WHITE PAPER

Kyle Harper,
Marketing director, multicore processors
Texas Instruments

TEXAS INSTRUMENTS

Enhancing the KeyStone II architecture with multicore RISC processing

Abstract

The KeyStone II multicore architecture, TI's second-generation KeyStone architecture, adds multicore RISC processing alongside its industry-leading multicore DSP capabilities. The ARM® Cortex™-A15 MPCore™ processor combines leading processing capabilities with a very low power/performance ratio, multicore hardware-based cache coherency and broad industry software support. The integration of Cortex-A15 processors into the KeyStone architecture brings a new level of performance to multicore ARM-based SoCs.

Introduction

Unlike most multicore ARM SoCs, which target consumer devices like mobile phones and tablets, the KeyStone II architecture addresses the demanding performance requirements of embedded infrastructure applications. Some of the features that differentiate a KeyStone-based embedded infrastructure-class SoC from a consumer-class SoC are memory performance, error correction, and robust I/O and infrastructure grade reliability of 100,000 hours of constant power-on MTBF performance. The first SoC platforms based on the KeyStone II architecture, the 66AK2Hx, 66AK2Ex and AM5K2Ex devices, provide up to 9.6 GHz of DSP and 5.6 GHz of ARM computational processing power.

This white paper summarizes how multicore Cortex-A15 processors in the KeyStone II architecture enable optimum performance for embedded infrastructure applications. Both the hardware and software aspects of Cortex-A15 processors and their memory path are discussed. In addition, the architectural advancements of the KeyStone II architecture are

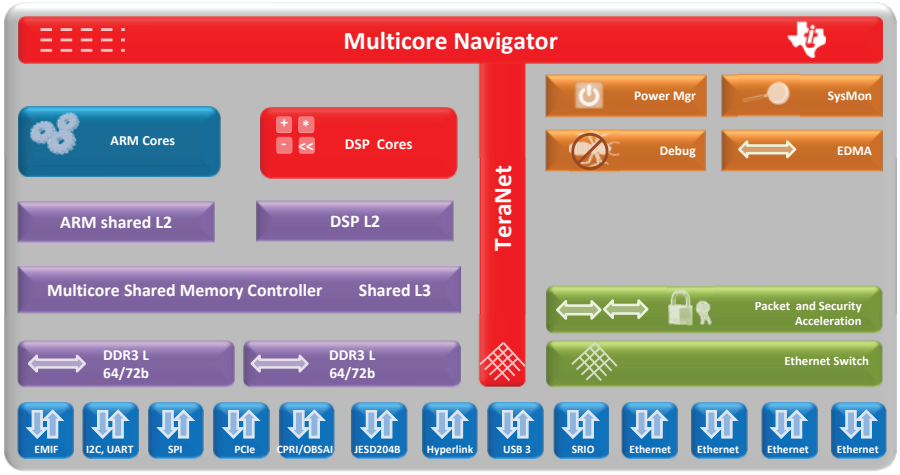


Figure 1: KeyStone II architecture

described. Specifically, various features of Cortex-A15 processors relative to the memory environment within which they operate are correlated to aspects of the KeyStone II architecture. This explains how KeyStone II SoCs are able to deliver very high performance for various software use cases that arise in the most demanding of industry and infrastructure applications, such as cloud infrastructure, network/enterprise, medical imaging, mission critical radar, high-performance computing, test and measurement, high-end inspection, video analytics and industrial automation.

KeyStone II architecture

Full multicore entitlement

In the KeyStone II architecture, as shown in Figure 1 on the previous page, applications can take full advantage of multiple processing cores. The critical elements of the KeyStone architecture are TeraNet, HyperLink, Multicore Navigator, CorePacs, Advanced Packet Processing, Multicore Shared Memory Controller and infrastructure-class I/O, which will be expanded on below.

TeraNet

TeraNet maximizes the throughput of the on-chip data flow within the KeyStone multicore architecture. TeraNet is a multilevel interconnection of high-speed, non-blocking channels that delivers over two terabits per second of concurrent data throughput. With TeraNet, data flows freely and effectively, providing full multicore entitlement where every processing element can operate near full capacity at all times.

HyperLink

HyperLink extends TeraNet off-chip to other devices. Each HyperLink port is a 50-Gbaud SERDES-based interface with minimal protocol overhead, resulting in low latency and high throughput. Very efficient connections to other TI SoCs and to third-party devices with HyperLink connectivity are enabled. Multiple SoCs interconnected with HyperLink appear as one large virtual device, greatly simplifying software development and improving system performance. To the software programmer, the resources of all HyperLink-connected SoCs and third-party devices are addressed as local address-mapped resources.

