**Introduction**

Multicore digital signal processors (DSPs) are starting to find increasing acceptance in various market segments, including test and measurement, mission critical, industrial automation, medical and high-end imaging equipment, and high-performance computing. As processing demands of these applications continue to increase, Texas Instruments has responded with a new generation of scalable, high-performance TMS320C66x multicore DSPs. The C66x devices are based on TI's KeyStone architecture which is the platform for true multicore innovation by providing full processing capability to every core in a multicore device. TI's C66x high performance DSPs include pin-compatible and scalable devices available in one, two, four and eight-core configurations.

**Multicore software development kit**

TI has taken a very holistic view of the multicore DSP programming model and has developed an extensible suite of tools and software to enable rapid development on TI's multicore DSP platform. This paper introduces TI's Multicore Software Development Kit (MCSDK) by outlining the various software packages available, along with utilities and tool chains that can aid programmers in development for high-level operating systems such as Linux, and the real time operating system SYS/BIOS.

**MCSDK benefits**

TI's MCSDK aims to provide a software development environment that enables rapid development and faster time to market with TI's high-performance multicore DSP platforms. The MCSDK accomplishes this goal by:

- Providing well-integrated and tested common software layers to the customer so that customers do not have to develop common layers from scratch. For example, TI's MCSDK integrates and tests various drivers to configure and control various on-chip peripherals and accelerators. Customer can use the driver interfaces to enhance on-chip input and output (I/O) mechanisms and acceleration. Since the software is optimized for the device, customers using the MCSDK benefit through optimum performance entitlement.
- Integrating support for the SYS/BIOS real time operating system and Linux high level operating system.
- Providing well-defined application programming interface for ease of programming as well as future portability on TI's scalable multicore DSP platforms. For example, the APIs provided for intracore communication can be scaled to support two, four or eight core versions of TI's multicore DSPs without any code changes. The same APIs can also be used to realize inter-device communication using industry standard IOs like Serial RapidIO®.
- Documenting examples to enable programmers to develop their application. These examples will provide help in running RTOS on multiple cores as well as running RTOS and HLOS simultaneously on multiple cores. The examples will showcase application scenarios and enable customers to develop new applications, and also provide a migration path from single core to multicore system or vice versa.
- Integrating with TI's tools like Code Composer Studio™ as well as TI's third party tools ecosystem.
The following picture (Figure 1) aims to provide overview of the MCSDK and various components described above.

▲ Fig. 1 – TI’s MCSDK components

TI’s MCSDK is actually comprised of two software ecosystems. The first ecosystem is based on Linux, and the second one is based on SYS/BIOS. Both include all of the foundational software to get a customer started with development quickly and easily. Each ecosystem includes mechanisms for programming multicore devices (e.g., interprocessor communication) and can be used independently or in combination on different cores of the same device. The following sections will provide an overview of each MCSDK component.

**Linux MCSDK**

The Linux Multicore Software Development Kit (MCSDK) provides the necessary foundation for support of a Linux ecosystem on TI’s C66x generation of high performance, multicore DSPs. The kit includes a ready-to-use Linux kernel, drivers, sample applications, and verified tools for customer product development. The Linux MCSDK is available as an open source distribution at [www.linux-c6x.org](http://www.linux-c6x.org) and includes pre-built binaries ready to run on reference platforms to demonstrate the capability of TI high performance multicore DSPs.

In general, Linux for C66x multicore DSPs is intended to be a collaborative, community-driven effort among C66x multicore customers and independent developers and vendors to engage and strengthen the overall development ecosystem. Multiple participants are currently involved in development and upstream alignment for both kernel and tool chain. This includes CodeSourcery (now part of Mentor Embedded™) for GCC tool chain and Linux developers well known to the kernel community.
Support for the Linux ecosystem is expected to be a key enabler for a richer set of high-performance applications enabling customers to easily bring up their platforms based on TI C66x multicore DSPs. Overall, the infrastructure allows customers to reduce cost of development and focus on increasing their value proposition by adding market-driven applications.

The Linux MCSDK uses a uClinux-type kernel because its smaller footprint is ideally suited for multicore DSP development. In addition, the Linux MCSDK includes device drivers to enable access to all peripherals within the DSP, which will also vary based on the actual DSP. Figure 2 highlights peripherals and drivers that will be supported for KeyStone devices.

![Fig. 2 – Software ecosystem components for Linux MCSDK](image-url)

**Ecosystem components**

One of the key programming requirements for multicore architectures is to have efficient communication across the various cores. TI’s Linux MCSDK includes support for communication across the cores running SYS/BIOS as well as other Linux cores, providing flexibility to address a range of potential product needs.

