Programming TI KeyStone II-based ARM + DSP devices using industry standard tools

Ellen Blinka
Texas Instruments
Dallas, TX, USA

Abstract— **Heterogeneous multicore architectures are used to meet the demanding needs of data and signal processing intensive applications such as those found in the high performance computing and defense markets. While heterogeneity is a powerful design tool for optimizing hardware for specific classes of algorithms, it comes with the challenge of how to best take advantage of the available cores in a simple software development framework. This paper discusses how these challenges are addressed by Texas Instruments for its Keystone II-based ARM + DSP devices using tools such as OpenMP, OpenCL and Open MPI along with easily accessible accelerated linear algebra and FFT libraries.**

I. INTRODUCTION

There are many examples of heterogeneous multicore processors including various combinations of CPU, DSP [1], GPU [2] and FPGA [3]. These devices were developed to provide dense compute power and to improve efficiency for specific applications, from large scale equipment such as industrial machines and defense equipment, to smaller devices such as mobile handsets. While they are able to provide solutions for increasingly strong performance in small areas and constrained power budgets, these devices also create new challenges including how to efficiently segment tasks between the different types of cores, and how to implement effective programming models.

Manually partitioning workloads to cores can offer high entitlement, but has drawbacks due to the static partitioning needing to be redone for each system configuration, a lack of portability, and the need for detailed knowledge of SoC architectures. To address this issue, multiple tools have been developed by the HPC and other computing communities including OpenMP, OpenCL, and MPI. These tools provide ways to abstract parallelization, dispatch tasks among heterogeneous compute units, and allow portability to multiple architectures.

Texas Instruments (TI) provides these tools for the KeyStone™ II architecture ARM® + DSP devices to allow developers to fully utilize the processing power of the ARM® Cortex®-A15 and C66x DSP cores. In addition to standard OpenMP, OpenCL, and OpenMP accelerator model implementations, TI has also implemented an OpenCL wrapper for OpenMP code to make existing parallel DSP code easily callable from the ARM. Likewise, TI has added transport layer support to Open MPI for its high speed SRIO and Hyperlink interfaces (in addition to Ethernet) to enable applications to use these for partitioning across SoCs. The result is a standard Linux-based gcc development environment on the ARM with standard access to multiple ARM cores, multiple DSP cores and multiple devices for acceleration.

The remaining sections of this paper describe the KeyStone II architecture, software development tools with an emphasis on features unique to their application to KeyStone II, and library examples that take advantage of these tools.

II. KEYSTONE II ARCHITECTURE FOR ARM + DSP DEVICES

TI's KeyStone II architecture combines up to four ARM Cortex-A15s with up to eight of TI's C66x DSP cores to create scalable SoCs for compute intensive embedded markets. The architecture provides a packet based message transfer system (Multicore Navigator), a switch fabric with over 2 Tbps capacity (TeraNet), and a multicore shared memory controller (MSMC) that provides both a high bandwidth path between cores and memory, and up to 18 MB on-chip SRAM with ECC capability[4]. The 66AK2H14 SoC provides up to 154Gbps of high-speed SERDES bandwidth from Ethernet, PCIe, SRIO and HyperLink, a TI proprietary interface that provides a high-speed, low-latency interface between KeyStone SoCs.

The ARM Cortex-A15 cores run at up to 1.4GHz and are capable of four single precision MACs (using the NEON extensions) or one double precision MAC per cycle [5]. The DSP cores operate at up to 1.2GHz and are capable of eight single precision or two double precision MACs per cycle.

III. KEYSTONE II SOFTWARE DEVELOPMENT TOOLS

A. OpenMP for Homogeneous Parallelization

OpenMP is the de facto industry standard for shared memory parallel programming and provides scalability and portability while allowing parallelization of existing code bases. OpenMP operates on homogeneous multicore systems or groups of homogenous cores within heterogeneous SoCs.

For KeyStone II-based SoCs, OpenMP can be used on the DSPs or on the ARM cores. For the DSPs, TI has integrated OpenMP support into its C66x compilers and DSP runtime software; for the ARM cores, TI is using the industry standard GCC compiler and corresponding runtime, ligomp, and Linux.

TI has also implemented a bare metal OpenMP DSP runtime using hardware queues found in the Multicore Navigator [6]. This enables the runtime to efficiently initiate and terminate parallel regions across DSP cores. The OpenMP runtime for C66x DSPs performs tasks such as inter-processor communication and cache control for memory consistency.

OpenMP provides an advantage when migrating from single core architectures since OpenMP constructs can be added to legacy applications to take advantage of the multiple

978-1-4799-6233-4/14/$31.00 ©2014 IEEE

DSP cores by annotating existing code with pragmas, avoiding major restructuring. OpenMP is useful in new developments as well since it abstracts parallelization across the multiple DSP cores of KeyStone II-based processors and allows developers to concentrate on other portions of their algorithm.

B. OpenCL and OpenMP for Heterogeneous Acceleration

OpenCL is a framework for heterogeneous devices where a number of host processors, ARM cores on KeyStone II-based processors, dispatch kernels that operate on accelerators, C66x DSP cores. Like OpenMP, OpenCL provides a scalable model for parallelization that is portable across various devices, and includes methods for data parallelism and task parallelism [7]. OpenCL offers the most control on TI's SoCs for heterogeneous acceleration as it has the lowest level abstraction and allows for direct control over data movement and memory management. TI's OpenCL implementation for KeyStone II differs from implementations takes advantage of the Multicore Navigator to dispatch workgroups and tasks across DSP cores.

