user to check whether their own or a supplied algorithm meets the eXpressDSP-compliance requirements.

## Tools and Software Homepage

TI's DSP Village is the place to look for more information or technical support for any of TI's software products. There is a discussion group, DSP KnowledgeBase, DSP Hotline Online, application notes, user manuals and much more. Visit [www.dspvillage.com](http://www.dspvillage.com) to get a jump-start on application development.

## Training

Training is available both online and in-person. Online there are two different courses. A half-hour overview training course covers the high-level objectives of the standard and how it fits into the total eXpressDSP™ Software picture. A more technical two-hour online training course is also available. This covers not only high-level objectives of the standard, but also discusses the eXpressDSP demonstration software, the rationale behind the XDAIS rules, and finally, some of the issues that relate to the overall software make vs. buy decision process. Both online training classes can be accessed from the Algorithm Standard section in

[www.dspvillage.com](http://www.dspvillage.com)

If you have a larger team or group, TI's Technical Training Organization offers a complete three-day course at the facilities of your choice. This course goes into detail of both the producer and consumer aspects of the TMS320 DSP™ Algorithm Standard. For further information visit [www.ti.com/softwaretraining](http://www.ti.com/softwaretraining)

# TMS320™ DSP Algorithm Standard

## Mandatory Rules

1. All algorithms must follow the run-time conventions imposed by TI's implementation of the C programming language.
2. All algorithms must be reentrant within a preemptive environment (including time-sliced preemption).
3. All algorithm data references must be fully relocatable (subject to alignment requirements). That is, there must be no "hard-coded" data memory locations.
4. All algorithm code must be fully relocatable. That is, there can be no hard-coded program memory locations.
5. Algorithms must characterize their ROM-ability; i.e., state whether they are ROM-able or not.
6. Algorithms must never directly access any peripheral device. This includes, but is not limited to on-chip DMAs, timers, I/O devices and cache control registers.
7. All header files must support multiple inclusions within a single source file.
8. All external definitions must be either API identifiers or API and vendor prefixed.
9. All undefined references must refer either to the operations specified in Appendix B (a subset of C run-time support library functions and the DSP/BIOS™ Kernel) or other eXpressDSP-compliant modules.
10. All modules must follow the naming conventions of the DSP/BIOS Kernel for those external declarations disclosed to the client.
11. All modules must supply an initialization and finalization method.
12. All algorithms must implement the IALG interface.
13. Each of the IALG methods implemented by an algorithm must be independently relocatable.
14. All abstract algorithm interfaces must derive from the IALG interface.
15. Each eXpressDSP-compliant algorithm must be packaged in an archive which has a name that follows a uniform naming convention.
16. Each eXpressDSP-compliant algorithm header must follow a uniform naming convention.
17. Different versions of a standard-compliant algorithm from the same vendor must follow a uniform naming convention.
18. If a module's header includes definitions specific to a "debug" variant, it must use the symbol `_DEBUG` to select the appropriate definitions; `_DEBUG` is defined for debug compilations and only for debug compilations.
25. All C6000™ algorithms must be supplied in little endian format.
26. All C6000 algorithms must access all static and global data as far data.
27. C6000 algorithms must never assume placement in on-chip program memory; i.e., they must properly operate with program memory operated in cache mode.
28. On processors that support large program model compilation, all core run-time support functions must be accessed as far functions; for example, on the C54x™ DSP, the calling function must push both the XPC and the current PC.
29. On processors that support large program model compilation, all algorithm functions must be declared as far functions; for example, on the C54x DSP, callers must push both the XPC and the current PC and the algorithm functions must perform a far return.
30. On processors that support an extended program address space (paged memory), the code size of any object file should never exceed the code space available on a page when overlays are enabled.
31. All C55x™ algorithms must document the content of the stack configuration register that they follow.
32. All C55x algorithms must access all static and global data as far data; also the algorithms should be instantiable in a large memory model.
33. C55x algorithms must never assume placement in on-chip program memory; i.e., they must properly operate with program memory operated in instruction cache memory.
34. All C55x algorithms that access data by B-bus must document: the instance number of the IALG\_MemRec structure that is accessed by the B-bus (heap-data) and the data-section name that is accessed the B-bus (static-data).
35. All C28x™ algorithms must access all global and static data as far data; also, the algorithm should be instantiable in a large memory model.

## eStore

TI's eStore is your one-stop shop for eXpressDSP-compliant algorithms. All algorithms listed in the eStore have passed compliance testing and adhere to the standard. The eStore is also home to TI tools and software. Visit the eStore at [dspvillage.ti.com](http://dspvillage.ti.com)

## Algorithm Standard Developer's Kit

Version 2.51 of the Algorithm Developer's Kit is now available. Either download it from the [www.dspvillage.com](http://www.dspvillage.com) web site or via the CCStudio Update Advisor. It is designed to run on CCStudio v2.1 or later.