Multicore Navigator

A breakthrough feature of the KeyStone architecture is the innovative Multicore Navigator, which functions in conjunction with TeraNet to bring single-core simplicity to multicore devices. Multicore Navigator provides 16,000 hardware-based multiple-in and multiple-out queues (doubling that of KeyStone I). With support for 40 million push-and-pop pairs per second and working in collaboration with multiple packet-based direct-memory-access engines, a high-throughput on-chip network for intra-SoC communication is created. Multicore Navigator greatly improves and virtualizes resource management, inter-processor (ARM-to-ARM, DSP-to-DSP or ARM-to-DSP) communications and inter-process communication. Multicore Navigator, in

conjunction with support for industry-standard explicit multicore programming models such as the OpenMP® API specification for parallel programming, provides a hardware-assisted software API abstraction layer, which allows fast and efficient development of software that scales optimally to however many ARM or DSP cores, or hardware accelerators are available at runtime.

Software running on Cortex-A15 processors and the Linux™ open-source operating system will see significant performance enhancements because Linux-based inter-process calls will be accelerated through Multicore Navigator. In fact, Multicore Navigator will assist software APIs that operate under open-source operating systems such as Linux.

For developers of bare-metal and low-overhead Linux software, Multicore Navigator's low latencies and zero interrupts, as well as its transparent operations will enable new and more effective task dispatcher programming models. With Multicore Navigator's "fire-and-forget" software tasking, developers save significant time and effort by defining repetitive tasks only once and thereafter accessing and running these tasks automatically without additional coding.

CorePacs

CorePacs are the main processing elements in the KeyStone architecture. The KeyStone II architecture introduces Cortex-A15 processors alongside TI's TMS320C66x DSP cores. KeyStone II-based SoCs can be configured with multicore ARM CorePacs, DSP CorePacs and combinations of both.

In addition to DSP and ARM cores, each CorePac contains Layer 1/Layer 2 cache subsystems. There are three levels of memory in the KeyStone architecture. For DSP cores, each C66x CorePac has its own Level 1 program (L1P) and Level 1 data (L1D) memory, as well as its own Level 2 cache/SRAM that can be independently configured as memory-mapped SRAM, cache or both. Each Cortex-A15 processor in an ARM CorePac has its own Level 1 instruction and data cache. A 4-MB Level 2 cache/SRAM is shared among all Cortex-A15 processors within an ARM CorePac. All Level 1 and Level 2 memories for ARM CorePacs are Error Correction Code (ECC) protected. Level 1 and Level 2 memories on the DSP CorePacs are also protected from soft errors.

Level 3 shared memory is implemented via the KeyStone architecture's high-performance shared memory subsystem called the Multicore Shared Memory Controller (MSMC). The MSMC allows the DSP and ARM CorePacs to dynamically share on-chip Level 3 SRAM and an off-chip DDR memory port. Details on Level 2/3 memory and the MSMC are described later in this paper in the "Path to Memory" section.

Advanced Packet Processing

The KeyStone architecture features several innovative co-processing accelerators that offload processing tasks from the SoC's DSP and ARM cores, enabling and sustaining high application processing rates. The SoCs contain a network coprocessor block that consists of a Packet Accelerator and Security Accelerator. These accelerators work in tandem to support inline IPSec processing in both ingress and egresss directions. The Packet Accelerator speeds the data flow throughout the SoC by transferring data to peripheral interfaces

such as the Ethernet or Serial RapidIO® ports, and offloads substantial Layer 2–4 packet processing from the DSP and ARM cores. The Security Accelerator provides security processing for a number of popular encryption modes and algorithms, including IPSec, SCTP, SRTP, 3GPP, SSL/TLS.

Cortex-A15 processors

At the heart of the KeyStone II architecture's ARM CorePac are clusters of Cortex-A15 processors containing one, two or four ARM cores. Running at 1.4 GHz, each Cortex-A15 processor offers 4900 Dhrystone MIPS, as well as 11.2 GMACS of floating-point performance. The Cortex-A15 processor is a 15-stage integer pipelined, 17–25 stage floating-point pipelined, out-of-order execution, three instructions-per-clock superscalar implementation of the ARM v7, an architecture with optimized branch prediction and branch cache. It has a tightly coupled VFPv4 floating-point unit for double-/single-precision floating point and a NEON™ unit containing SIMD extensions for codecs, 2D/3D graphics and other advanced processing. It also contains full hardware virtualization for running multiple operating systems simultaneously, as well as the ARM TrustZone extensions and hypervisor support. Each Cortex-A15 processor contains its own private Level 1 cache memory comprised of 32-KB instruction and 32-KB data cache. Both caches are two-way set associative running at the full RISC CPU clock rate with ECC protection (single-bit correction, two-bit error bit detection). Moreover, both big-endian and little-endian memory storage formats are supported, as well as secure boot from either endian format. Regardless of whether the KeyStone SoC contains one, two or four Cortex-A15 processors in its ARM CorePac, all Cortex-A15 processors share 4 MB of 16-way Level 2 cache memory operating at the top clock rate of CPU clock/1.

