IBM PowerNP™ NP4GS3 Network Processor Solutions

Introduction

The IBM PowerNP NP4GS3 network processor can form the core of fast, powerful, and scalable network hardware and systems. IBM can provide the tools and services to exploit the NP4GS3's full potential and efficiently design, develop, and bring an NP4GS3-based product to market as quickly as possible. IBM offers a reference platform, reference hardware, and reference software to aid in design and initial development. Basic and advanced software offerings combined with IBM's technical expertise can help guide the development process and augment the power of the NP4GS3.

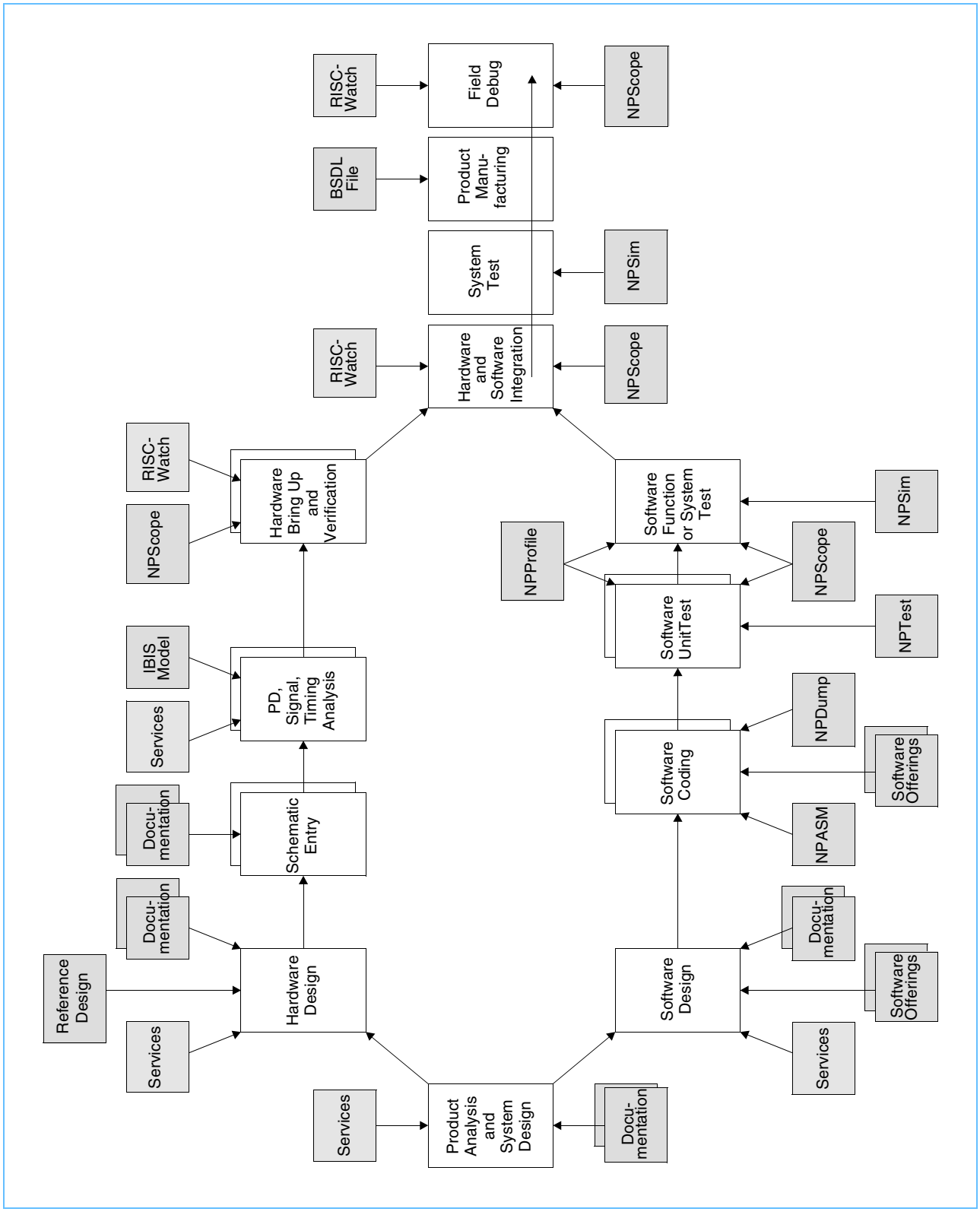
The Software Developer's Toolkit is an example of the type of versatile tools that IBM can provide. It contains a PowerNP network processor simulator that allows control point and picocode software to be developed concurrently with hardware. The Toolkit simulates multiple interconnected NP4GS3s and control processors forwarding traffic from a variety of sources. It also provides timing data for analyzing picocode performance. Frequently, a complete software stack can be tested before hardware even exists.

Once hardware is available, the Toolkit can ease the development and testing of new code on a single host computer. It can also help debug the hardware. All of these features can shorten the development cycle and decrease the product's time to market. For detailed information about the toolkit, see the *IBM Power Network Processor Software Overview.*

This overview summarizes PowerNP products and services and explains how each can enhance the product development process. *Figure 1* shows a typical product development cycle and the IBM tools and services that can help during each phase. In all of the figures in this document, IBM products and services appear in shaded boxes.

For detailed information on particular PowerNP tools or offerings, contact your IBM sales representative.

Figure 1. IBM PowerNP Tools and Services Used During the Product Cycle



PowerNP NP4GS3 Reference Platform

The PowerNP NP4GS3 Reference Platform is comprised of a 5U high, 4-slot chassis with forced-air cooling, an internal power supply, a backplane, and a PowerPC 750™ compact PCI control point processor board. The chassis can be free-standing or mounted in a 19-inch open rack that meets the EIA RS-310C universal rack-mounting hole pattern standard. The cPCI processor board is installed in the bottom slot of the chassis, and an IBM Packet Routing Switch PRS28.4G board can be installed in the top slot. An NP4GS3-based board or an I/O interface card can be installed in each of the two center slots. Up to eight NP4GS3 boards in four Reference Platforms can be interconnected via the PRS28.4G boards.

The NP4GS3 and its subsystem components are packaged on a carrier board that contains two connectors for I/O interface card attachments. The following I/O interface card options are available:

- Twenty-port Fast Ethernet 10/100 TX (maximum of one per carrier board)
- Dual-port Gigabit Ethernet SX
- Single-port OC-48 Packet over SONET (occupies both carrier board connectors)

Hardware Reference Design

IBM offers an NP4GS3 hardware reference design as a starting point to help you develop your own NP4GS3 board. It includes hardware design guidelines, schematics, and a bill of materials. Detailed information about the NP4GS3 device is contained in the *IBM PowerNP NP4GS3 Network Processor Datasheet*. Contact your IBM sales representative for more information.

Software Offerings

IBM provides two software offerings to make it easier to develop NP4GS3-based systems.

The Base Software Offering provides a thin layer of control point and picocode infrastructure software that can be used as a template for a developer's software or as an example for a developer's implementation. This package requires the developer to write the data plane software.

The Advanced Software Offering provides completely functional control point and picocode that provides switch and router capability. This package accelerates the software development process, allowing the developer to concentrate on higher-level functions, adding functions, or tailoring the software to unique requirements.

IBM PowerNP NP4GS3 Base Software Offering

The Base Software Offering contains sample software and picocode. The Base Software Offering can be executed on the simulation model provided as part of the Software Developer's Toolkit and can be used as a template for creating your own software for the NP4GS3.

