Product Bulletin

DSP/BIOS™ Kernel Scalable, Real-Time Kernel for TMS320™ DSPs

Key Features:

- Fast, deterministic real-time kernel
- Scalable to very small footprint
- Tight integration with Code Composer Studio IDE
- Integrated real-time analysis tools
- Integrated peripheral configuration and management
- Royalty-free and proven in thousands of designs

DSP/BIOS is a real-time kernel optimized to meet the specific needs of applications based on the Texas Instruments TMS320C5000™ and TMS320C6000™ platform. DSP/BIOS is royalty-free and an core component of the Code Composer Studio integrated development environment. Using DSP/BIOS's robust multithreading and analysis services, developers can achieve faster time-to-market and produce well-structured applications that are easier to upgrade and maintain. DSP/BIOS has been proven in thou- sands of DSP applications and its technology has been refined by over ten years of development.

Why DSP/BIOS

Traditionally, many DSP developers have structured their software applications around sequential processing loops and state machines. While this approach is perfectly adequate when the DSP is performing one main function, it has significant limitations once the DSP must perform multiple functions. As it becomes more common for DSPs to support multiple functions operating at different sampling rates, developers are adopting the multithreaded design approach that is well-established in microprocessor-based real-time applications. In a sequential processing loop, the addition of a new function or the modification of an existing function affects the times at which other functions are serviced. As a result, more complex real-time applications designed in this way often become very difficult to maintain and modify.

For multi-function applications, or applications to which more functions will be added in future versions, a better approach is to use a design paradigm that makes response-times for each function relatively independent of the implementation of other functions. Multithreaded applications accomplish this by assigning each discrete system function its own execution thread. These threads execute in parallel, with a scheduler controlling when a particular thread is given CPU time. Each thread is a given a priority that determines how quickly it will be scheduled once it is ready to run, since the scheduler will preempt a lower-priority thread if a higher-priority thread becomes ready to execute. As a result, the response time of a critical function executed in a high-priority thread will not be affected by modifications to other functions or the addition of new lower-priority functions.

DSP/BIOS: Optimized for Real-Time and Embedded DSP Applications

Most real-time operating systems are primarily designed for microprocessor-based applications. In contrast, DSP/BIOS has been designed from the ground up to meet the specific needs for DSP applications:

- **Limited Memory Size:** In DSP applications, it is often critical to be able to fit the application entirely in on-chip memory for performance reasons. Since many DSP chips are limited to 16, 32 or 64K words of on-chip memory, DSP/BIOS is designed to provide useful configurations in as little as 1K works of memory.

- **Extreme Real-Time Performance Constraints:** DSPs must often process streams of continuous real-Time events. DSP/BIOS provides very lightweight threads that provide faster context switch times than conventional tasks.

- **Multi-rate Applications:** DSP applications must often process a datastream at a regular sample rate. In addition, many systems have different functions sampling at multiple rates. DSP/BIOS includes time-triggered threads that simplify architechting an application to process multi-rate data.

In addition, use of a multi-threading kernel may reduce power consumption. Because multithreading applications can be completely interrupt-driven, it is possible to eliminate polling from the application. This lowers processor utilization and enables power-saving modes to be invoked during idle periods.

Scalable Run-Time Services

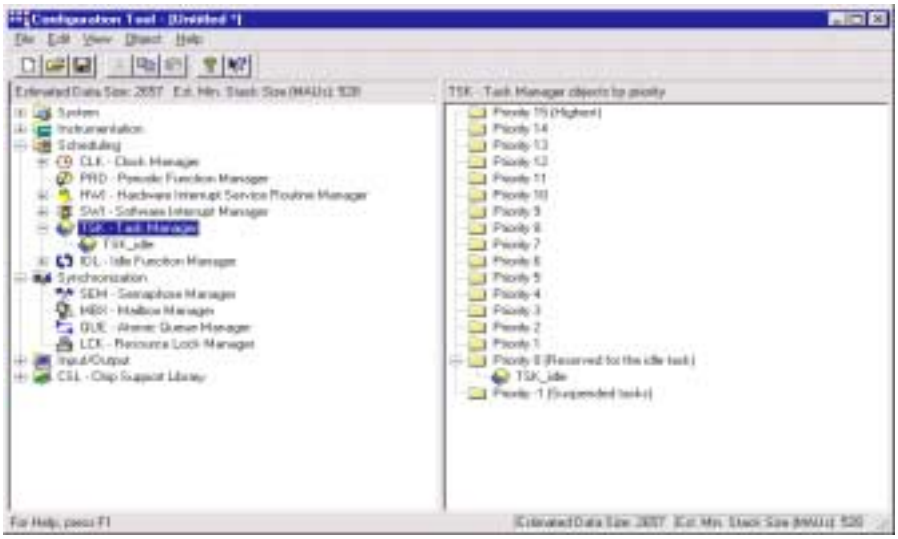
DSP/BIOS™ addresses the need of DSP application developers through a set of scalable run-time services including:

- Graphical configuration
- A deterministic, preemptive multithreading scheduler
- Interprocess communication
- Dynamic memory management
- Interrupt management
- Real-time analysis and logging
- Peripheral device management

DSP/BIOS is available exclusively for the TMS320C5000™ and TMS320C6000™ and is highly optimized for these architectures, with all performance-critical code sections hand-written in assembly language.

Static Configuration Reduces OS Memory Footprint

Like all conventional real-time operating systems, DSP/BIOS enables an application to dynamically create operating system objects, such as tasks or semaphores, at any time during program execution. However, in reality many real-time applications simply create all the required operating system objects at the start of the application. The drawback of this approach is that the code for creating the operating system objects must be present in target memory even though it is only used memory even though it is only used once. In addition to dynamic creation, DSP BIOS provides a host-based graphical configuration tool that enables developers to statically generate a configuration DSP/BIOS tailored to the needs of their application. This significantly reduces target memory footprint by eliminating the need to include code that creates operating system objects.



The DSP/BIOS configuration tool provides point-and-click configuration of operating system services to achieve a minimal footprint.

To further reduce memory footprint, DSP/BIOS provides more granular configurability when compared to conventional real-time operating systems. DSP/BIOS configurations that support scheduling services start as low as 1K words.

DSP/BIOS kernel services are ROMable to provide further options for reduced memory usage. The kernel functions required by the application may be placed in on-chip ROM using customer DSP masks or off-chip ROM when available.

Kernel Configuration	5402 (in 16-bit words)		6211 (in 8-bit bytes)	
	Code	Data	Code	Data
3 Software Interrupts, Interrupt Vector Table	1040	158	5940	532
4 Software Interrupts, a Periodic Function, 2 Pipes, Real-time Analysis, RTDX, Real-time Clock, Interrupt Vector Table	3178	897	14812	3132
4 Tasks, 2 Software Interrupts, a Mailbox, a Semaphore, 2 Pipes, Real-time Analysis, RTDX, Real-time Clock, Interrupt Vector Table	5519	1242	24024	4393

DSP/BIOS memory requirements: The table illustrates the memory footprint of three different DSP/BIOS configurations. The code size numbers also include constant data. The data size numbers do not include system or task stacks since these are application-specific.

Deterministic Kernel Delivers Real-Time Performance

DSP/BIOS™ provides a rich set of real-time kernel services, including multiple scheduling mechanisms that enable developers to create sophisticated applications without compromising real-time deadlines. Because predictable performance is critical in real-time applications, most DSP/BIOS system calls are deterministic and therefore guaranteed to complete execution within a fixed time. Only functions, such as task creation and deletion, that are typically not used in time-critical code sections are non-deterministic. Many DSP applications have some sections coded in assembly language, its possible to invoke many DSP/BIOS services either as C functions or through an assemble language macro interface.

To provide the fast response required by DSP applications, DSP/BIOS includes Software Interrupts (SWIs) and Periodic Functions (PRDs), in addition to conventional tasks. SWIs are lightweight preemptible threads that share a common stack. This results in lower memory overhead and faster context switch times, as there is no need to save and restore a task stack. PRDs are time-triggered high-priority threads that can be easily set up to process samples of data arriving at fixed time intervals, simplifying the design of multi-rate systems. To facilitate the design of more complex applications, DSP/BIOS provides a rich set of intertask communication services, including semaphores, mailboxes, and queues.

In addition to scheduling and intertask communications services, DSP/BIOS also provides real-time clock management, memory - management, and device-independent I/O. The DSP/BIOS memory manager provides re-entrant functions for dynamic allocation and freeing of memory. Because DSP applications typically have a fragmented memory configuration that utilizes both on-chip and off-chip RAM, the memory manager provides logical segments that abstract away the exact location of the physical memory. DSP/BIOS provides a device-independent I/O model that is optimized for efficient processing of real-time streams of data. By abstracting away specific dependencies on peripheral devices and memory configurations, DSP/BIOS makes it simpler to port applications to new DSP processors or board designs.

