Consideration of the TMS320C6678 Multi-Core DSP for Power Efficient High Performance Computing

Richard M. Wallace[1], Bogdan Vacaliuc[1], Dwight A. Clayton[1], Bruce R. Chaffins[2], Olaf O. Storaasli[3] Dave Strenski[4], Dan Poznanovic[4]

[1] Measurement Science and Systems Engineering Division, ORNL, [2] Information Technology Services Division, ORNL, [3] Computer Science and Mathematics Division, ORNL, [4] Custom Engineering, CRAY Inc.

***Abstract*— To be effective, current heterogeneous computational architectures using accelerators** [1] **require extensive system knowledge and esoteric programming methods. These systems are composed of commodity processors integrated with Field Programmable Gate Arrays (FPGA) and/or Graphics Programming Units (GPU). A new Digital Signal Processor (DSP) architecture may be a viable alternative to the FPGA/GPU and avoid the performance problems associated with integrating accelerators into computer systems. The Texas Instruments TMS320C6678 multi-core DSP demonstrates equivalent power, cost efficiency to the best accelerators available today, and has an identical programming paradigm as multi-core CPUs. As recommended to the DOE** [2] **we propose the development of accurate, power-aware, performance models to explore the viability of incorporating a large number of these DSPs into heterogeneous computing systems compatible with HPC platforms.**

I. Introduction

Heterogeneous multi-processor systems represent the leading edge of HPC systems. These systems take advantage of different types of computing hardware by assigning computation tasks to the most appropriate hardware type. A common example is the hybrid CPU/FPGA combination. Recent breakthroughs in the use of GPUs as computation nodes have expanded the range of processors that could be used in a target hardware system. With both the FPGA and GPU as accelerators, very high computational efficiencies have been observed [1]. Not all codes are suited for FPGAs or GPUs. Programming FPGA-based processors is difficult for application programmers because the available tools are designed for hardware logic synthesis [3]. Although GPU programming is based on C/C++ languages such as CUDA or OpenCL, achieving good performance is challenging because data structure and access patterns must be rearranged. These obstacles render both FPGAs and GPUs unsuitable for many applications [4].

With the TMS320C6678 multi-core DSP, it is now possible to have high computational efficiency in a processor that can *participate in a large percentage of the source code* [5]. This is achieved without dependencies on out-of-line code such as in FPGA and GPU systems.

II. Computing Efficiency

A survey of state-of-the-art processors in Table 1 highlights important attributes of the most power-efficient processors currently available. The table is sorted by watts per gigaflop (W/GF) from lowest to highest. Higher heat dissipation devices have orange shading and lower dissipation devices have green shading.

Device	Cores	Frequency (MHz)	GF (DP/SP)	TDP (W)	Cost ($)	W/GF (DP/SP)	DP Efficiency (vs. Fermi)	SP Efficiency (vs. Fermi)
AMD Cypress HD5870 (40nm) [B]	1600	850	680/2720	188	370	0.28/0.07	166%	333%
TI TMS320C6678-1250 (40nm) [C]	8	1250	40/160	17	200	0.43/0.11	108%	216%
Nvidia Fermi M2050 (40nm)	448	1150	515/1030	238	2500	0.46/0.23	100%	100%
Intel i7-2715QE (32nm)	4	2100	67/134	45	1000	0.67/0.33	69%	69%
IBM Power7 (45nm)	8	4000	256/256	200		0.78/0.78	59%	29%
AMD Opteron 6164 HE (45nm)	12	1700	82/163	65	872	0.80/0.40	58%	58%
Fujitsu SparcVIIIfx (45nm)	8	2000	64/128	58		0.91/0.45	51%	51%

Table 1 Potential Ultrascale Processors [A, D]

A. Thermal design power (TDP) is the maximum amount of power the chip package can dissipate; the cooling system must be designed to accommodate this.
B. For the HD5870, although the peak GFLOPS is greater, application-oriented benchmarks show actual throughput to be less [15] than M2050, highlighting the need for performance modeling when composing systems of systems. Also, see the PARSEC benchmark at http://parsec.cs.princeton.edu.
C. ±7W; 24W TDP for the TI chip is from the packaging specification (absolute max); some TI literature uses 10W
D. Other processors such as Intel's SCC and Tensilica's Xtensa LX3 or CommX cores are difficult to consider since they are experimental or require an ASIC development. Tilera's GX-100 does not have strong floating-point capacity.