The Base Software Offering includes:

- Control picocode for performing control functions, such as power-up, diagnostics, control frame processing, memory management, and table management
- Sample picocode for forwarding packets using Layer 2 bridging
- A control interface specification for communication between the control point code and network processor picocode using control frames

IBM PowerNP NP4GS3 Advanced Software Offering

The Advanced Software Offering contains functional control point software and picocode. The Advanced Software Offering can be executed on the simulation model or on the PowerNP4GS3 Reference Platform. The Advanced Software Offering contains infrastructure software for multiple environments that can be ported to unique environments. It also supports many of the protocols used in WAN and campus applications.

Network hardware developers can leverage the software system and protocols provided with the Advanced Software Offering to enable faster time-to-market.

The Advanced Software Offering contains code and documentation for network processor application services (control point software) and network processor picocode. Some of the protocols supported by the Advanced Software Offering include:

- Layer 2 functions
- IP forwarding (unicast and multicast)
- Multi-field classifier (for example, for filter rules)
- Multiprotocol Label Switching (MPLS)

Control point reference software documentation and code are provided for Red Hat Linux. This software can easily be ported to other operating systems.

For more information on the Base and Advanced Software Offerings, see the *NP4GS3 Base Software Offering Overview* and the *NP4GS3 Advanced Software Offering Overview*. Additional details are available in the *IBM PowerNP NP4GS3 Advanced Software Offering User's Guide*, the *IBM PowerNP NP4GS3 Reference Software for Linux User's Guide*, and the *IBM PowerNP NP4GS3 Reference Platform Developer's Code Kit*. Contact your IBM sales representative for information on obtaining these documents.

Software Developer's Toolkit

The Software Developer's Toolkit includes the following components:

- Core Simulation Model
- NPASM, an assembler
- NPDump, an interpreter of the picocode binary image file
- NPScope, a debugger
- NPSim, a network simulator
- NPProfile, a performance profiler
- NPTest, a test-case generator

For compatibility with various development environment needs, the Software Developer's Toolkit is supported on several common operating systems, including Microsoft® Windows®, Sun Solaris, and Red Hat Linux. The Software Developer's Toolkit uses the Tool Command Language (TCL) and Toolkit (Tk). The Toolkit provides a set of TCL extensions that enable users to modify and enhance the Toolkit by writing TCL scripts.

The Software Developer's Toolkit also provides demonstration scripts to help users understand how to use the Toolkit.

Core Simulation Model

The core simulation model is a software framework that provides an accurate simulation of the functions of the NP4GS3 network processor. The model can “execute” (emulate) thousands of picocode instructions per second, allowing quick testing of code under various conditions.

NPASM

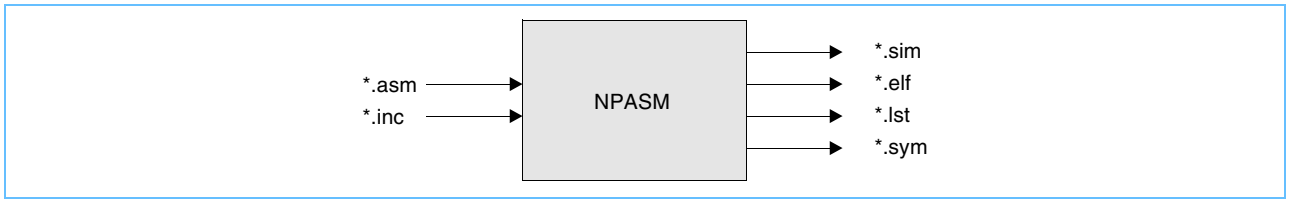
NPASM is the assembler for creating images designed to execute on the NP4GS3. The assembler instruction set is specifically designed for networking applications, allowing fast and efficient packet processing when combined with the powerful coprocessors embedded in the NP4GS3. NPASM assembler instructions seem like a higher-level language to the NP4GS3 because they support integer expressions, string expressions, looping constructs, macros, and more.

NPASM generates files used to execute picocode on the core simulation model or the NP4GS3, as well as files that picocode programmers can use for debugging. Depending on what options are specified, NPASM can generate the following files:

- Picocode load file for the core simulation model (*.sim)
- Symbol table that cross-references label names with addresses (*.elf)
- Listing of the assembled code (*.lst)
- Symbol table listing (*.sym)

Figure 2 illustrates how NPASM processes files.

Figure 2. NPASM



For detailed information on NPASM, see the *IBM PowerNP NP4GS3 Assembler Language Programmer’s Guide and Instruction Summary*. Contact your IBM sales representative for information on obtaining this document.

NPDump

NPDump is a tool that takes the picocode binary executable (*.elf) file produced by NPASM and displays it as ASCII text. This enables programmers to validate that data structures were coded correctly by comparing output with the actual data being used.

NPScope

NPScope is a debugger with a TCL/Tk graphical user interface (GUI). It contains a comprehensive set of features that allows a developer to:

- Access and monitor core simulation model components, such as registers, control store memory, data memory, and queues.

- View, execute, and debug PowerNP picocode. Execution can be controlled step-by-step using break-points or on a per-frame basis.
- Inject frames to test picocode processing.
- Detect and identify common picocode coding errors, such as an invalid access to control store or data memory.
- Retrieve clock cycle timing information for use in picocode performance analysis.
- Configure the core simulation model to include only subsets of the PowerNP network processor components to tailor execution to focus on certain paths. For example, a developer can:
  - Inhibit enqueuing frames from the enqueue coprocessor.
  - Restrict the flow of frames at different points between the processor and the I/O ports.
- Test and debug NP4GS3 hardware components and picocode executing on the hardware working in conjunction with a RISCWatch probe).

For more information on NPScope, refer to the *IBM PowerNP NP4GS3 Software Developer's Toolkit User's Guide* and the *IBM PowerNP NP4GS3 Generating and Running the NP4GS3 Demonstration Test Cases for NPScope* manual. Contact your IBM sales representative for information on obtaining these documents.

NPSim

NPSim provides a TCL interpreter with extensions for accessing the core simulation model using a set of documented primitives. These extensions allow developers to create TCL applications and to execute control point and picocode software in a simulated network environment.

The Software Developer's Toolkit includes a sample NPSim TCL application that can, among other things, perform the following tasks:

- Loading picocode
- Creating and managing socket interfaces that integrate control point software in the simulation environment and receive egress data from the simulation model
- Providing ingress traffic generation capabilities either within NPSim or through an external traffic generator attached with an Ethernet connection
- Tracing and logging of control frame (guided cell) traffic and ingress and egress network data traffic
- Recording or playing back captured control frames or network traffic
- Providing TCL procedures that can be triggered by events such as:
  - receipt of Ethernet frames
  - event notifications from the simulation model
  - accesses to certain memory locations

Developers can easily modify or add to the TCL source code to support specific development needs.

The sample application also provides picocode debugging facilities, using the current version of NPScope, which supports multiple blades (units, usually cards, that contain one NP4GS3).

