Preprint
UCRL-JC-144409

Increasing Inspectability of Hardware and Software for Arms Control and Nonproliferation Regimes

G. K. White

This article was submitted to
42nd Annual Meeting of the Institute of Nuclear Materials Management, Indian Wells, California, July 18, 2001

U.S. Department of Energy
Lawrence Livermore National Laboratory

July 18, 2001

Increasing Inspectability of Hardware and Software for Arms Control and Nonproliferation Regimes*

Gregory K. White
Lawrence Livermore National Laboratory
Livermore, California, 94550, USA (925) 423-6732

ABSTRACT

As the U.S. and the Russian Federation get closer to deploying systems for monitoring nuclear material within arms control and nonproliferation transparency regimes, the level of inspectability of the system hardware and software must increase beyond the systems demonstrated to date. These systems include the Trilateral Initiative prototype,[1] the Fissile Material Transparency Technology Demonstration (FMTTD) system,[2] and the Trusted Radiation Attribute Demonstration System (TRADS).[3] Toward this goal, several alternative technologies will be discussed along with ways in which they would increase inspectability. Some examples of such technologies include the use of microcontrollers instead of fully capable computers, open source operating systems, runtime environments, and compilers.

INTRODUCTION

As we progress toward the actual deployment of monitoring systems for nuclear material, two important goals must be observed: protection of the host country's sensitive information and the assurance to the monitoring party that the nuclear material is what the host country has declared it to be. These goals are met by certification in the host country and authentication in the monitoring party.

During both certification and authentication, each party will need to understand all of the operating parameters of all hardware and software in the deployed system. The goal of both the Trilateral Initiative and the FMTTD System was to measure U.S. nuclear material in the presence of Russian Scientists while protecting information about the material being measured. Demonstrated systems at the Trilateral Initiative and FMTTD went a long way toward providing the openness necessary for a deployed system. The TRADS system uses two Pentium-class computers separated by a controlled interface to manage sensitive information. It has been tested extensively during a rigorous campaign of measurements in a realistic laboratory setting.

A "perfect" system has the following attributes:

1. The software and hardware are only as complex as they need to be to complete the task.
2. The hardware and software deployed on the system are available in source forms to both countries. The author believes that source form is well-understood for software, but it is less clear for hardware. The source form includes both a high-level definition of the hardware and a gate-level definition of the hardware. This source form should include:
    - All software components deployed on the system
        - Basic Input Output System (BIOS)
        - Operating system or runtime environment
        - Data acquisition and analysis code

* This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

- All hardware components deployed on the system
  - Hardware schematics of PC boards
  - Integrated circuit (IC) schematics and gate layouts

3. The tools used to create the hardware and software are available in source form to both countries. This should include compilers, assemblers, and IC design tools.

Although this "perfect" system is probably unattainable, a system that is closer to these attributes is better than one that is farther from them. One of the tenets shared by both authentication and certification is that simpler is better. For instance, if everything else is equal, a system based on Microsoft MSDOS is more desirable than a system based on the Windows NT operating system. Similarly, an Intel 80386 microprocessor is more desirable than a Pentium 4 microprocessor.

Furthermore, at some level in the system, both sides need to rely on functional testing to assure themselves that each hardware or software component performs as specified. Neither side is likely, for example, to create an RS-232 line driver in custom hardware; instead, each would tend to rely on mass-produced products for easier acceptance.

In this paper, I discuss one software and one hardware solution, which if implemented, more closely approach that "perfect" system.

SOFTWARE SOLUTION

Cygnus Software (a division of RedHat) has created a small operating system targeted for embedded system development. eCos[4] (embedded Configurable operating system) is an open source operating system providing many of the attributes we want in a deployable operating system. eCos has been used in Brother laser printers and the Iomega HipZip MP3 Player.

Because it is available in complete open source form and is royalty-free, all parties can fully examine the contents and understand how the operating system works. The development environment for eCos is GNU C under Linux, and the entire development environment is open source. Because Linux is also developed using the GNU C compiler, there is closure with respect to the set of tools necessary to certify or authenticate the product. This is far superior to many other operating environments. For instance, with ROMDOS, (a MSDOS-compatible operating system targeted to embedded developers), source code is available for about $10,000 per party and requires five closed source development tools (assemblers and compilers) to recreate the binary version of the operating system.

The eCos license[5] is flexible enough that it should not constrain either party. Applications written in eCos need only contain a short notice indicating that the application contains eCos. eCos is royalty-free. This includes both the development environment and applications written under the eCos operating system. The only limitation placed on developers is that any change made to eCos must be made available to others on the eCos developer's mailing list.

eCos supports a wide range of microprocessors, allowing each party to choose the microprocessor with the optimal set of features necessary for the project. eCos supports ARM, x86, MIPS, PowerPC, SPARC, and other microprocessors. It also supports over 40 hardware boards as targets. Applications written for one microprocessor under eCos can be quickly ported to another microprocessor. eCos also supports PC-compatible hardware and a synthetic target that runs under Linux. This synthetic target is a software-only environment that allows the developer to debug eCos applications without the normal target hardware. This makes the learning process simpler for new developers and might offer ways to make authentication simpler and faster.