### New tools include:

- **QualiTI** – eXpressDSP™ compliance self-compliance checking tool.  
Code Composer Studio™ Development Tool users save time by testing algorithms for compliance to the TMS320™ DSP Algorithm Standard.
- **DOSA** – Algorithm Optimization tool for Static Applications. Quickly facilitates the systematic process of “stripping out” unnecessary elements from a fully dynamic algorithm for use in a static system.
- **Hyperception Component Wizard™** – Automatic XDAIS wrapper generation. Step-by-step wizard leads the algorithm writer through the process of automatically creating a pre-built XDAIS wrapper and also prevents the user from using non-conforming entries.
- **DSP Support**  
TMS320C5000™, TMS320C6000™, and TMS320C2000™ DSP platforms fully supported.

## Technical Documentation

Most existing application notes have been updated, including the standard itself, SPRU352. In addition,

## TMS320 DSP Algorithm Standard (Continued)

### Performance Characterization Rules

19. All algorithms must characterize their worst-case heap data memory requirements (including alignment).
20. All algorithms must characterize their worst-case stack space memory requirements (including alignment).
21. Algorithms must characterize their static data memory requirements.
22. All algorithms must characterize their program memory requirements.
23. All algorithms must characterize their worst-case interrupt latency for every operation.
24. All algorithms must characterize the typical period and worst-case execution time for each operation.

### DMA Rules

- DMA Rule 1: All data transfer must be completed before return to caller.
- DMA Rule 2: All algorithms using the DMA resource must implement the IDMA interface.
- DMA Rule 3: Each of the IDMA2 methods implemented by an algorithm must be independently relocatable.
- DMA Rule 4: All algorithms must state the maximum number of concurrent DMA transfers to each logical channel.
- DMA Rule 5: All algorithms must characterize the average and maximum size of the data transfers per logical channel for each operation. Also, all algorithms must characterize the average and maximum frequency of data transfers per logical channel for each operation.
- DMA Rule 6: C6000™ DSP algorithms must not issue any CPU read/writes to buffers in external memory that are involved in DMA transfers. This also applies to the input buffers passed to the algorithm through its algorithm interface.
- DMA Rule 7: If a C6000 DSP algorithm has implemented the IDMA2 interface, all input and output buffers residing in external memory and passed to this algorithm through its function calls, should be allocated on a cache line boundary and be a multiple of the cache line length in size. The application must also clean the cache entries for these buffers before passing them to the algorithm.
- DMA Rule 8: For C6000 DSP algorithms, all buffers residing in external memory involved in a DMA transfer should be allocated on a cache line boundary and be a multiple of the cache line length in size.
- DMA Rule 9: C6000 DSP algorithms should not use stack allocated buffers as the source or destination of any DMA transfer.
- DMA Rule 10: C55x DSP algorithms must request all data buffers in external memory with 32-bit alignment and sizes in multiples of 4 (bytes).
- DMA Rule 11: C55x DSP algorithms must use the same data types, access modes and DMA transfer settings when reading from or writing to data stored in external memory, or in application-passed data buffers.

tion, three new application notes have been added. To access these application notes separately from the Development Kit, use the Technical Documentation link in the Related Links box of the DSP Algorithm Standard page on the DSP Village.

## For More Information

For more information or to order the TMS320 Algorithm Standard,

visit the TI DSP Village web site at [www.dspvillage.com](http://www.dspvillage.com)

# TI Worldwide Technical Support

---

## Internet

**TI Semiconductor Product Information Center Home Page**  
support.ti.com

**TI Semiconductor KnowledgeBase Home Page**  
support.ti.com/sc/knowledgebase

## Product Information Centers

### **Americas**

Phone +1(972) 644-5580  
Fax +1(972) 927-6377  
Internet/Email support.ti.com/sc/pic/americas.htm

### **Europe, Middle East, and Africa**

Phone  
Belgium (English) +32 (0) 27 45 55 32  
Finland (English) +358 (0) 9 25173948  
France +33 (0) 1 30 70 11 64  
Germany +49 (0) 8161 80 33 11  
Israel (English) 1800 949 0107  
Italy 800 79 11 37  
Netherlands (English) +31 (0) 546 87 95 45  
Spain +34 902 35 40 28  
Sweden (English) +46 (0) 8587 555 22  
United Kingdom +44 (0) 1604 66 33 99  
Fax +(49) (0) 8161 80 2045  
Email epic@ti.com  
Internet support.ti.com/sc/pic/euro.htm

### **Japan**

Fax International +81-3-3344-5317  
Domestic 0120-81-0036  
Internet/Email International support.ti.com/sc/pic/japan.htm  
Domestic www.tij.co.jp/pic

### **Asia**

Phone  
International +886-2-23786800  
Domestic Toll-Free Number  
Australia 1-800-999-084  
China 108-00-886-0015  
Hong Kong 800-96-5941  
Indonesia 001-803-8861-1006  
Korea 080-551-2804  
Malaysia 1-800-80-3973  
New Zealand 0800-446-934  
Philippines 1-800-765-7404  
Singapore 800-886-1028  
Taiwan 0800-006800  
Thailand 001-800-886-0010  
Fax 886-2-2378-6808  
Email tiasia@ti.com  
Internet support.ti.com/sc/pic/asia.htm

**Important Notice:** The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

Real World Signal Processing, the Black/Red banner, TMS320, TMS320C2000, TMS320C5000, TMS320C6000, C2000, C5000, C6000, Code Composer Studio, and DSP/BIOS are trademarks of Texas Instruments. All other trademarks are property of their respective owners.

**A010203**