Using VSIPL as an Embedded DoD Application Programming Interface (API) on DARPA Polymorphous Computing Architectures (PCA)

Joe Cook (joseph.j.cook@lmco.com)[2], Steve Crago (crago@east.isi.edu)[1], Kristin Daly (kristin.l.daly@lmco.com)[2],
Karandeep Singh (karan@east.isi.edu) [1], Matthew French (mfrench@east.isi.edu) [1], Rick Pancoast
(rick.pancoast@lmco.com)[2], Jinwoo Suh (jsuh@east.isi.edu)[1]
University of Southern California – Information Sciences Institute[1], 3811 N. Fairfax Drive, Suite 200, Arlington, VA 22203,
Phone: 703-243-9422, Fax: 703-812-3712 and Lockheed Martin, Maritime Systems and Sensors (MS2) [2], 199 Borton
Landing Road, Moorestown, NJ 08057, Phone 856-722-2354, Fax: 856-722-2055

This briefing describes an effort to implement advanced embedded DoD signal processing algorithms, such as radar Pulse Compression, on multiple Polymorphous Computing Architectures (PCAs), under development by DARPA, using the VSIPL industry standard signal processing API. Critical embedded, high performance radar processing algorithms for shipboard ballistic missile defense have been implemented at Lockheed Martin MS2 in COTS PowerPC architectures using equipment from various vendors, including CSPI and Mercury Computers, utilizing standard API libraries such as VSIPL and MPI. Of particular interest is how can we easily port these embedded applications from conventional COTS architectures for execution on various embedded PCA morphable processor architectures? This successful demonstration of Application Portability is key to DoD acceptance of non-conventional computing architectures, such as PCA, for embedded tactical processing applications. This effort successfully demonstrated:

- A critical embedded DoD signal processing benchmark (Frequency Domain Radar Pulse Compression) using industry standard VSIPL, executing on the DARPA RAW Polymorphous Computing Architecture

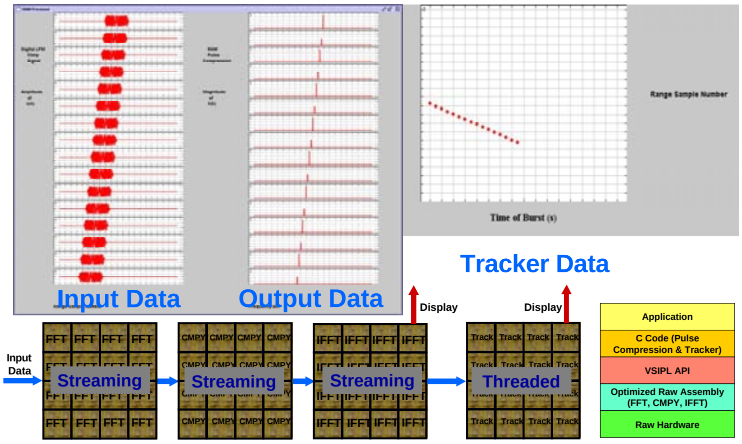
- A successful Morph from C-VSIPL Streaming Pulse Compression to a C-VSIPL Threaded Tracking Function, on an actual RAW processor in the USC-ISI Laboratory

- A successful port of the RAW C-VSIPL Streaming Pulse Compression code and C-VSIPL Threaded Track Processing code to an alternative PCA Architecture (UT Austin TRIPS Processor Simulator)

- A Java-based GUI to display the input data, output data and PCA Processor Status while running the C VSIPL benchmarks

- Key embedded PCA technology, running a DoD Benchmark, written in VSIPL, for key US Defense Department representatives and developed a concept to transition PCA-based architectures into embedded DoD tactical systems

These new embedded processor technologies, being developed by the DARPA Polymorphous Computing Architectures (PCA) Program, promise to provide alternative, morphable embedded computing architectures as compared to classical, static PowerPC or Pentium-based architectures. The PCA architectures can adapt their structure by morphing their architecture as the mission and processing requirements change, thereby enhancing performance as the mission dictates architecture changes, or morphs. The PCA embedded processor designs promise to save significant equipment space and to provide a lower cost solution by providing an architecture that can adapt to the mission at hand. The issue that arises is the potential complication of efficiently programming a new, non-conventional computing architecture. DoD contractors cannot afford to hire specialized architects and programmers that are adept at programming niche architectures. By providing industry standard APIs, such as VSIPL and MPI, for new PCA architectures, we have demonstrated legacy applications implemented in new, morphable embedded PCA hardware and middleware architectures with industry standard C and C++ programming methodology.



Pulse Compression & Track on RAW GUI

In this briefing, we describe the successful implementation of an embedded processing application that includes streaming Pulse Compression radar signal processing, followed by an architecture morph and threaded track processing. Processing functions are coded using a high level language (C) and an industry standard API (VSIPL) on multiple PCA host processors which execute the function on embedded PCA architectures (in this case, the MIT RAW processor hardware and the UT Austin TRIPS Simulator). Benchmark results are provided that demonstrate the same C VSIPL benchmark algorithm

executing on a conventional, static embedded architecture (such as PowerPC) and two different PCA morphable architectures (MIT RAW and UT Austin TRIPS). Finally, the graphical results of the portable benchmark algorithm, running identical source code on an actual PCA RAW processor and the TRIPS Simulator, using C VSIPL, will be demonstrated.