Values are taken from vendor datasheets, or conference reports. We recognize the roadmap of NVidia (with Kepler and Maxwell)[1] and AMD (with Cayman and Bulldozer)[2] but we do not consider them for this proposal since they are not released or documented products.

The metric of W/GF is one of the key measures of the operational cost of an installed system. From a system architecture perspective,

> *... shallower pipelines with in-order execution have proven to be the most area and energy efficient. Given these physical and micro architectural considerations, we believe the efficient building blocks of future architectures are likely to be simple, modestly pipelined (5-9 stages) processors, floating point units, vector, and SIMD processing elements.* [6]

Compared with the current state-of-the-art microprocessors fielded by AMD, Fujitsu, IBM, Intel, and NVIDIA, the DSP microprocessor from Texas Instruments embodies this recommendation best.

As an example, the "Blue Waters" [7] computer being built at the NCSA will reach 10 Petaflops and dissipate 7.5MW from 37500 chips. The facility is designed for 17.6MW input and has an overall conversion efficiency of 85% (15MW), leaving about 7.5MW for the I/O, cooling and storage functions[3]. If the machine were built using TI DSP, it would need 250,000 DSP chips, but dissipate a total of only 4.25MW; this represents a potential savings of over 26 million kWh ($1.6M) annually if run continuously at peak capacity[4].

III. Need for Performance Modeling

The 2011 ORNL agenda has goals of sustaining the world's most capable complex for computational science through world leadership in transformational research and scientific discovery. Developing hardware, software, and staff expertise for Ultrascale applications performance at several hundred Petaflops is the next plateau required to maintain ORNL's capabilities as a world leader. In order to do this, new computer systems must be designed, constructed, and programmed.

Kerbyson, Wasserman and Hoisie have shown that

> *...performance models have been used to explore the possible achievable performance on future, hypothesized, architectures.* [8]

These systems are composed of hundreds of thousands of computation nodes interconnected with a variety of high-speed data interfaces. The Blue Waters project is using performance modeling and simulation extensively during its design and construction [7]. In the International *Exascale Software Project Roadmap*, Dongarra and other authors write that, "…algorithms must be developed that are a good match to the available hardware. One of the most challenging demands is power; algorithms that minimize power use need to be developed. ***This will require performance models that include energy***." [9] *Ed. emphasis added*. Our proposal is to construct usable performance models for a specific computation node (TI DSP) to see if it can be used as the basis of a larger computational system effectively.

IV. Research Plan

ORNL and Cray, Inc. are partnered to develop a method of constructing performance models that can be used to evaluate new computing architectures that integrate into existing HPC systems. Cray researchers will develop a multi-mode test application that will have a function-call interface. This interface represents a work-unit or 'kernel subroutine' that would normally be executed on a certain processor or accelerator. The interface requires that, in addition to supplying the correct result, the subroutine shall provide execution time and average energy consumed by the computation. The test application will aggregate the estimates made by the subroutine calls and report them in a structured manner for comparison and analysis. ORNL researchers will develop the subroutines that are called by Cray's interface. These subroutines are specific to each application that is evaluated, according to the computational model established by the benchmark code. The original work lies in composing an accurate performance model that is parameterized by the data type, vector sizes, and system-related metrics.

The research environment block-diagram is shown in Figure 1. The subroutines are shown within the subroutine kernel. Micro benchmarks and simulations are conducted off-line to create the data and metadata needed for the analytic performance models. GPU performance models are also created because the comparison between DSP and GPU implementations is of interest to both Cray and ORNL.