Hardware-Based Cache Coherency

The Cortex-A15 processor provides hardware-based cache coherency, maintaining ARM processor-to-processor coherency and I/O coherency. This eliminates the need for developers to design software that will maintain coherent copies of the same main memory data values cached in any Level 1 or Level 2 caches on

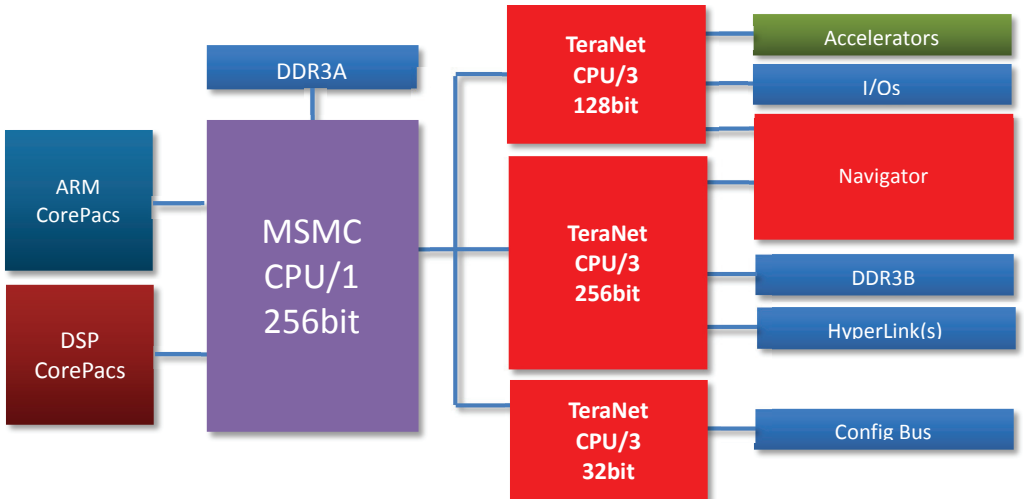


Figure 2: KeyStone II architecture MSMC and memory path

any of the Cortex-A15 processors. This greatly simplifies the architecture and development of cache management and I/O system software, and greatly increases the performance of system and application software. Not only is coherency of data maintained across all Cortex-A15 processors, but it is also maintained across all I/O bus masters.

Multicore Shared Memory Controller – Path to memory

The KeyStone architecture's multicore shared memory controller (MSMC), as shown in Figure 2 on the previous page, is critical for demanding applications where memory performance is often a limiting factor on system throughput. The upgraded MSMC in KeyStone II SoCs allows all on-chip memories to operate at the same clock rate as the cores, resulting in very low latency, very high bandwidth interconnects among all cores, all on-chip memories and all off-chip DDR memories. The MSMC can also control from 2 MB to 6 MB of Level 3 SRAM. This Level 3 SRAM is shared by all ARM and DSP CorePacs. The MSMC is directly connected to a 72-bit DDR3/3L-1600 external memory port (64 bit plus ECC) which is treated as part of Level 3 memory. The MSMC DDR interface bypasses TeraNet so that memory accesses do not interfere with other on-chip communications. Likewise, DSP and ARM memory accesses do not impede other data or I/O traffic on TeraNet. A second external 72-bit DDR3/3L-1600 memory port specifically dedicated to I/O traffic is available on some KeyStone II SoCs. This second port is connected directly to TeraNet and offers high-bandwidth data access to all bus masters in the system, not just the DSP and ARM cores. All memories, including on-chip SRAM and off-chip DDR, are ECC protected.

Cache coherency is maintained in hardware for all memories that are accessed by Cortex-A15 processors via the MSMC, thus eliminating the need for software to maintain ARM processor-to-processor and processor-to-I/O memory traffic coherency. The result is an on-chip memory datapath that provides unblocked full entitlement of the processing capabilities for all Cortex-A15 processors and C66x DSP cores to the application. All on-chip memory operates at the SoC's top clock rate of CPU clock/1. Applications running on the Cortex-A15 processors benefit significantly from the resulting low latency, high-bandwidth external memory access rates. In a typical use case, most of the operating system code and data reside in external memory. With the MSMC, each access to the external memory requires far fewer CPU cycles, significantly increasing application performance.