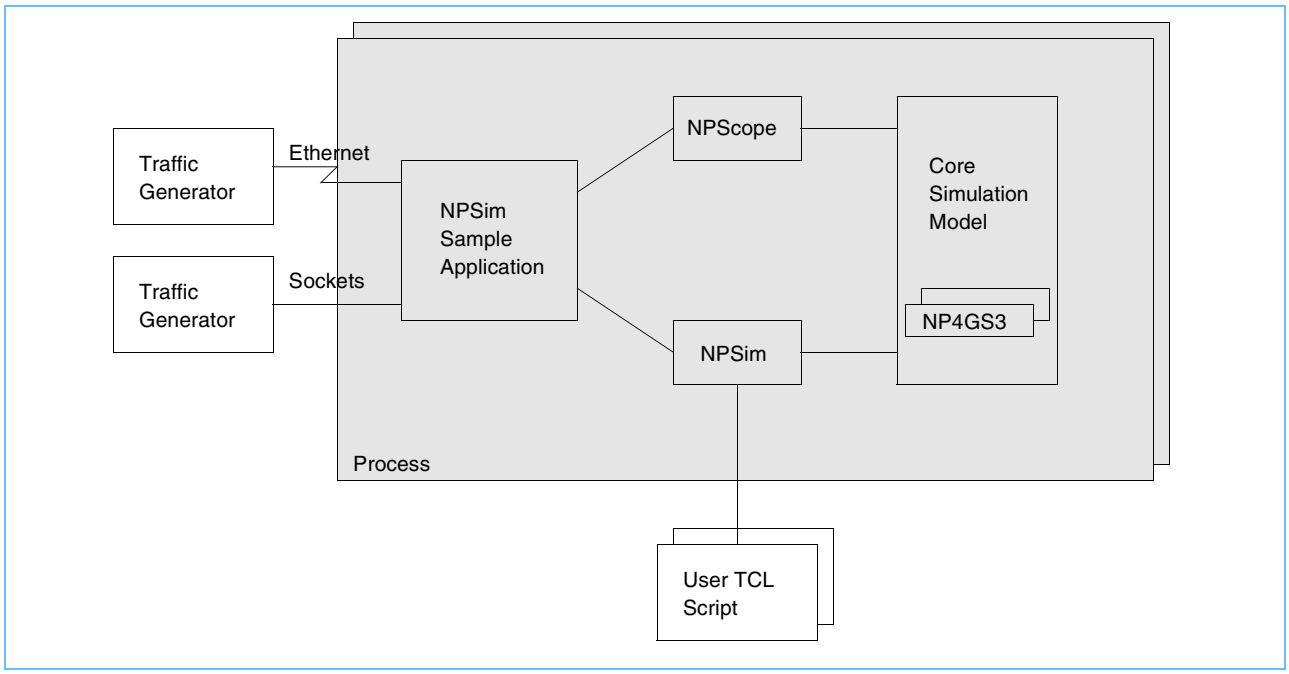
The NPSim sample application can be used to create test network configurations of various sizes and complexity, such as:

- A single NPSim instance configured for one or more blades. (When there are multiple blades, the blades are connected through a simulated switch fabric.)
- Multiple NPSim instances, either in the same host computer or in a different ones. (Communication between NPSim instances occurs over sockets.)

One or more control points using sockets can control each blade. The control points can be in the same host computer as NPSim or in a different one.

Figure 3 illustrates NPSim and its related components.

Figure 3. NPSim



For detailed information on NPSim, see the *IBM PowerNP NP4GS3 Software Developer's Toolkit User's Guide*. Contact your IBM sales representative for information on obtaining this document.

NPProfile

The NPProfile tool analyzes picocode performance. It receives event information from NPScope, derives picocode execution performance data, and displays the data in graphs. This information can answer performance-related questions, such as:

- How many cycles does it take to process frames?
- If a cycle budget isn't being met, how are the cycles being spent?
- Which instructions are executed?
- How often are instructions not executed immediately, and how long is the wait?

For detailed information on NPProfile, see the *IBM PowerNP NP4GS3 Software Developer's Toolkit User's Guide*. Contact your IBM sales representative for information on obtaining this document.

NPTest

NPTest is a test-case generator. It uses a TCL/Tk scripting interface to provide functions for creating test cases, which in turn are used to set up memory and issue hardware configuration commands. These test cases provide a controlled environment for executing picocode, and they minimize dependencies on software normally needed to execute functions being tested. NPScope can execute the test cases generated by NPTest.

NPTest removes the need to force certain code paths using external events. Test scenarios can be contrived for handling both normal paths and errors. And they can be run repetitively, either on the same or different versions of code, to ensure consistent test execution.

Some other NPTest functions include:

- Configuring tree search engine memory and allocating tree search engine memory blocks
- Creating trees
- Configuring ports
- Creating control or data packets
- Configuring the NP4GS3's hardware classifier
- Allocating and initializing counters

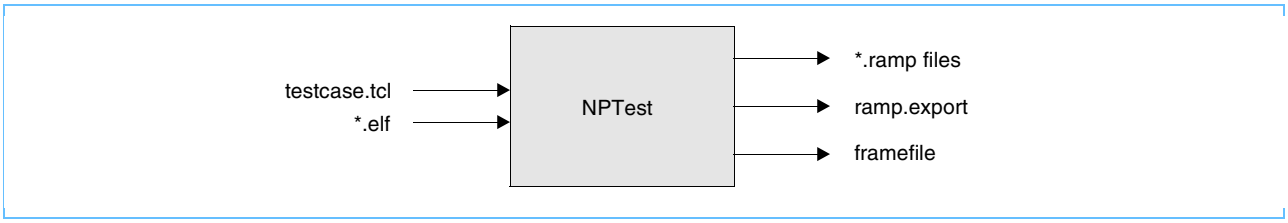
Figure 4 depicts NPTest processing. The inputs to NPTest are:

- *testcase*.tcl (where testcase is the name of your particular testcase): TCL commands for executing NPTest functions.
- *.elf: static memory that NPTest automatically allocates.

After NPTest processes the input, it generates the following files:

- ramp.export: test case loaded into NPScope.
- *.ramp files: referenced by ramp.export and used by NPScope.
- framefile: referenced by ramp.export if packets are created and used by NPScope

Figure 4. NPTest



For detailed information on NPTest, see the *IBM PowerNP NP4GS3 Test Case Generator Programmer's Guide,* the *IBM PowerNP NP4GS3Test Case Generator Command Reference,* and the *IBM PowerNP NP4GS3 Generating and Running the NP4GS3 Demonstration Test Cases for NPScope* manual. Contact your IBM sales representative for information on obtaining these documents.

RISCWatch

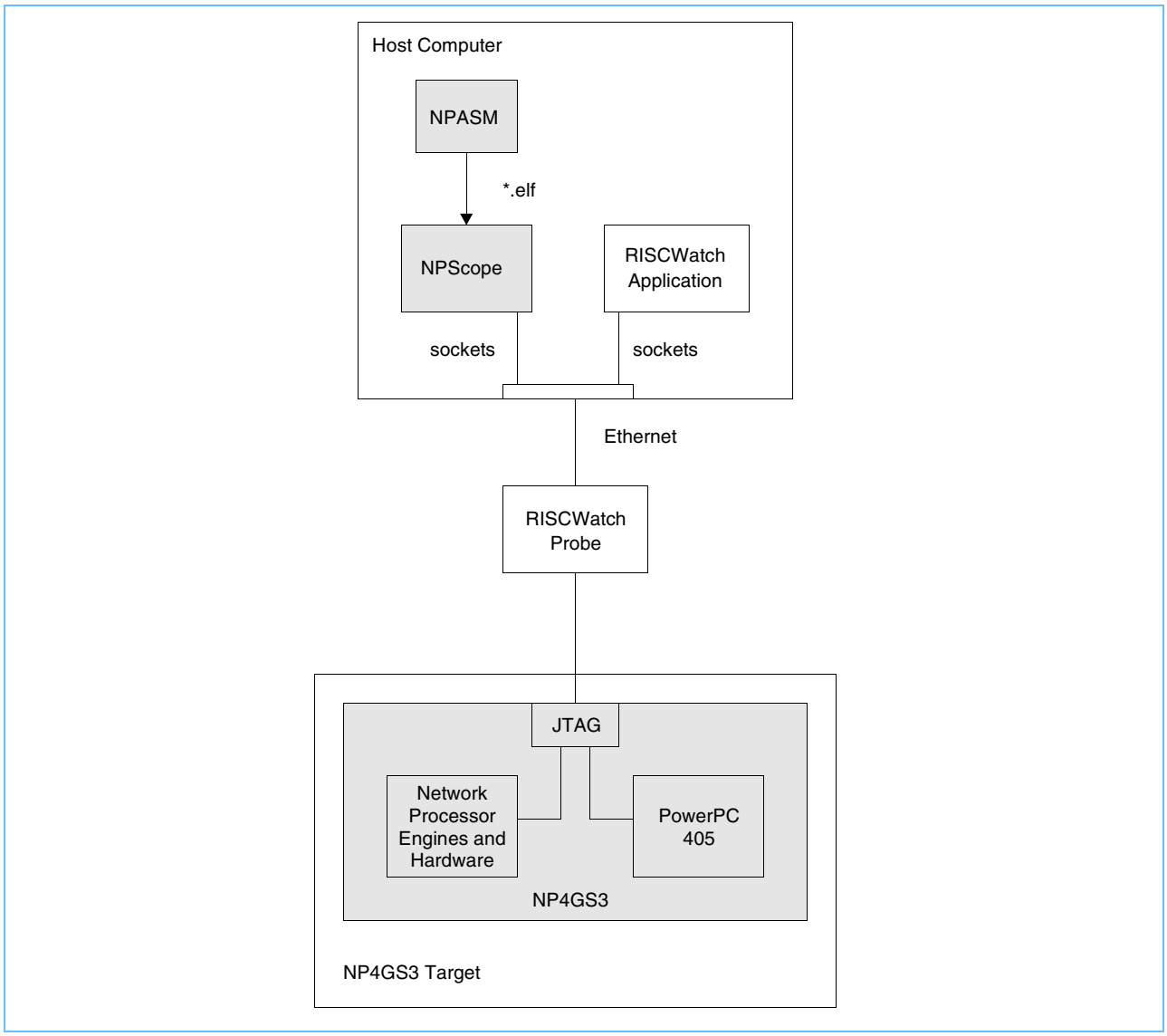
The RISCWatch software application and the RISCWatch probe are provided separately and are not part of the Software Developer's Toolkit.

The RISCWatch probe connects a host computer to the NP4GS3's JTAG pins through Ethernet connections. The same RISCWatch probe can simultaneously debug both the NP4GS3, with NPScope, and its embedded IBM PowerPC 405 processor, with the RISCWatch application.

During hardware bringup, you can use NPScope to access NP4GS3 registers, memory, and so on, and RISCWatch software to debug the embedded PowerPC 405. For both the network processor and the embedded PowerPC 405, you need a RISCWatch probe.

Figure 5 illustrates a sample configuration used during hardware bringup and verification.

Figure 5. Hardware Bringup Configuration



Note: The application included with the RISCWatch probe does not run under Linux, but NPScope on Linux communicates with the probe.

RISCWatch provides developers with a wealth of advanced debugging capabilities and features, including:

- On-chip debugging via an IEEE 1149.1 (JTAG) interface
- Target monitor debugging
- Open-source real-time operating system-aware debugging
- Source-level and assembler debugging of C/C++ executables
- Real-time trace support using the RISCTrace feature for the PowerPC 400 Series
- Network support for remote debugging of a system under development
- Support for industry-standard Embedded ABI for PowerPC and XCOFF ABI
- Command-file support for automated test and command sequences
- Ethernet-to-target JTAG interface hardware
- Support for multiple hosts
- Intuitive and easy-to-use windowed user interface

For more information on RISCWatch, contact your IBM sales representative.

Using the Software Offerings, the Toolkit, and RISCWatch

Software Unit Test

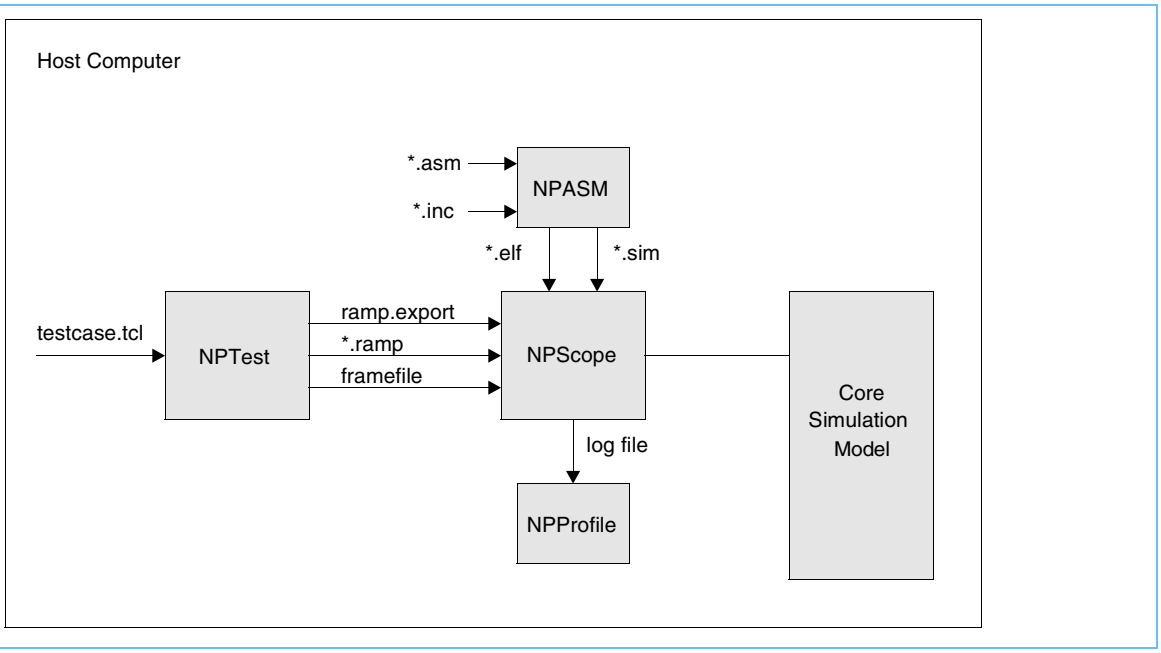
Use the core simulation model to unit-test picocode. The hardware might not yet exist or might not yet be available to execute the picocode. Even if hardware is unavailable, executing the picocode on the model can still reduce the testing environment's complexity and the number of variables. It's easier to test new picocode on a working simulation model than on new hardware.

Use NPTest to create and initialize data structures for the code being tested. NPTest can also create packets that the simulator can process to exercise the code under test.