---

[1] http://www.semiaccurate.com/2010/09/23/nvidias-kepler-and-maxwell-barely-beat-moores-law/

[2] http://lensfire.blogspot.com/2010/11/amd-processor-roadmap-2011-2013.html

[3] Timothy P. Morgan, "IBM drops Power7 drain in 'Blue Waters'", The Register, Dec 8, 2008, http://www.theregister.co.uk/2008/12/08/power7_bluewaters_data_center/

[4] $0.0623/kWh industrial rate from 2009, see http://www.eia.doe.gov/emeu/aer/txt/ptb0810.html

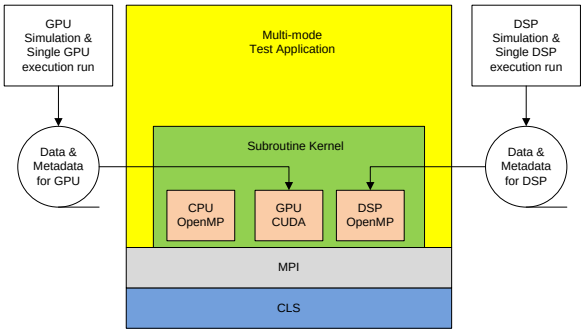

Figure 1 Simulation Block Diagram

The research effort is segmented into four phases: establishment of the benchmark codes to be used, development of DSP and GPU performance models, integration of the software on Cray XE6 hardware[5], execution of the benchmarks and analysis of the results.

Phase 1 – Establish Exemplar Benchmark Code and Interfaces

This phase implements the initial group of benchmarks to be executed onto the CRAY XE6 target hardware. Table 2 details the three application benchmarks and an alternative for evaluation.

Benchmark	Focus
Fast Fourier Transform (FFTW)	DSP-favorable
High Performance LINPACK (HPL)	GPU-favorable
NASA General Purpose Solver (GPS)	multi-core CPU favorable
Alternative to GPS: Computational Fluid Dynamics (OpenFOAM)	single CPU favorable[6,7]

Table 2 Benchmark Focus

These benchmarks are chosen as they have different computational and data access patterns that favor particular architectures. We are interested in evaluating the operation of the DSP when integrated into a system of disparate processor types, so it makes sense to use applications that are both favorable and unfavorable to the types it might displace. As part of this task, the function interface is established between the ORNL performance modeling subroutines and the Cray multi-mode test environment. The products of this task are three multi-mode test applications. An alternate to GPS is OpenFOAM; this application poses challenges for GPU and OpenMP but may implement well on the DSP, as the DSP will execute an asymmetric multi-processing model equally well.

Phase 2 – DSP Simulink Model

In this phase we build a MATLAB/Simulink model of the DSP computation that produces meta-data regarding performance characteristics of a given computational model. We recognize the important contributions of Alam and Vetter in developing the modeling assertions (MA) approach [10]. Simulink is used to model the computational structure of the algorithm for the benchmarks in Table 2. We chose this tool because we require flexibility in composing systems of TI DSP chips into accelerators. For example, it takes 8 DSP chips to achieve the computational equivalent of a Fermi GPU. Our model needs to take into consideration inter-core and inter-chip communications and latencies. Simulink is an ideal environment for composing and analyzing such structures.

This effort will generate the necessary system-related metrics needed by the performance modeling subroutines, as well as the code for the routines themselves. The product of this task is a library of functions that are called by Cray's multi-mode test applications.

Phase 3 – GPU Model

This phase will run the benchmark codes externally on a stand-alone GPU system to gather specific execution and GPU-chip reported thermal data. A database of performance observations will be collected per-application. Code will be generated to interpolate values from this database according to passed-in parameters. The product of this task is the comparison of the GPU versus the DSP throughput and power performance.

Phase 4 -- Execution of Models and Reporting Results

This phase produces a report comparing the execution time and power consumption (electrical and thermal dissipation) in the form of a research paper to be submitted to an HPC conference such as SAAHPC'11 or SC'11.

V. Prior Work

The two key technologies factoring toward the realization of a viable performance model of a multi-chip system using the TMS320C6678 DSP are the use of Simulink as a modeling tool and the availability of OpenMP in the compiler environment. The researchers in this proposal also are uniquely qualified to conduct this project due to their background and experience.

[5] http://www.cray.com/Assets/PDF/products/xe/CrayXE6Brochure.pdf
[6] multi-processor via MPI and domain decomposition; the solvers execute as a single process; e.g. http://www.openfoam.com/docs/user/damBreak.php#x7-610002.3.11
[7] Hallberg, Hallgren and Lopez have reported on their effort using OpenMP to parallelize OpenFOAM: http://openfoam-openmp.googlecode.com/files/DATX02-10-01_Rapport.pdf

Common Radar Environment Simulator

The research team has experience with TI DSP components and is involved in the ongoing work on a common radar environment simulator providing the basis for model driven engineering. The simulator is programmed using a Simulink-based framework for generating heterogeneous multi-processor applications [11]. This is accomplished using a processor-aware plug-in that generates code optimized for Texas Instruments DSPs and Xilinx FPGAs.

OpenMP

OpenMP is a compiler extension that is designed to integrate into existing source code via the addition of #pragma directives. When using compilers that are not OpenMP aware, the #pragma directives are silently ignored, offering a pedantic mechanism for providing "hints" to the compiler of the programmer's intention and the parallelism that might be exploited[8]. Considerable work has been done within the high performance community with OpenMP. Chapman, et. al., have evaluated OpenMP on the TI multicore DSPs [5] .

Application Accelerators

The research team also has experience programming computation accelerators to speed-up applications such as genome sequencing, FFT, matrix equation solution, weather and climate prediction, and molecular dynamics by a factor of 100 using FPGAs [12], [13], [14]. Applications were tested on FPGA systems from 1 to 150 FPGAs. Despite the remarkable speedups achieved, the research team considers the major problem of FPGA usage being programming ease.

VI. Future Application

The analytical results and the proof-of-concept tools developed during the proposed effort will be of direct interest to companies that are investing in real-time embedded and energy-efficient high performance computing systems for applications such as guidance systems, remote vision systems, security, robotics, automotive and manufacturing. Many of these companies are looking to use private or public clouds to gain access to computing resources[9,10]. The cloud computing market is expected to exceed $16B by 2013, an investment research firm predicts[11]. Our work can help these companies make informed choices regarding their investment.

The Ubiquitous High Performance Computing 2010 BAA[12] sets a goal of 50GFlops/W (0.02 W/GFlop, pg. 17) for the HPL benchmark for a proposed system. Our proposal addresses the needs of these requirements: "1) development and optimization of Extreme-Scale architectures, technologies, execution models, and the critical co-design of hardware and software; 2) low-energy architectures and protocols for logic, memory, data access, and data transport; 3) dynamic systems that adapt to achieve optimal application execution goals…"

The International Exascale Software Roadmap [9] lists 2011 milestones as, "Initial solicitations for software development programs based on the software roadmap for international partners," with "Initial software deliveries and evaluations" occurring in 2012-2013.

OMB Memorandum M-10-30[13] states "Prioritize R&D on advanced vehicle technologies, **particularly modeling and simulation**…" [*Ed. emphasis added*]. This research supports modeling and simulation efforts by enabling informed choices regarding the construction and selection of computing systems for these programs.

Additionally, the work in this research also serves as a model framework for incorporating not just DSP, but arbitrary new computation architectures such as Tilera, Achronix and other devices into high performance multiprocessor systems of interest to the DOE and DOD. Cray is interested in understanding the role that DSP based accelerators can play, and the range of applications that can benefit from their use.

[8] http://developers.sun.com/solaris/articles/openmp.html
[9] http://www.synopsys.com/Systems/Saber/CapsuleModule/2008-01-0288.pdf
[10] http://www.fastmr.com/catalog/product.aspx?productid=106345&dt=t
[11] http://www.stockbloghub.com/2010/12/19/isln-16-7-billion-cloud-computing-industry-to-be-dominated-by-one-company/62729
[12] DARPA-BAA-10-37 "Ubiquitous High Performance Computing" (March 2010) https://www.fbo.gov/utils/view?id=3d7cfa30b2b1a93332a444047dea52d8
[13] OMB Memorandum M-10-30 (July 2010) http://www.whitehouse.gov/sites/default/files/omb/assets/memoranda_2010/m10-30.pdf

REFERENCES

[1] AR Brodtkorb, C Dyken, TR Hagen, JM Hjelmervik, and OO Storaasli, "State-of-the-art in heterogeneous computing," *SCIENTIFIC PROGRAMMING*, vol. 18, no. 1, pp. 1-33 , 2010, http://babrodtk.at.ifi.uio.no/files/publications/brodtkorb_etal_star_heterocomp_final.pdf.

[2] DOE/Boston University, "Opportunities and Challenges of Exascale Computing," Report of the ASCAC Subcommittee on Exascale Computing http://www.er.doe.gov/ascr/ASCAC/Reports/Exascale-Subcommittee-Report.pdf, 2010.

[3] HPC Wire, "Déjà Vu All Over Again -- Lessons from the past offer insight into HPC advancements of today…a conversation with Steve Wallach of Convey Computer," *HPC Wire*, Nov. 2009, http://www.hpcwire.com/features/Deja-Vu-All-Over-Again-70250362.html?viewAll=y.

[4] Cristian Grozea, Zorana Bankovic, and Pavel Laskov, "FPGA vs. Multi-core CPUs vs. GPUs: Hands-On Experience with a Sorting Application," *Lecture Notes in Computer Science*, vol. 6310, pp. 105-117, 2011, http://dx.doi.org/10.1007/978-3-642-16233-6_12.

[5] B. Chapman et al., "Implementing OpenMP on a high performance embedded multicore MPSoC," *Parallel & Distributed Processing*, pp. 1-8, May 2009, http://dx.doi.org/10.1109/IPDPS.2009.5161107.

[6] K. Asanovic and et. al., "The Landscape of Parallel Computing Research: A View From Berkeley," Univ. of California, Berkley, Berkley, CA, Technical Report No. UCB/EECS-2006-183 http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf, 2006.

[7] NCSA, "Blue Waters: Sustained Petascale Computing," Univ. of Illinois, Urbana, IL, Presentation http://www.ncsa.illinois.edu/BlueWaters/pdfs/snir-power7.pdf, 2010.

[8] Darren J. Kerbyson, Harvey J. Wasserman, and Adolfy Hoisie, "Exploring Advanced Architectures Using Performance Prediction," in *International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems*, Washington, DC, 2002, pp. 27 - 37, http://doi.ieeecomputersociety.org/10.1109/IWIA.2002.1035016.

[9] J. Dongarra, P. Beckman, and et al., "The International Exascale Software Roadmap," *to appear in International Journal of High Performance Computer Applications*, vol. 25, no. 1, 2011, http://www.exascale.org/mediawiki/images/2/20/IESP-roadmap.pdf.

[10] Sadaf R. Alam and Jeffrey S. Vetter, "A framework to develop symbolic performance models of parallel applications," in *20th international conference on Parallel and distributed processing*, Rhodes Island, Greece, 2006, pp. 320-320, http://dx.doi.org/10.1109/IPDPS.2006.1639625 .

[11] Mark A. Buckner et al., "From Requirements Capture To Silicon: A Model-Driven Systems Engineering Approach To Rapid Design, Prototyping And Development Used In The Oak Ridge National Laboratory's Cognitive Radio Program," in *Proceedings of the SDR '08 Technical Conference and Product Exposition*, Washington, D.C., 2008, p. Session 4.6, http://groups.winnforum.org/p/cm/ld/fid=56 .

[12] Olaf Storaasli and Dave Strenski, "Beyond 100x FPGA Speedup," in *Proceedings of 2009 CUG*, Atlanta GA, 2009, http://ft.ornl.gov/~olaf/pubs/StoraasliCUG09.pdf .

[13] Michael Babst, Roderick Swift, and Dave Strenski, "Evaluation of running FFTs on the Cray XD1 with attached FPGAs," in *Proceedings of the Cray User Group*, Albuquerque, NM, 2005, http://cug.org/5-publications/proceedings_attendee_lists/2005CD/S05_Proceedings/pages/Authors/Strenski/Strenski_paper.pdf .

[14] Dave Strenski, Prasanna Sundararajan, and Ralph Wittig, "The Expanding Floating-Point Performance Gap Between FPGAs and Microprocessors," *HPCwire*, Nov 2010, http://www.hpcwire.com/features/The-Expanding-Floating-Point-Performance-Gap-Between-FPGAs-and-Microprocessors-109982029.html .

[15] A. Danalis et al., "The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite," in *Third Workshop on General-Purpose Computation on Graphics Processors*, Pittsburgh, PA, 2010, pp. 63-74, http://dx.doi.org/10.1145/1735688.1735702 .