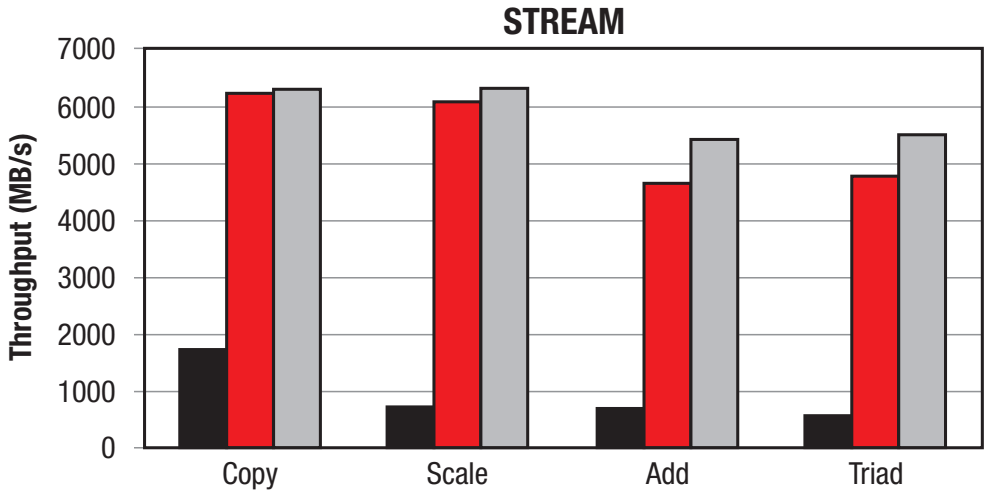
MSMC also provides a non-blocking path to the external DDR memory for ARM and DSP CorePacs. In all cases, external memory access from applications running on Cortex-A15 processors will not be blocked by ongoing external memory accesses from DSP cores except when the same memory bank is accessed simultaneously.

KeyStone II performance

A very good system-level benchmark reflecting the Cortex-A15 processor performance within the Keystone II SoC architecture is STREAM. Consisting of four vector kernels, the STREAM benchmark is the de facto industry-standard benchmark for the measurement of sustainable memory bandwidth. It is a combination of

floating-point and memory-bandwidth benchmarks, containing four individual benchmarks: COPY (floating-point loads and stores), SCALE (floating-point multiply), SUM (floating-point add) and TRIAD (floating-point multiply and add). Benchmarks were performed to compare the STREAM performance of ARM® Cortex-A8 processors in the KeyStone I architecture and Cortex-A15 processors in the KeyStone II architecture. For the Cortex-A8, the TCI6614 EVM system configured with a 1-GHz Cortex-A8 processor and 64-bit DDR3-1333 external memory was used. For the Cortex-A15, the KeyStone II RTL simulator configured with a 1.4-GHz Cortex-A15 processor and 64-bit DDR3-1600 external memory was used.

Table 1 below compares the results of running STREAM on Cortex-A8 and Cortex-A15 single cores, and on Cortex-A15 dual cores. Comparing single-core performance, the Cortex-A15 processor's 6228 MB/s COPY bandwidth is 3.57 times better than that of the Cortex-A8 processor's, indicating significant improvement in memory reads and writes by the Cortex-A15 processor and the KeyStone II MSMC memory path. Similarly, the Cortex-A15 processor's SCALE, SUM and TRIAD performance is better than the Cortex-A8 processor's. Its 6073 MB/s SCALE bandwidth is 8.34 times better, the 4633 MB/s SUM bandwidth is 6.52 times better and the 4753 MB/s TRIAD bandwidth is 8.07 times better. These results represent substantial improvements resulting from the Cortex-A15 processor's VFPv4 floating-point processing and from KeyStone II's MSMC-based path to memory.



	Copy	Scale	Add	Triad
A8 - 1 core	1744	728	711	589
A15 - 1 core	6228	6073	4633	4753
A15 - 2 core	6280	6291	5390	5488

Cortex-A8: 1 GHz, 64-bit DDR2-1333 memory, Codesourcery GCC v4.6.3 compiler;
Cortex-A15: 1.4 GHz, 64-bit DDR3-1600 memory, Linaro GCC v4.6.3 compiler

Table 1. Comparison of results of running STREAM on Cortex-A8 and Cortex-A15 single cores and Cortex-A15 dual cores

Furthermore, STREAM benchmarking, when the results are expressed as a percentage of theoretical maximum DDR memory bandwidth or "wire speed," also shows how effectively KeyStone II's path to DDR memory is utilized. Achieving close to 50 percent of wire speed on a DDR port is considered state-of-the-art

for practical bandwidth utilization. KeyStone II's maximum wire speed is 12800 MB/second, whereas KeyStone I's is 10664 MB/second. The STREAM results table shows KeyStone II's single Cortex-A15 processor with MSMC delivers 36–48 percent utilization of the DDR port's wire-speed bandwidth, a dramatic improvement over the 5–16 percent utilization range of KeyStone I single Cortex-A8 processor with MSMC. With the dual-core Cortex-A15 processor and MSMC, the DDR utilization range increases further to 42–50 percent.

Enabling efficient communication

The KeyStone II architecture enables very effective communication between software running on the Cortex-A15 processor, C66x DSP and the other elements in the SoC, such as I/O, accelerators and co-processors.

The key architectural element responsible for this is the Multicore Navigator. Multicore Navigator provides:

1. Dynamic load optimization between cores
2. Effective routing between peripherals (I/O)
3. Efficient intra-device communication

In order to achieve these goals, Multicore Navigator uses packet-based communication to reduce power consumption and increase software efficiency in multicore systems.

Figure 3 below provides a high-level view of Multicore Navigator. The two key elements to Multicore Navigator are:

1. Queue Manager
2. Packet Direct Memory Access (Packet DMA)

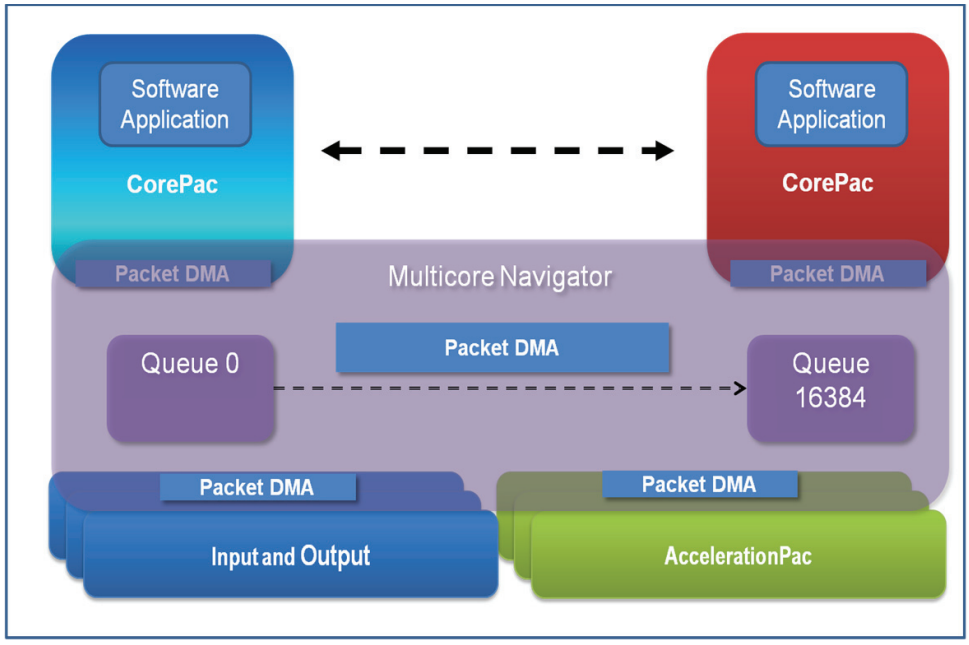


Figure 3: Multicore Navigator

The Queue Manager acts as a hardware-based communication API between various SoC elements (including ARM and DSP cores), while the Packet DMA ensures guaranteed delivery of packets to the destination.

Figure 4 illustrates Multicore Navigator in action. When a software application wants to communicate with other SoC elements, the application builds a packet containing the data to be sent to the destination. The packet is then queued to the Multicore Navigator using the Queue Manager as the API. Multicore Navigator is intelligent so it can understand where the destination of the packet is. It then delivers the packet to the destination using packet DMA. Destination elements act on the packet and perform further actions.

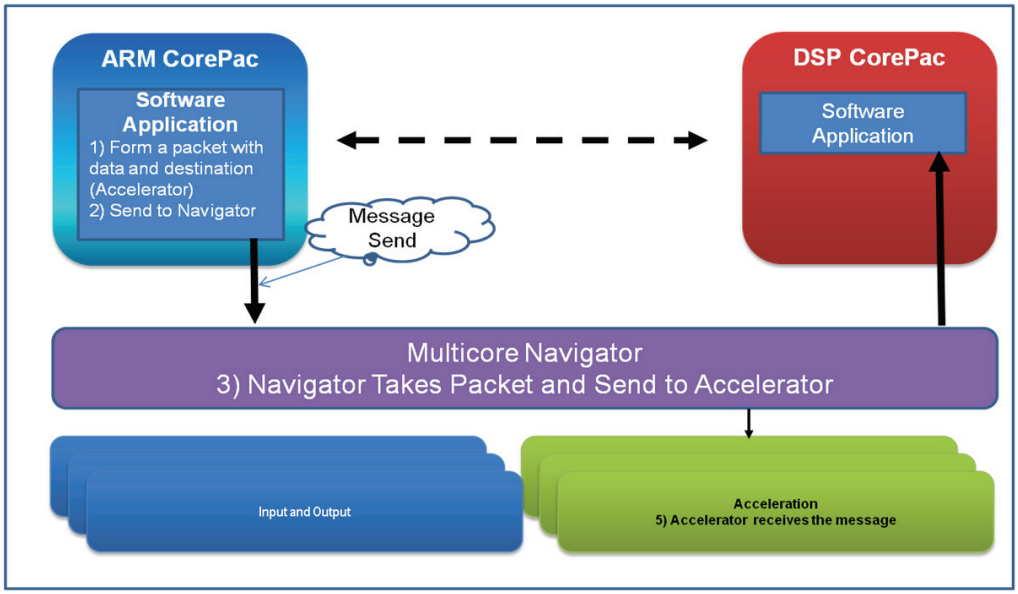


Figure 4: Efficient, low-overhead utilization of SoC hardware resources

This innovative approach provides a simplified abstracted view for software development and results in very efficient, low-overhead utilization of SoC hardware resources.

From the software perspective, it is beneficial to consider how software would behave in the absence of Multicore Navigator. Figure 5, on the following page, illustrates this. In traditional multicore systems, software applications running on a core, such as the Cortex-A15 processor, would communicate with other SoC elements by typically using a shared resource to populate the message. Then, following the sending of the message, precious processor cycles would be consumed in order to acknowledge that the message had indeed been received by the intended recipient and is being acted upon. While effective, this approach places a significant amount of processing overhead, a "tax" if you will, on CPU resources to acknowledge communications between various SoC elements.

The Multicore Navigator approach is shown in Figure 6, on the following page. With Multicore Navigator, the processing "tax" placed on the CPU to manage communications among SoC elements is reduced to essentially nothing.

Software running on a Cortex-A15 processor sends messages to Multicore Navigator which takes responsibility for delivering these messages to their destinations. Similarly, when other SoC elements want to send a

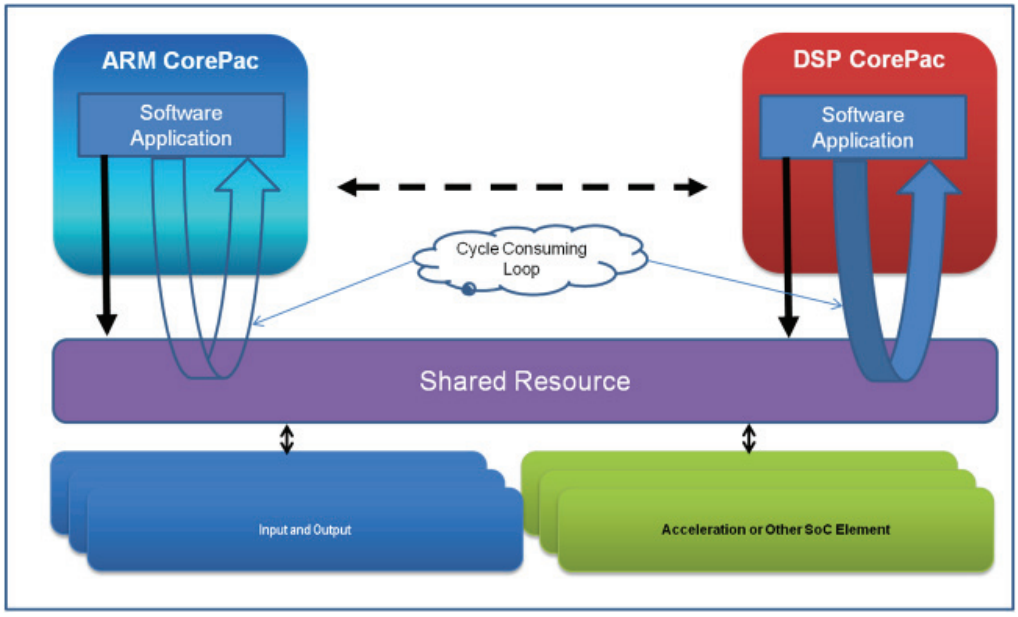


Figure 5: Software applications in traditional multicore systems

message to the software application running on the Cortex-A15 processor, for example, Multicore Navigator performs the communications process for them.

With the help of KeyStone II's Multicore Navigator coordinating the communications among ARM cores or between ARM cores and other SoC elements, a system designer can achieve optimum efficiency and produce software that scales to any number of processing cores and SoC elements without redesigning the software.

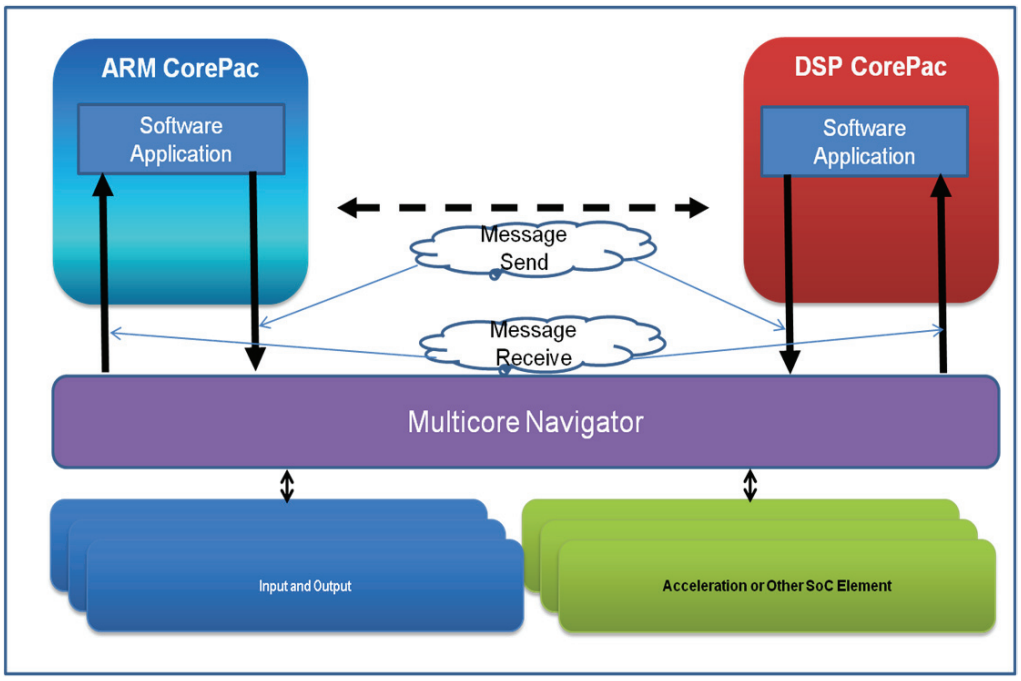


Figure 6: Software applications with Multicore Navigator

Multiple software use cases

With the integration of Cortex-A15 processors into the KeyStone architecture, many different software use cases can be supported. These include:

1. **Symmetric Multi Processing (SMP)**

   This use case refers to an operating system's ability to use a cluster of Cortex-A15 processors transparently within a software application. An application running on an SMP-capable OS like SMP Linux can immediately take advantage of multiple cores in the SoC.

   SMP Linux has been supported on the ARM architecture since the ARM 11 MPCore generation was introduced in 2004. Since then, SMP Linux on ARM has gained significant adoption and has evolved into a mature and stable implementation. SMP Linux support on the KeyStone platform builds upon these developments.

   Beyond traditional multiprocessing capabilities, the KeyStone hardware architecture is capable of a higher level of SMP performance by supporting a wide variety of features such as I/O coherency, fast multicore interaction with DMAs and with other CPUs without implementing spin-locks and semaphores (Multicore Navigator), hardware broadcast for TLB/cache operations (DVM signaling), an interrupt architecture tuned for SMP operation and a memory architecture designed for the high-throughput demands of enterprise and infrastructure needs. The primary focus of KeyStone Linux is to enable features such as these and optimize their operations in order to deliver a high-performance solution for SMP applications.

2. **Asymmetric Multi Processing (AMP)**

   Each individual core runs an independent OS in AMP mode of operation. AMP is supported on both KeyStone I and KeyStone II devices.

   Several generations of TI SoCs have supported heterogeneous AMP on ARM/DSP architectures. The necessary software infrastructures were built into these architectures to boot, manage and communicate with other on-chip and off-chip cores. TI has leveraged this history and expertise to provide similar support for AMP on KeyStone II multicore ARM SoCs.

   The foundational real-time operating system (RTOS) for ARM cores in AMP applications consists of TI's SYS/BIOS™ RTOS supported by a broad suite of low-level drivers and chip support libraries suitable for use in an RTOS environment. The SYS/BIOS RTOS is widely used on TI DSPs. The addition of Cortex-A15 processor support for SYS/BIOS RTOS maintains a consistent operating environment across ARM and DSP AMP cores.

   Multi-threaded applications such as packet processing can benefit significantly from AMP operations.

3. **Simultaneous SMP and AMP**

   This use case is a combination of SMP and AMP. The Cortex-A15 processors in the KeyStone II architecture are optimal for this use case.

4. Hypervisor

Virtualization based on the Kernel Virtual Machine (KVM) is supported by the KeyStone II architecture.

The Cortex-A15 processor on the KeyStone platform supports an additional privilege level for hypervisor execution. This enables more precise control of guest machine accesses to privileged registers.