NPScope has an instruction counter for tracking execution path lengths. When timing is enabled, NPScope updates the cycle counter in the status bar to indicate the number of cycles executed. NPScope, when configured, also includes the timing information in a log file. With the information in this file, NPProfile can be used to locate bottlenecks and determine how long they last. This helps in ordering the instructions for optimal performance.

Figure 6 shows a picocode unit test setup that executes on a single workstation. The figure does not show packet-generation mechanisms other than those provided by NPTest.

Figure 6. Picocode Unit Test Configuration



The control point platform determines how the control point software unit testing is configured. For example, if the control point runs under Linux, the control point software can be unit-tested on a standard personal computer or available hardware such as the IBM PowerNP NP4GS3 Reference Platform. If the control point runs under Wind River Systems VxWorks, an evaluation kit, such as the NP4GS3 Reference Platform, is required to execute control point software. The VxWorks development environment requires a license.

Figures 7, 8, and 9 show some examples of configurations for unit testing.

Figure 7 depicts a system in which the control point software being tested and the Software Developer's Toolkit run on different host computers. This configuration is necessary when, for example, the control point software uses an operating system not supported by the Software Developer's Toolkit.

Figure 7. Control Point Unit Test Configuration Example 1: CP and Toolkit on Different Host Computers

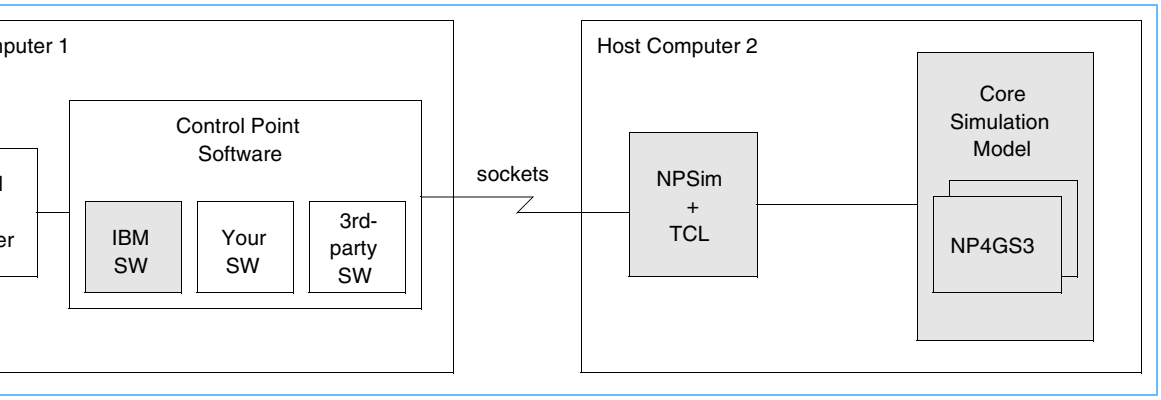


Figure 8 shows a system in which the control point software being tested and the Software Developer's Toolkit run on the same host computer (in this case, under the Linux operating system).

Although IBM offers an integrated Linux development environment using the Advanced Software Offering and the Software Developer's Toolkit, it does not provide the control point debugger. A debugger might be included with the Linux operating system.

Figure 8. Control Point Unit Test Configuration Example 2: CP and Toolkit on the Same Host Computer

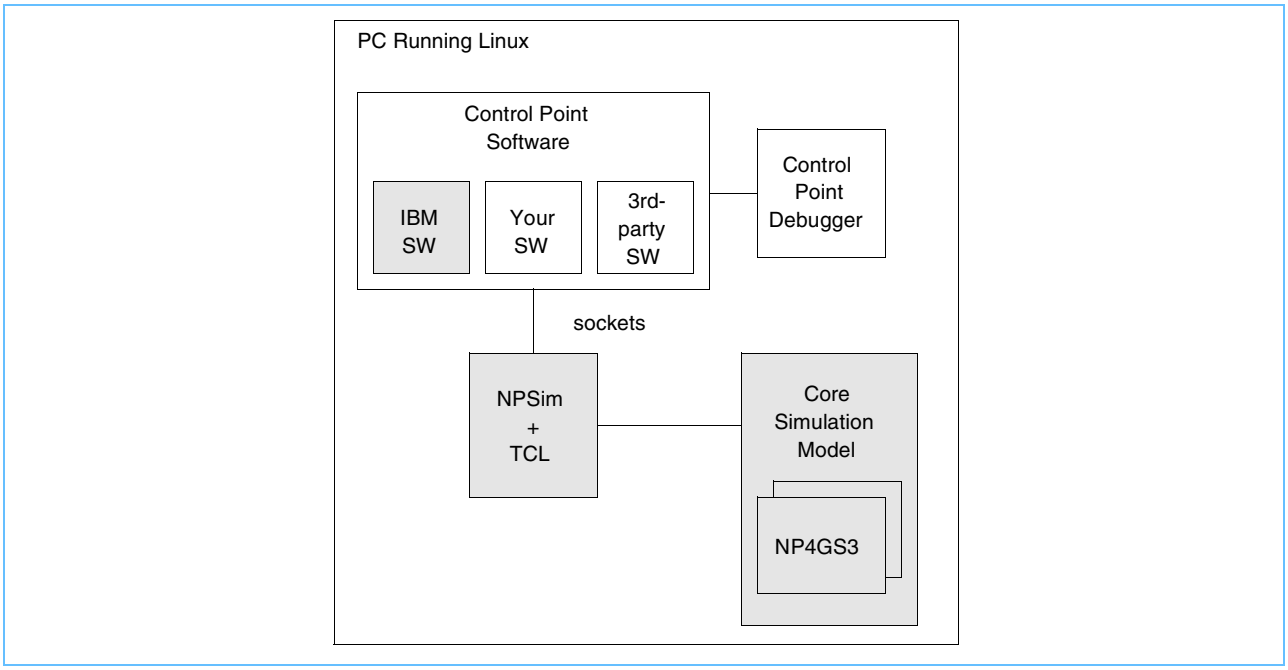
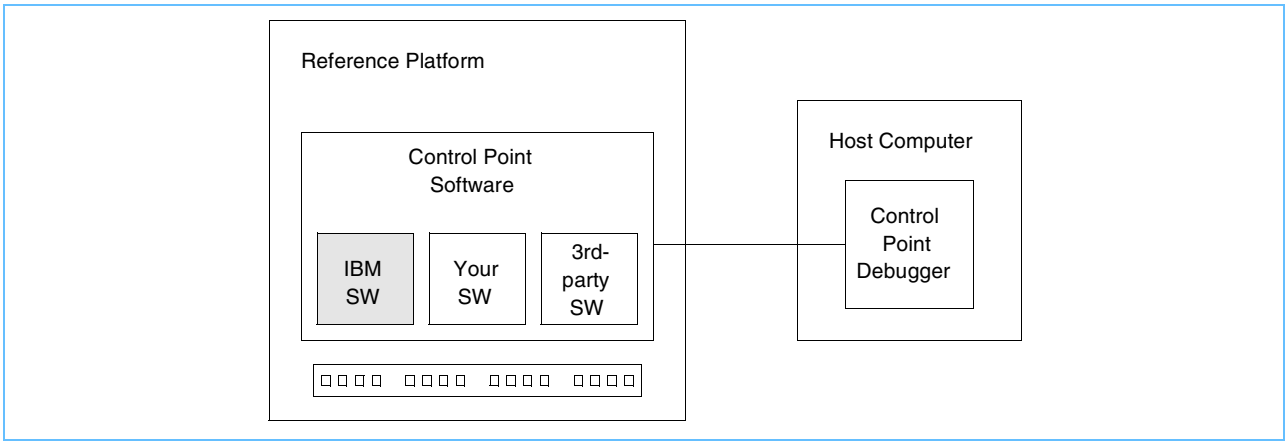


Figure 9 illustrates a system in which the control point software executes on reference hardware, such as the NP4GS3 Reference Platform. This configuration works with control point software running on the Linux, VxWorks, and similar operating systems. The Reference Platform in this figure is a logical subset of the NP4GS3 Reference Platform.

Figure 9. Control Point Unit Test Configuration Example 3: CP on Reference Hardware



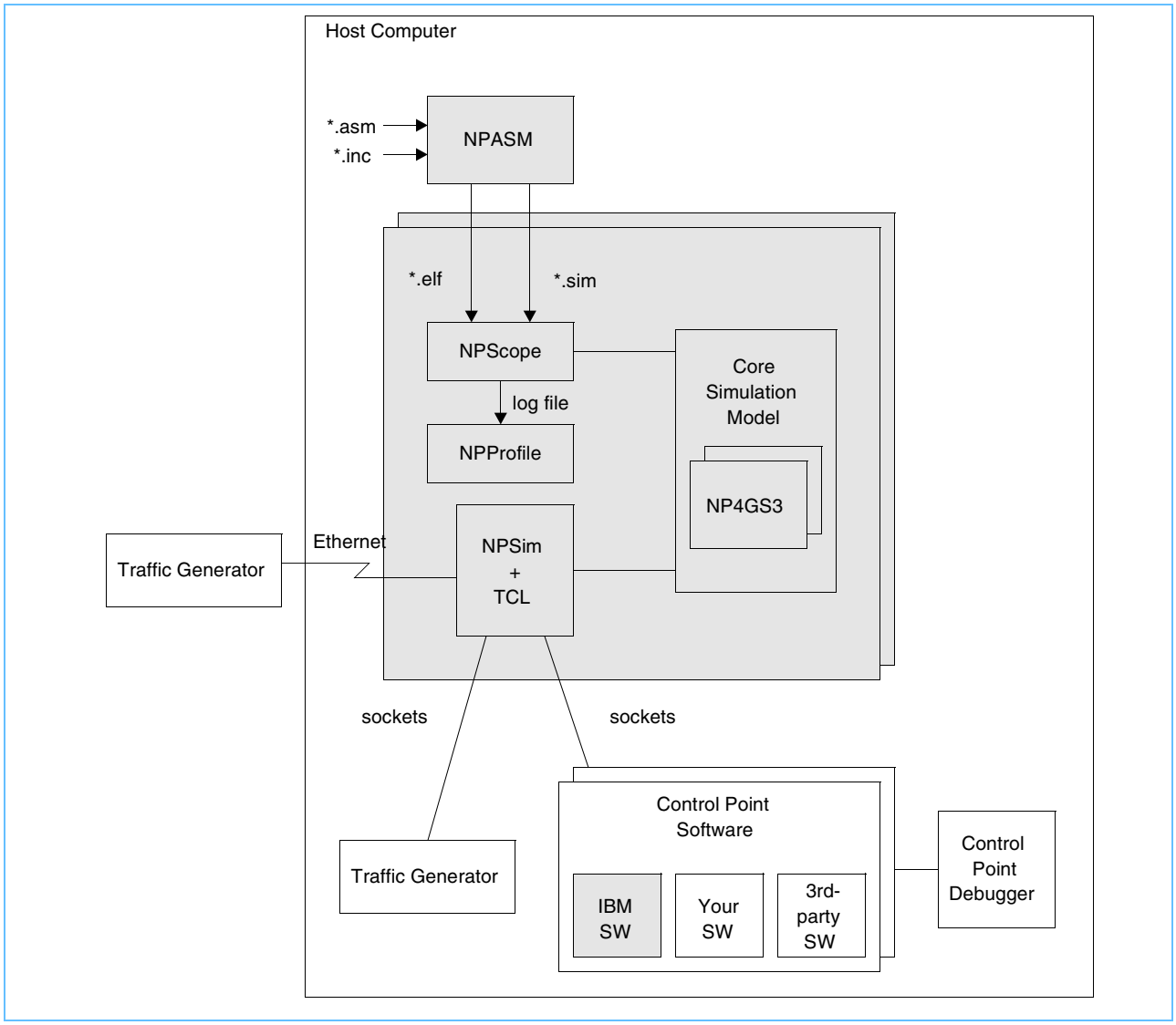
Software Function / System Test

NPSim and its sample application contain the functions required to integrate and test all software components (control point and network processor picocode). Using a socket interface, NPSim can forward control traffic between a Linux reference control point software and the core simulation model.

The control point software can be on the same host computer as the Software Developer's Toolkit. This makes extensive software testing convenient, whether at the developer's desk or anywhere there's access to a host computer containing the software.

Figure 10 shows an example of a configuration for software function test. The control point software being tested doesn't have to be on the computer running the Software Developer's Toolkit. This configuration would be used when, for example, the control point software uses an operating system not supported by the Software Developer's Toolkit.

Figure 10. Software Function / System Test Configuration



A complete software function test can be run without using hardware by configuring NPSim to use multiple blades, connecting multiple NPSim instances together using sockets, or both. Each NPSim can interface with one or more control points. A system with multiple blades can simulate a chassis containing multiple NP4GS3s connected through a switch fabric.