**Inter Process Communication**

One of the key programming requirements for multicore architectures is to have efficient communication across the various cores. TI’s Linux MCSDK includes support for communication across the cores running SYS/BIOS as well as other Linux cores, providing flexibility to address a range of potential product needs.

**IPC between cores**

TI’s Linux MCSDK includes an Inter Process Communication (IPC) driver through SYS/Link module to communicate between cores running Linux High Level Operating System (HLOS) and multiple cores running the SYS/BIOS Real Time Operating System (RTOS). An expected use case is to have control code running on the Linux core while signal processing applications are executed on the BIOS cores. The software
architecture will allow the host core running Linux to distribute signal processing across multiple cores while also continuing to do other tasks without blocking. SYS/Link IPC module provides MessageQ support for sending and receiving variable length messages across multiple cores.

As highlighted in Figure 3, Linux MCSDK will also enable multiple instances of Linux running across cores within a single DSP using virtIO which is a standardized Linux framework for IO virtualization.

The Linux MCSDK provides the user a number of options for compiler tools as well as debugging. On the compiler side, GCC and binutils commonly used within the open source community are available for C66x multicore DSPs through Mentor Embedded. In the case of applications such as signal processing code, a developer would be able to derive higher performance using compilers from TI. Additionally, the GCC and TI compilers are interoperable so that a developer could selectively use the TI compiler or GCC for different parts of the application code and still retain the rest of the system built with GCC. This provides the user with the flexibility to make trade-offs associated with code performance.

On debugging, applications in user mode can use GDB available through Mentor Embedded. For kernel debugging or programs running without any operating system, JTAG-based debugging will be supported using Debug Sprite from Mentor Embedded and CCS based debugger from TI.

A final component of the Linux MCSDK is the inclusion of “out-of-the-box” (OOB) demonstration software and example applications. The OOB demonstration application includes a web-based interface that gives users control panel access to an EVM through a PC connected over Ethernet to provide a variety of functions after boot. In the initial releases, the control panel will provide an easy way for the user to update the boot loader and kernel. Later releases will include support for other basic functions that illustrate and demonstrate capabilities associated with multicore, such as IPC communication between core running BIOS and/or Linux, downloading and booting multicore applications, etc.
The BIOS Multicore Software Development Kit (MCSDK) provides core foundational building blocks in an integrated fashion to facilitate application software development on TI’s high performance multicore DSPs using the SYS/BIOS real-time operating system. The BIOS MCSDK bundles all the key embedded software in a single downloadable package and is available for free on TI’s website. Software included in the BIOS MCSDK is in source form along with pre-built libraries, and it is distributed under a BSD license. In addition to the foundational software elements, the BIOS MCSDK provides demonstration applications that utilize the components and show how a customer can create an application using the BIOS MCSDK.

One of the driving considerations in designing the organization of the BIOS MCSDK was to simplify the development flow across platforms as well as consider customer migration across TI devices. TI understands that customers may have multiple products with common software that span different devices, and the migration strategy to leverage their software investment on existing and future TI devices is an important consideration. An example of this flow is to begin by running the included TI demonstration software on the TI evaluation platform, migrate the demonstration to the customer platform, then bring up the customer application on the customer platform. One additional step is to port this application to the next generation TI architecture (Figure 4).

BIOS MCSDK uses TI Code Composer Studio™ (CCS) as the Integrated Development Environment (IDE). For software updates, BIOS MCSDK utilizes the Eclipse update feature to automate discovery and installation of new software via CCS.

The software components (Figure 5) in BIOS MCSDK can be categorized into the following:

- Device-specific software drivers
- Platform-specific software
- Core target software
- Demonstrations and tools
Device-Specific Software Drivers – This set of software includes chip support library, low level drivers, platform library and transports. The software in this set is focused on simplifying access to the device hardware, including accelerators, and is offered as a minimum layer API for application development.

Platform-Specific Software – This software provides example implementations of platform-specific functionality for use with TI reference platforms, but with the intent to be used as an example for customer platform development activities.

The Platform Library abstracts the platform using a common API to simplify migration across devices. It provides software utilities to control hardware parts (e.g., EEPROM, FLASH, UART) and performs platform/device initialization.

The Platform Library API is leveraged by common tools spanning different platforms. These include EEPROM writer, NAND/NOR writer, and Power-On Self Test. Thus, to add these tools for a new device or platform, the work is isolated to adding a Platform Library unique to that hardware. As with other components, the Platform Library comes with a unit test application that can be used when porting the module to a new platform.