Kernel Module	Description
Hardware Interrupts	Interface from hardware interrupts to DSP/BIOS
Software Interrupts	Lightweight preemptible threads that use program stack, but cannot yield
Tasks	Independent threads of execution that can yield the processor
Periodic Functions	Time-triggered lightweight threads
Mailboxes	Synchronized data exchange between tasks
Lock	Nestable binary samaphores
Semaphores	Counting semaphores
Queues	Atomic Link Lists
Clock	Interface to hardware timers
Streams	Streaming I/O for tasks
Pipes	Streaming I/O for software interrupts
Memory Manager	Low overhead dynamic memory allocation

DSP/BIOS provides a rich set of kernel functions to support real-time applications.

Kernel Operation	C5000 CPU Cycles	Time (usec) for 100 MHz 5410	C6000 CPU Cycles	Time (usec) for 200 MHz 6201
Task Yield				
TSK_Yield	266	2.66	263	1.315
Semaphores				
Post Semaphore, No Task Switch	173	1.73	182	0.91
Post Semaphore, Task Switch	297	297	288	1.44
Pend Semaphore, No Task Switch	182	1.82	152	0.76
Pend Semaphore, Task Switch	325	3.25	275	1.39
Software Interrupts				
Post of Software Interrupt, No Context Switch	80	0.80	118	0.59
Post of Software Interrupt, With Context Switch	191	1.91	238	1.19
Periodic Functions				
Interrupt Calling PRD_Tick	126	1.26	113	0.565
Interrupt Calling PRD_SWI	379	3.79	360	1.80
Interrupt to Periodic Function	468	4.68	471	2.355
Hardware Interrupts				
Interrupt Prolog (Minimum)	39	0.39	32	0.16
Interrupt Prolog for Calling C Function	51	0.51	41	0.205
Interrupt Epilog (Minimum)	47	0.47	52	0.26
Interrupt Epilog Following C Function Call	59	0.59	65	0.325
Hardware Interrupt to Blocked Task	767	7.67	798	39.9
Hardware Interrupt to Software Interrupt	305	3.05	336	1.68
Interrupt Latency	45	0.45	72	0.36
Mailboxes				
Post Mailbox, No Task Switch	494	4.94	431	2.155
Post Mailbox, Task Switch	901	9.01	800	4.0
Pend Mailbox, No Task Switch	495	4.95	433	2.165
Pend Mailbox, Task Switch	357	3.57	300	1.5

DSP/BIOS performance: The table illustrates the performance of selected DSP/BIOS kernel functions on the TMS320C5000™ and TMS320C6000™ architectures. All measurements were made using DSP/BIOS for Code Composer Studio™ IDE v.1.2 and use the C language interface except for the hardware interrupts.

Flexible Interrupt Management

To efficiently handle interrupts, DSP/BIOS™ provides an interrupt dispatcher. The dispatcher performs operations such as context saving and restoration and temporary disabling of the scheduler to ensure interrupt service routines (ISR) can interact correctly with the kernel. This approach reduces code size by eliminating the need to specifically add such code to each ISR. A further benefit is that the ISRs may be written in C because the interrupt directly vectors to the dispatcher which can then call a standard C function as the interrupt handler once all the save/restore and other operations are completed. All ISRs use a common stack rather than the stack of the task they interrupt. This reduces program RAM requirement by eliminating the need to allow for maximum interrupt nesting on each task stack.

DSP/BIOS also provides macros- one for entry and one for exit – that may be added to an ISR to perform The same operations as the dispatcher. Since the macros are added to each ISR, this approach increases memory requirements. However, the macros enable the developer to optimize register save and restore operations to only those used by that specific ISR, thus minimizing interrupt response time. DSP/BIOS enables both the macros and dispatcher to be used in the same application to achieve the optimal balance of performance and memory footprint.

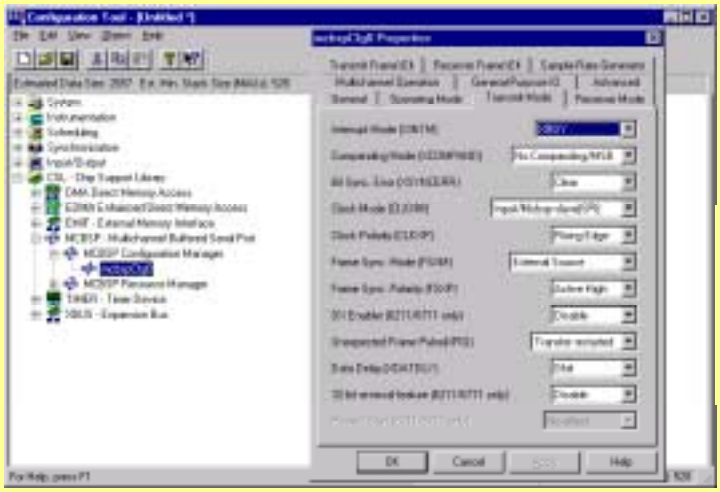
In both the dispatcher and macro implementations, great care has been taken to minimize the periods for which interrupts are disabled. This minimizes interrupt latency and enables fast response times to critical interrupts.

Simplified Peripheral Configuration

For each supported processor, DSP/BIOS™ includes a set of peripheral configuration and management functions known as the Chip Support Library (CSL). The CSL supports on-chip peripherals for all C5000™ and C6000™ devices.

The CSL is fully integrated into the DSP/BIOS configuration tool, which greatly simplifies configuration of complex peripheral control registers. Rather than having to painstakingly calculate bit-maps, a developer can graphically select individual register flags and set them to the correct value. The configuration tool then generates the appropriate CSL calls to configure the registers. The CSL also provides a set of functions and macros that facilitates writing programs to configure and manage peripherals meaningful names for register flag values and convenience utilities to generate the register bit mask from a list of separate flag values, CSL provides resource management. This prevents a developer from inadvertently assigning a resource, such as a DMA channel, for more that one operation simultaneously.

The Chip Support Library



The Chip Support Library supports graphical configuration of DSP peripherals such as the multi-channel serial port, eliminating the need to remember and calculate bit-field values

CSL Module	Description
Cache	Cache configuration and management
DAT	Device-independent DMA operations (will work on both DMA & EDMA devices)
DMA	DMA operations
EDMA	Enhanced DMA operations
EMIF	Configuration of Extended Memory Interface Registers
HPI	Configuration of Host Processor Interface port registers
IRQ	Interrupt configuration & management
MCBSP	Configuration of Multi-Channel Buffered Serial Port
PWR	Configuration of power-down control registers and invocation of power-down modes
TIMER	Configuration of timer registers

The Chip Support Library supports configuration of standard TMS320C5000 and TMS320C6000 on-chip peripherals.

Integrated Analysis and Debugging Tools Reduce System Integration Time

DSP/BIOS™ is tightly integrated into the Code Composer Studio™ Interactive Development Environment (IDE). As a result, DSP/BIOS developers have access to the industry's most powerful DSP development tools. Code Composer Studio's Visual Edit tool provides automatic highlighting and expansion of DSP/BIOS system calls. Developers can use Code Composer Studio's kernel object browser to examine the state of operating system objects, such as tasks and semaphores, enabling them to better understand the interaction between the application and the operating system and produce an optimal software architecture.

The kernel object browser also provides data on stack usage and indicates whether stack overflow or underflow may have occurred. This information enables developers to optimize stack size to save memory and to easily detect stack overflow problems, which are notoriously difficult to debug.

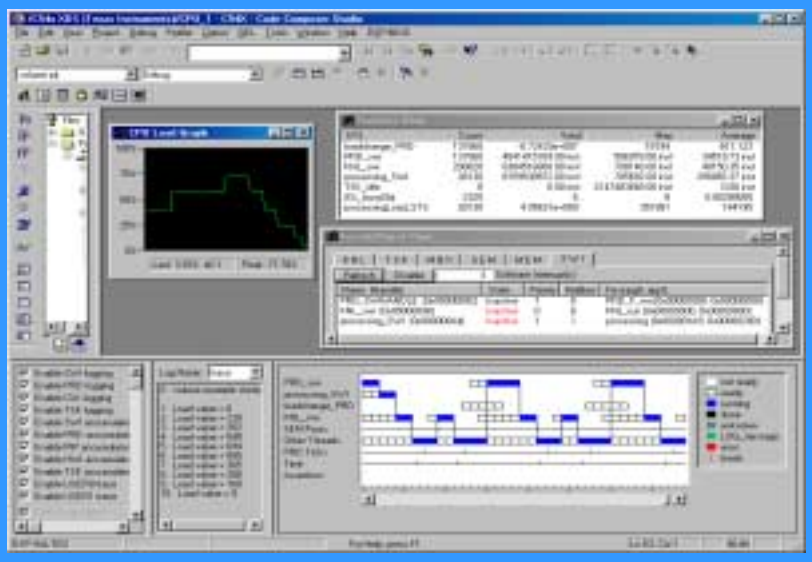
As applications become more complex, the system integration process has become more difficult as developers need to understand the interaction of multiple functions before they can successfully resolve problems or tune overall performance. DSP/BIOS provides powerful system-level analysis tools that are fully integrated into Code Composer Studio. The analysis tools acquire data without stopping the target processor, making them ideal for applications that must respond to real-time events.

The real-time analysis tools include a graphical view of an application's run-time behavior through an event graph that displays the sequence of thread execution. The CPU load graph illustrates the level of processor loading against program execution, enabling developers to determine areas where CPU resources are being stretched or where there is headroom to add more functions. To tune performance for individual parts of the application, real-time analysis enables the developer to acquire statistics on individual functions, including minimum, maximum, and total execution time.

In addition, real-time analysis enables developers to easily add simple 'printf-style' instrumentation at any point of an applications, via the LOG_printf and LOG_event system calls. The LOG_printf function eliminates the overhead of the standard C printf by doing formatting and string look-up on the desktop.

For more sophisticated data transfer up to the desktop host, DSPBIOS supports the Real-Time Data Exchange (RTDX protocol. RTTDX enables developers to set up data streams between the desktop host and target RTDX enables data on program behavior to be plotted in real-time in sophisticated host-based analysis tools such as spreadsheets or DSP system design tools. This greatly simplifies the debugging of complex algorithms by enabling the development team to observe in real-time how an algorithm responds to external conditions like changing acoustics or sudden noise.

Code Composer Studio



DSP/BIOS is seamlessly integrated with Code Composer Studio to provide a powerful set of real-time analysis and debugging tools.

The real-time analysis tools have been carefully designed to minimize the impact on real-time performance. Application and kernel instrumentation use the very low-overhead log functions to store information to a buffer in memory. This buffer is uploaded to the host using the RTDX protocol during idle time, ensuring that no cycles are stolen from the application itself. The user can configure which kernel events are logged to further minimize the instrumentation and upload overhead.

Real-time Analysis Operations	C5000 CPU Cycles	Time (usec) for MHz 5410	C6000 CPU Cycles	Time (usec) for 200 MHz 6201
Log Operations				
LOG_event	59	0.59	33	0.165
LOG_printf	59	0.59	36	0.18
Statistics Operations				
STS_set	19	0.19	14	0.07
STS_add	42	0.42	15	0.075
STS_delta	48	0.48	21	0.105

DSP/BIOS performance: The table illustrates the performance of selected DSP/BIOS real-time analysis functions on the TMS320C5000 and TMS320C6000 DSP architectures. All measurements are based on calls using the C language interface and DSP/BIOS for Code Composer Studio version 1.2.

TI Worldwide Technical Support

Internet

TI Semiconductor Product Information Center Home Page
www.ti.com/sc/support

TI Semiconductor KnowledgeBase Home Page
www.ti.com/sc/knowledgebase

Product Information Centers

Americas

Phone	+1(972) 644-5580
Fax	+1(214) 480-7800
Internet	www.ti.com/sc/ampic

Europe, Middle East, and Africa

Phone	
Belgium (English)	+32 (0) 27 45 55 32
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	+44 (0) 1604 66 33 34
Email	epic@ti.com
Internet	www.ti.com/sc/epic

Japan

Phone	International	+81-3-3344-5311
	Domestic	0120-81-0026
Fax	International	+81-3-3344-5317
	Domestic	0120-81-0036
Internet	International	www.ti.com/sc/jpic
	Domestic	www.tij.co.jp/pic

Asia

Phone		
International	+886-2-23786800	
Domestic	Local Access Code	TI Number
Australia	1-800-881-011	-800-800-1450
China	1-0810	-800-800-1450
Hong Kong	800-96-1111	-800-800-1450
India	000-117	-800-800-1450
Indonesia	001-801-10	-800-800-1450
Korea	080-551-2804	–
Malaysia	1-800-800-011	-800-800-1450
New Zealand	000-911	-800-800-1450
Philippines	105-11	-800-800-1450
Singapore	800-0111-111	-800-800-1450
Taiwan	0800-006800	–
Thailand	0019-991-1111	-800-800-1450
Fax	886-2-2378-6808	
Email	tiasia@ti.com	
Internet	www.ti.com/sc/apic	

B060101

TEXAS INSTRUMENTS