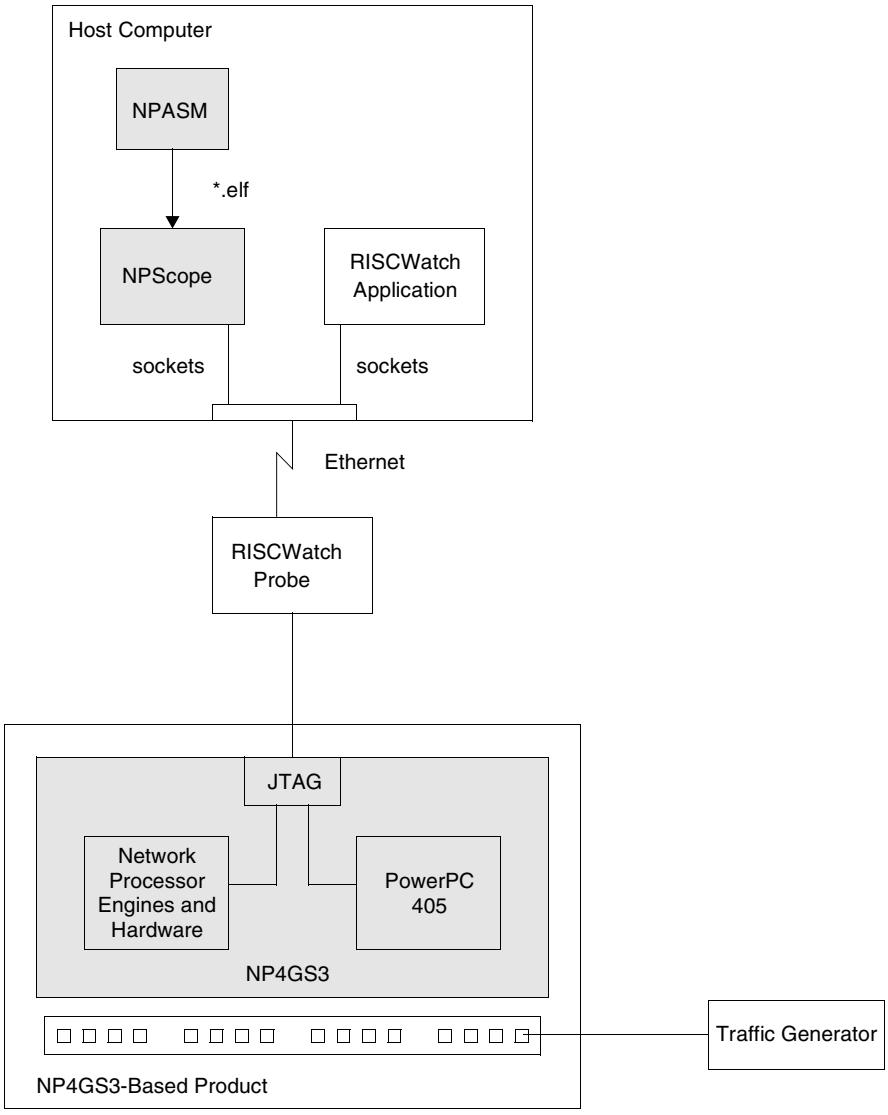
Hardware and Software Integration

Loading code onto a target network processor using NPScope requires the executable image created by NPASM. Use NPScope to debug the target NP4GS3; use the RISCWatch application software to debug the embedded PowerPC.

While the RISCWatch probe needs to be located near the target product, NPScope and the RISCWatch application software can run on a different workstation. For example, the target can be in a lab and a developer can debug the hardware from an office.

Figure 11 on page 15 illustrates a test configuration used during hardware and software integration.

Figure 11. Hardware and Software Integration Test Configuration

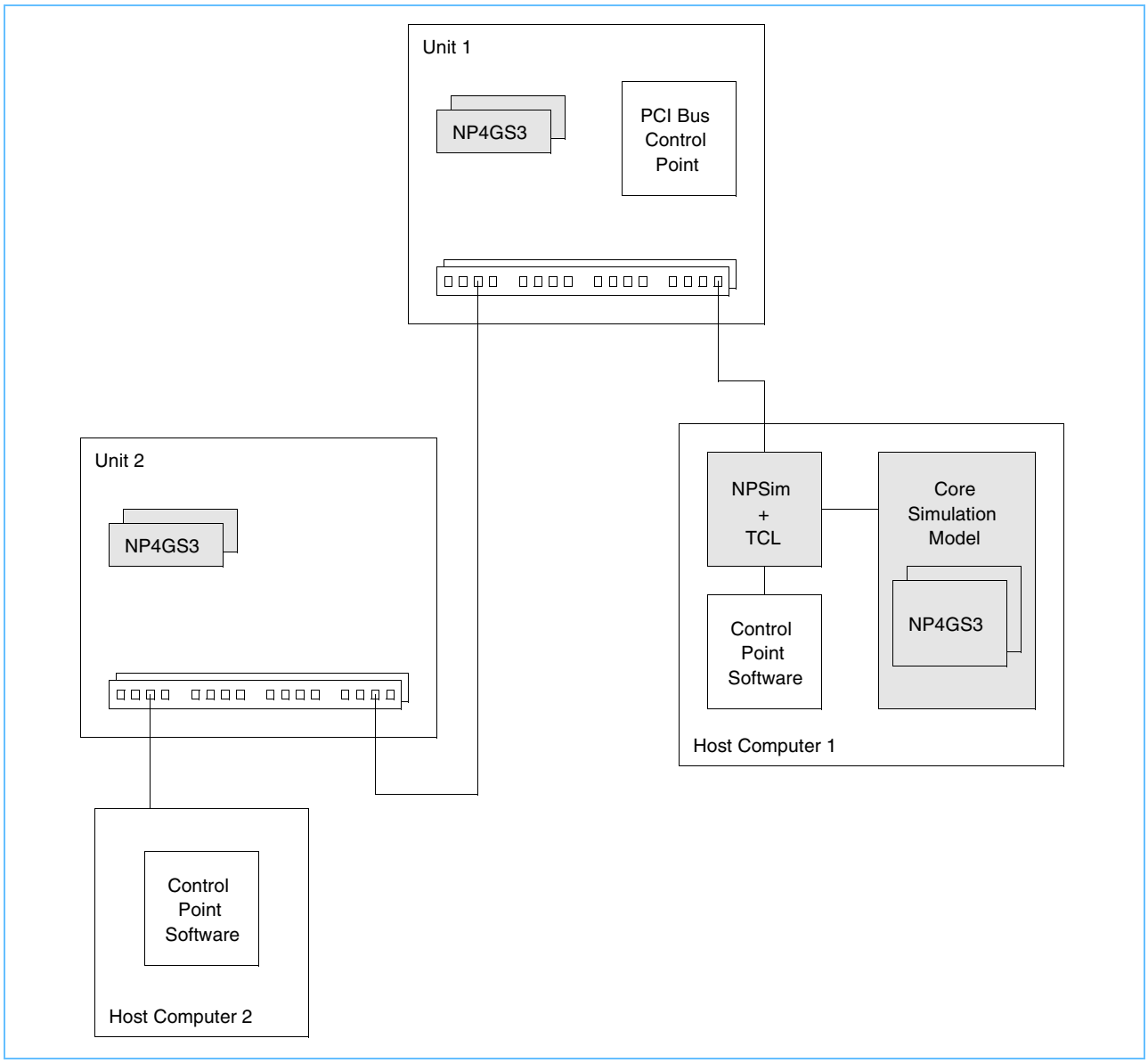


Product System Test

Normally, a system test is performed with only the production-level hardware and software in the configuration. However, sometimes complex configurations are needed and there isn't sufficient hardware to build the test configuration. Some may prefer to use all real hardware in the test, but NPSim can be included in the environment to expand the test configuration.

Figure 12 on page 16 shows a sample system test configuration. The hardware shown in the figure is a logical representation of a NP4GS3-based end product, such as a switch or router. Unit 1 is a networking device in which the control point is on a processor attached to the PCI bus. Unit 2 is a networking device in which the control point is in Host Computer 2, which is external to the device. Host Computer 1 contains NPSim and control point software and is included in the configuration to create a larger network.

Figure 12. System Test Configuration



Product System Test

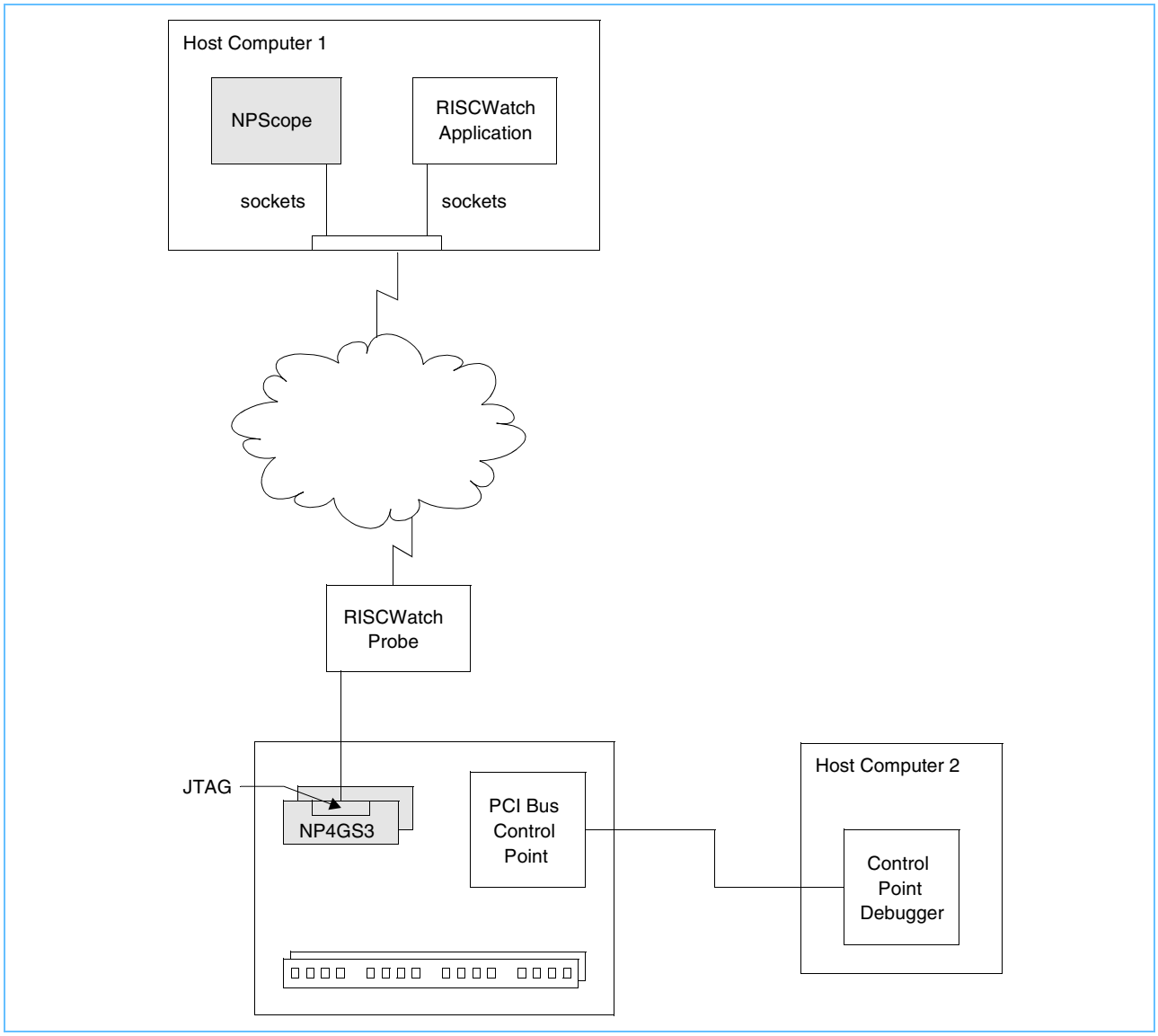
To help validate the connections between the NP4GS3 and the board where the NP4GS3 resides, IBM provides the Boundary Scan Description Language file necessary to perform the JTAG boundary scan during manufacturing tests.

Field Debug

In some situations, problems at a customer site cannot be reproduced in the development lab. In this case, you need debugging facilities in the field. A configuration similar to the one used during hardware and software integration can debug hardware or software in the field.

Figure 13 depicts such a field debug configuration.

Figure 13. Field Debug Configuration



PowerNP NP4GS3 Services

IBM's experienced networking professionals can help you produce a high-quality, value-added product using the PowerNP NP4GS3. They can support all phases of a development cycle, and are highly experienced in developing everything from simple to complex systems in various environments.

Some of the services that IBM offers include:

- System design (overall product system design, hardware design, or software design), which can include design review, design creation, and advice on design creation
- Architecture adaptations, illustrating best uses for the network processor in your existing architecture

- Focus sessions for analyzing applications, understanding requirements, and mapping requirements into a hardware platform and software architecture using a PowerNP network processor
- Performance analysis of your design or implementation, identifying potential problems and suggesting improvements
- Education on PowerNP hardware and software
- Software development, including:
  - Providing sample software
  - Customizing software that meets your requirements
  - Porting or integrating software with your software or third-party software
- Implementation support

Design

The IBM PowerNP NP4GS3 Reference Platform and software can provide a head start in designing a product, or it can be used as a basis for value-added services and features.

IBM also offers design services from networking experts, who can help you make architecture and design decisions, evaluate design correctness, assess performance, and perform similar activities.

IBM provides documentation for product system, hardware, and software design on the NP4GS3, its components, the reference platform, and all of the associated software and tools. Overview information is available at http://www-3.ibm.com/chips/techlib. Some documents, such as those listed in the following paragraphs, require a password for access. Contact your IBM sales representative for more information.

The *IBM PowerNP NP4GS3 Network Processor Datasheet* provides technical information for the NP4GS3 hardware reference design and examples to help create logic schematics. It also contains the information needed to compare analysis results against NP4GS3 requirements.

The *PC Card Layout Guidelines* provide information for component placement and wiring.

The *I/O Buffer Information Specification* model gives timing information for use during timing analysis.

Conclusion

IBM has the hardware, software, tools, and services to help customers create a high-quality PowerNP NP4GS3-based product and enable quick time-to-market. It provides the framework for easily including value-added functions to increase time *in* market. The breadth and depth of IBM's offerings can enable customers to exploit the full potential of their PowerNP network processors and provide system-level solutions for building network processor-based products.

References

Documentation for the NP4GS3 and all of the products and services mentioned in this overview is located in IBM's Technical Library at http://www-3.ibm.com/chips/techlib. The *Networking Technology* section contains product overviews, datasheets, application notes, technical notes, white papers, and other information.

While most product overviews and briefs are available for public viewing, many of the other documents are located on secure Web sites. Contact your IBM sales representative for information on obtaining these documents.

Some of the publications available include:

Hardware/Software Overviews:

IBM PowerNP NP4GS3 Network Processor Solutions Overview
IBM Power Network Processor Software Overview
IBM PowerNP NP4GS3 Advanced Software Offering User's Guide
IBM NP4GS3 Base Software Offering Overview
IBM PowerNP NP4GS3 Reference Platform Overview

Hardware:

IBM PowerNP NP4GS3 Network Processor Datasheet
IBM PowerNP NP4GS3 Network Processor Hardware Reference Manual

Software:

IBM PowerNP NP4GS3 Advanced Software Offering Overview
IBM PowerNP NP4GS3 Assembler Language Programmer's Guide and Instruction Summary
IBM PowerNP NP4GS3 Generating and Running the NP4GS3 Demonstration Test Cases for NPScope
IBM PowerNP NP4GS3 Reference Platform Developer's Code Kit
IBM PowerNP NP4GS3 Reference Software for Linux User's Guide
IBM PowerNP NP4GS3 Software Developer's Toolkit User's Guide
IBM PowerNP NP4GS3 Software Overview
IBM PowerNP NP4GS3 Test Case Generator Command Reference
IBM PowerNP NP4GS3 Test Case Generator Programmer's Guide
RISCWatch Debugger User's Guide

Revision Log

Rev	Contents of Modification
April 2, 2001	Initial release (00).
April 6, 2001	Corrections to initial release - reinserted cross references to other documents - minor technical adjustments
April 13, 2001	Corrected publication date

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6351

The IBM home page can be found at
http://www.ibm.com

The IBM Microelectronics Division home page
can be found at http://www-3.ibm.com/chips

np_overview.fm.00
April 16 2001