In the standard OpenCL model multiple tasks must be dispatched to parallelize across the multiple DSP cores, creating a significant amount of overhead. To enhance this model, TI developed a method to dispatch OpenMP code with OpenCL. A TI extension allows an OpenCL kernel to act as a wrapper that invokes C functions containing OpenMP regions. An OpenCL host kernel executing on the ARM side dispatches an OpenCL kernel containing OpenMP regions to DSP Core 0 where the OpenMP main thread is invoked. When an OpenMP region is encountered while executing the main thread, DSP core 0 and other DSP cores work in parallel to execute the OpenMP region.

This is useful when existing multicore code using OpenMP is to be integrated into the ARM + DSP SoC since it is easily adapted into the OpenCL framework by adding a small wrapper function call inside an OpenCL kernel. This model also allows developers to use existing third party or TI developed libraries that leverage OpenMP.

The OpenMP accelerator model, part of the OpenMP 4.0 standard, was developed to enable execution of OpenMP code on heterogeneous SoCs that include host processors and accelerators. It supports shared and distributed memory systems between host and target devices. TI has implemented this for KeyStone II-based SoCs as a way to dispatch parallel tasks to multiple C66x DSPs from the ARM cores.

C. OpenMPI for Multiple Devices

TI has implemented an MPI model to facilitate communication between clusters of KeyStone II-based SoCs as well as concurrent operation of multiple instances of the same program across all nodes of a system. OpenMPI on KeyStone II supports multiple transports with varying bandwidths and latencies over Ethernet, SRIO and HyperLink. The extension of OpenMPI to SRIO and Hyperlink was achieved by enhancing the byte transport layer (BTL) of OpenMPI and allows MPI to be used in architectures where SRIO is the preferred backplane interface. The use with HyperLink, a TI proprietary interface, offers message passing over the high bandwidth connection between two KeyStone I or KeyStone II-based SoCs. OpenMPI uses standard APIs and can be configured to pick the "fastest" transport if multiple options are available. Additionally, OpenMPI has a large support ecosystem including profilers, debuggers, and training.

IV. Software Libraries

The overarching goal of development tool choices made by TI is to provide a seamless path from development of algorithms on a test platform (ex: Linux on x86) to running these algorithms on KeyStone II-based SoCs. The use of a standard Linux tool chain for the ARM cores is one step towards this goal; another is creating libraries that are callable using common signal processing APIs.

TI has developed optimized versions of foundational signal processing libraries, such as dense linear algebra (BLAS) and FFTs (FFTW), that abstract accelerator usage so that commands are dispatched using standard ARM callable APIs and the decision to execute on ARM cores, DSP cores, or multiple DSP cores across multiple devices or even multiple cards, is abstracted. These libraries leverage TI's OpenCL implementation for dispatch from ARM to DSP cores, OpenMP for parallelization across DSPs, and OpenMPI to communicate with multiple nodes over Ethernet, SRIO or HyperLink.

V. Conclusions

TI's KeyStone II-based ARM + DSP SoCs combine standard ARM cores with the power efficient fixed and floating point acceleration of C66x DSPs. To make these heterogeneous platforms easier to use and to help developers use the right core for the right work, TI has implemented various tools for use in abstracting, parallelizing, and dispatching algorithms to the DSPs including OpenMP, OpenCL, and the OpenMP Accelerator Model. TI has also created an OpenCL wrapper for OpenMP and extended OpenMPI to work over SRIO and HyperLink interfaces in addition to the standard Ethernet interface, and has developed optimized version of foundational libraries such as BLAS and FFTW that utilize OpenCL, OpenMP and OpenMPI and are callable with standard APIs.

References

[1] Texas Instruments, "Literature Number SWPS041D: OMAP4430 Multimedia Device Data Manual," [Online]. Available: http://www.ti.com/product/OMAP4430. [Accessed July 2014].

[2] Qualcomm, "Snapdragon 810 Product Brief," [Online]. Available: http://www.qualcomm.com/snapdragon/processors. [Accessed July 2014].

[3] Xilinx, "DS190v1.6 Zynq-7000 All Programmable SoC Overview," [Online]. Available: http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/. [Accessed July 2014].

[4] Texas Instruments, "Texas Instruments Literature: SPRS866: 66AK2H12/06 Multicore DSP+ARM KeyStone II System-on-Chip (SoC)," [Online]. Available: http://www.ti.com/product/66AK2H12. [Accessed July 2014].

[5] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones and A. Ramirez, "Tibidabo: Making the case for an ARM-based HPC system," Future Generation Computer Systems, vol. 36, pp. 322-334, 2013.

[6] E. Stotzer, A. Jayaraj, M. Ali, A. Friedmann, G. Mitra, A. P. Rendell and I. Lintault, "OpenMP on the Low-Power TI KeyStone II ARM/DSP System-on-Chip," OpenMP in the Era of Low Power Devices and Accelerators, Springer, pp. 114-127, 2013.

[7] Khronos, "OpenCL: The open standard for parallel programming of heterogeneous systems".