Core target content – This set of software provides the next level of functionality and includes the real-time embedded operating system, inter-processor communication for communication across cores and devices, basic networking stack and protocols, optimized application-specific algorithm libraries, and instrumentation tools.

One key multicore enabler includes inter-processor communication to enable efficient communication across cores and devices using shared memory (inter-core communication) and peripherals such as sRIO, PCIe for inter-device communication. Interprocessor communication mechanism and API provide support for explicit threading multicore programming models. A common IPC interface also allows for easier movement of application processing nodes across core and or devices as required for a multicore application design and optimization.

A second key enabler is instrumentation tools which are critical for providing a user visibility into application execution, and the ability to analyze and optimize for performance. BIOS MCSDK includes a System Analyzer tool that...
defines a set of APIs that allow instrumentation code to be inserted into the software in a portable manner that can be reused across a wide range of TI platforms. An example of the graphical output of one application is below (Figure 6).

**Fig. 6 – BIOS MCSDK System Analyzer tool**

**Demonstrations and tools** – This set of software spans multiple devices and provides examples, demonstrations and tools that build on top of the above software. The BIOS MCSDK includes demonstration applications which are intended to serve as an example and show how to use foundational software to build multicore applications. Demonstrations available for download in the current release include an “out of box” application that showcases some key components of the BIOS MCSDK, and an image processing application that displays multicore signal processing.

Tools include a common boot loader, FLASH and EEPROM writers, power-on self test for an evaluation board, multicore/multi-image boot tool, and brief examples to boot from different modes (e.g., NAND/NOR, EMAC, SRI0).

**Demonstrations**

**Out of box demo**

The “out of box” demo for BIOS MCSDK is the high-performance DSP Utility Application (HUA) which is provided as a CCS project. Through illustrative code and web pages, it demonstrates how you can interface your own DSP application to the various BIOS MCSDK software elements including SYS/BIOS, Network Development Kit (NDK), the Chip Support Library (CSL), and Platform Library.

Upon execution, the HUA implements a web server that allows a user to connect via Ethernet to the platform with a PC. The HUA allows a user to perform a variety of functions, such as read/write flash, diagnostics, or provide statistics and information. Through illustrative code and web pages, it demonstrates how you can interface your own DSP application to the various BIOS MCSDK software elements including SYS/BIOS, Network Development Kit (NDK), the Chip Support Library (CSL), and Platform Library (Figure 7).

**Fig. 7 – High-performance DSP Utility Application (HUA)**
The Image Processing Demonstration (Figure 8) illustrates the integration of key components in the BIOS MCSDK multicore signal processing. It utilizes inter-processor communication, optimized image library, Network Development Kit, and system analyzer. The latter is used to collect and analyze benchmark information.

This demonstration can be configured to run on any number of cores supported by the device. It is partitioned into one master task on the first core and a number of slave tasks distributed across all cores. The master task is responsible for partitioning the input data, distributing the work to slave tasks, collecting results from slave tasks, and sending the output data. The algorithm that is currently supported is edge detection, but is expandable to more algorithms.

Summary

TI’s MCSDK provides customers with a robust and integrated software development environment for Linux and SYS/BIOS, with the goal of enabling rapid development and faster time to market with high-performance DSPs based on the KeyStone architecture.

For more information about the MCSDK, visit www.ti.com/mcsdk, or visit www.ti.com/c66multicore for more about the KeyStone architecture and C66x generation of high performance multicore DSPs.
# IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal and regulatory requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or “enhanced plastic.” Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer’s risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Communications and Telecom</td>
</tr>
<tr>
<td>Amplifiers</td>
<td>Computers and Peripherals</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>DLP® Products</td>
<td>Energy and Lighting</td>
</tr>
<tr>
<td>DSP</td>
<td>Industrial</td>
</tr>
<tr>
<td>Clocks and Timers</td>
<td>Medical</td>
</tr>
<tr>
<td>Interface</td>
<td>Security</td>
</tr>
<tr>
<td>Logic</td>
<td>Space, Avionics and Defense</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Transportation and Automotive</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Video and Imaging</td>
</tr>
<tr>
<td>RFID</td>
<td>Wireless</td>
</tr>
<tr>
<td>RF/IF and ZigBee® Solutions</td>
<td></td>
</tr>
</tbody>
</table>

TI E2E Community Home Page: e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated