Introduction to
Demonstration Applications
for the
Cmpware CMP-DK
(Demo Version 2.0 for Eclipse 3.0)

Cmpware, Inc.

Introduction

The *Cmpware Configurable Multiprocessor Development Kit (CMP-DK)* is a
multiprocessor simulation and software development environment. Its provides fast
and efficient modeling of multiprocessor architectures as well as support for software
development on such systems. The goal of supporting software development is
achieved by providing an interactive, display-rich environment that permits large
amounts of information to be displayed in a fast, simple and uncluttered format. Such
capabilities are essential in analyzing the behavior of multiprocessor systems.

This demonstration version of the *Cmpware CMP-DK* (version 2.0) for Eclipse 3.0 and
higher contains all features of the standard toolkit, but restricts the simulation model to
a 3 x 3 heterogeneous array of MIPS32 and SPARC-8 processors. All simulation
capabilities and displays are included. This includes:

- Source Code Tracing
- Source Code Variables
- Disassembly
- Memory Display
- Power Estimator
- General Purpose Registers
- Special Purpose Registers
- Command Line Interface
- Link Utilization

Demonstration Applications

Avaliable for use with the *Cmpware CMP-DK* version 2.0 is a series of demonstration
applications which are presented to introduce some of the features in the *CMP-DK*.
These applications start with small, simple programs gradually building up to more
complex applications exploiting relatively low-level parallelism. These demonstrations
stand alone and can be studied in any order, but it is best to start with the early

examples, which are smaller and simpler and build up to the larger ones. This provides a tutorial-like introduction to the features in the *Cmpware CMP-DK*.

While these demonstrations cover the application development aspects of this tool, much of the power in the *Cmpware CMP-DK* is in the ability to quickly model relatively complex multiprocessor systems. This modeling activity is reserved for licensed copies of the software. For more information on getting licensed copies of the *Cmpware CMP-DK,* contact Cmpware at *info@cmpware.com*.

The applications in this tutorial are as follows:

🔵 **Simple**: This application is one of the simplest C programs possible: an infinite loop which increments a single value. While not exercising any of the multiprocessor aspects of the *Cmpware CMP-DK*, this applicaion is a simple way to get familiar with executing compiled code as well as the use of the various interactive displays.

🔵 **Ping Pong**: This application is one of the simplest multiprocessor programs possible. It is a pair of programs which send a value back and forth, with the first node incrementing the value. This is a simple way to explore the multiprocessor capabilities of the *Cmpware CMP-DK*, including the ability to quickly probe system data from different processors as well as the views offering system level data.

🔵 **Hetero**: One of the most powerful features of the *Cmpware CMP-DK* is its ability to manage processors of different types. In the default demonstration configuration, a mix of MIPS32 and SPARC-8 processors are used. This demonstration is identical to the *Ping Pong* demonstration, except that one of the processing nodes used is a MIPS32 and the other is a SPARC-8. This allows the exploration of support for heterogeneous multiprocessing in the *Cmpware CMP-DK*.

🔵 **FIR Filter**: This demonstration program shows a popular Digital Signal Processing (DSP) function implemented across multiple processors. This function, the *Finite Impulse Response (FIR)* filter is used in a variety of application, particularly audio processing. This demonstration shows the FIR code running self-hosted on a PC or workstation, then on a single processing node in the *Cmpware CMP-DK*, then on multiple processing nodes. The ability to extend this simple algorithm to a large number of processors is also demonstrated using the licensed version of the software.

🔵 **AES Encryption**: This application demonstrates the *Advanced Encryption Standard (AES)*. As with the FIR filter, a self-hosted version is run on a PC or workstation, then on a single processing node in the *Cmpware CMP-DK*. The algorithm is then parallelized across multiple processing nodes. Parallel execution on from one to twelve nodes is demonstrated.

🔵 **FFT Filter**: This application demonstrates the *Fast Fourier Transform (FFT)* filter, another popular digital signal processing (DSP) algorithm. This example demonstrates the use of shared memory in the *Cmpware CMP-DK* as well as the synchronization necessary for correct operation. As with the other algorithms, a self-hosted version is compiled and run, then a single node version is run on the *Cmpware CMP-DK*. A parallel version is then run on from one to six processors.

🔵 **FFT Filter 2**: This second implementation of the FFT filter is identical to the previous version, except the shared memory approach is replaced by communication channels. This simplifies the software somewhat, permitting explicit synchronization to be avoided. Performance is compared to the shared memory version of the FFT.

The source and compiled code for these demonstration applications can be downloaded from the Cmpware Web site as a compressed ZIP archive at:

http://www.cmpware.com/Apps/CmpwareApps_2_0.zip

Conclusions

The *Cmpware CMP-DK* is a rich display environment combining fast simulation and flexible multiprocessor modeling. This makes it an ideal environment for architecture modeling and software development for these systems.

While the execution and display features of the *Cmpware CMP-DK* are notable, much of the power of the system lies in its ability to quickly and flexibly construct processor, network, link and multiprocessor models. This modeling capability is a large part of the commercial version of the *Cmpware CMP-DK*.

For more information on the commercial version of the *Cmpware CMP-DK* see our web site at:

http://www.cmpware.com/

or send an email to:

info@cmpware.com