Memory virtualization is supported via an added level of translation from Virtual Address to Intermediate Physical Address and again to Physical Address. VMID-tagged TLB entries enable efficient TLB management by avoiding TLB flushes during context switches. The interrupt controller supports virtualization extensions, eliminating the need to make I/O exits for PIC MMR emulation. These elements of the virtualization solution are based on standard ARM technology, allowing for the broadest software support and reuse.

KeyStone II I/O virtualization is based on the Multicore Navigator with its multicore shared hardware queues and a queue- and descriptor-aware scatter/gather DMA. I/O DMA accesses to physical memory go through an address translation and memory protection (memory protection and address extension, or MPAX) table. The Cortex-A15 processor's software accesses to I/O registers and memory go through memory protection units (MPUs). I/O virtualization takes advantage of specially developed approaches to achieve the bandwidth and performance levels required in demanding network infrastructure applications.

5. Open Source Multicore Programming Paradigms – OpenMP® and OpenCL™

TI led the industry by first enabling the OpenMP programming paradigm for multicore DSPs. The OpenMP execution model is supported by TI on Cortex-A15 processor and C66x DSP cores. The OpenCL runtime is supported on TI's C66x multicore DSPs.

With the introduction of KeyStone II, heterogeneous SoC architectures are possible with multiple ARM and DSP cores. In order to make programming easier for such heterogeneous systems, TI is co-chairing the OpenMP industry group and is proposing innovative models so that existing OpenMP code can be run on heterogeneous systems with very few changes. At the conceptual level, the intent is to enable an application running on a multicore Cortex-A15 processor to use C66x DSP cores as accelerators. An ARM core can be running an SMP or AMP Linux OS environment. With such a model, the programmer has the ability to indicate the region of the application to be accelerated by the C66x DSP core, while other code regions can be run on the general-purpose Cortex-A15 processors. A support toolset allows the programmer to cross-compile the code for this heterogeneous ARM/DSP system and run it.

The OpenCL runtime is also supported for two different use cases. In one use case, KeyStone II-based SoCs can be deployed as OpenCL accelerators in conjunction with an x86 processor. In this case, the OpenCL runtime environment will execute on the x86 processor and communicate with KeyStone II-based SoC(s) that enable online or offline compilation for ARM or DSP kernels. In the second use case, the OpenCL runtime environment executes on the KeyStone SoC's Cortex-A15 processor (executing in SMP or AMP Linux modes) and this runtime environment sends any DSP-designated OpenCL kernels to the C66x DSP cores for execution.

Both OpenMP and OpenCL are industry-standard explicit multicore programming paradigms. By supporting both x86 and Cortex-A15 cross compiling, the programmer is given options to chose from and ease-of-use is enhanced. Users with an OpenMP or OpenCL code base can readily take advantage of the C66x DSP technology without spending time learning how to program the C66x DSP architecture.

Similarly, programmers do not have to learn how to program the KeyStone architecture's Multicore Navigator either. TI's OpenMP and OpenCL implementations take full advantage of the Multicore Navigator. It accelerates and increases the performance of the OpenMP and OpenCL runtime implementations, making each much faster than those running on multicore SoCs from other vendors. Users of OpenMP and OpenCL are able to apply all of the performance enhancements of the KeyStone architecture without learning how to explicitly program and use the hardware elements in KeyStone.

Secure boot

Regardless of endianness of the processor, the secure boot mechanism implemented in KeyStone II protects the Cortex-A15 processors and C66x DSP cores against running non-authorized code. This mechanism also establishes SoC-based "root of trust" for future security needs.

KeyStone's secure boot mechanism includes the following capabilities which use SoC-based root of trust.

- IP protection
  - Prevents copying of code in external Flash
  - External contents of Flash (code/data) are encrypted
- Takeover protection
  - Device only executes trusted code
  - Non-volatile one-time-programmable memory within the SoC can be configured so only trusted software will boot the device
- Open manufacturing protection
  - Working in conjunction with users, TI can enable customer-specific encryption key programming within the device.

In order to provide such secure functionality, the KeyStone II architecture will support the programming of 256-bit private keys and 256-bit public keys. Users are also able to program such keys with 64 bits of redundancy. In addition, a one-time programmable 4k-bit space for customer usage is provided.

Conclusion

With the first-generation KeyStone architecture, Texas Instruments embarked on a journey focused on providing high performance at low power consumption. This journey continues with more innovations introduced in the KeyStone II platform and a strong roadmap of future devices. With the addition of multicore RISC processing alongside its industry-leading multicore DSP capabilities, TI offers leading processing capabilities with a very low power/performance ratio, multicore hardware-based cache coherency and broad industry software support. The integration of ARM Cortex-A15 processors into the KeyStone architecture brings a new level of performance to TI's multicore SoCs.

Learn more by visiting **www.ti.com/multicore**.

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 which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

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