One of eCos's strengths is that it offers developers the ability to easily remove portions of the operating system not needed for a specific application. Over 200 configuration options can be set. This allows developers to minimize the amount of software used in the system, which will aid authentication and certification. The eCos operating system can be paired down to occupy only four kilobytes of memory.

OPEN SOURCE INTEGRATED CIRCUITS

One project that makes it possible to deploy integrated circuits using an open source design model is the LEON embedded processor. The LEON embedded processor was designed for the European Space Agency.[6] It is a SPARC-compatible processor. The SPARC processor, designed by Sun Microsystems, is used in all their workstations. It has been implemented in both standard-cell ASIC and FPGA designs.

SPARC International is a member-funded organization that promotes the SPARC architecture, provides trademark protection, and certifies processors complying with the SPARC standards. Although not certified by SPARC International as a SPARC processor, the LEON processor has been extensively tested by its manufacturer and sponsor for compliance with the SPARC V8 standard. This should prove sufficient for exploratory development.

The VLSI Hardware Definition Language (VHDL) implementation is available under the GNU Lesser General Public License (LPGL).[7] This license allows developers to use the LEON VHDL implementation in proprietary (non-open source) implementations and charge for its use.

The LEON processor includes on-chip many additional features normally found in additional chips in other processors. The additional features include a memory controller, timers, serial ports, and digital inputs and outputs. Figure 1 is a block diagram of the processor internals. This drops the chip count required to implement a system. A simple system board could consist of one LEON processor implemented in an FPGA, one RAM chip, one ROM chip, a couple of RS-232 line driver chips, and some simple logic chips.

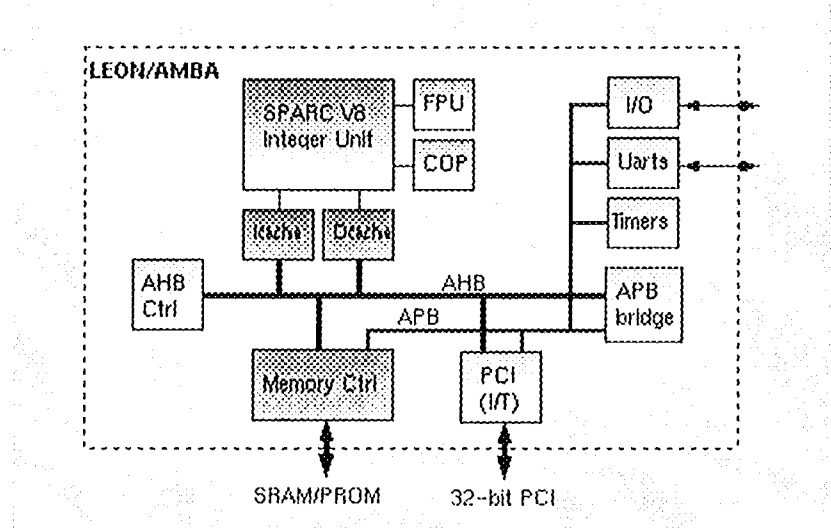

Figure 1: Block diagram of LEON processor internals

The LEON processor has the following features:[6]

- SPARC V8-compatible integer unit with 5-stage pipeline
- Separate, direct-mapped instruction and data caches
- Hardware multiply, divide, and MAC units
- Programmable 8/16/32-bit memory controller for external prom and static ram
- On-chip peripherals such as UARTs, timers, interrupt controller, and 16-bit I/O port
- Full implementation of AMBA-2.0 AHB and APB on-chip buses for expansion
- Interfaces for both floating-point unit and user-defined co-processor
- Power-down mode

In standard-cell ASIC and FPGA designs, the processor is capable of clock rates of 125 MHz and 45 MHz, respectively. In standard-cell ASIC designs, the processor is capable of 1.1 MIPS/MHz. In a standard-cell ASIC design, it occupies approximately 35,000 gates.

In keeping with the open source theme of the LEON project, a complete open source development environment is available, including the GNU C compiler, assemblers, and associated utilities. Although not open source, Gaisler Research has also created the TSIM Simulator for Solaris, Linux and Windows platforms.[8] This product simulates, in software, the LEON processor. This allows the developer to debug applications without the need for a hardware implementation of the LEON processor. The standard debugging tools, such as the GNU debugger (GDB), connect directly to the simulator.

CONCLUSION

Inspectability has emerged as a key facet of designing a monitoring system for authentication. In current systems, the computers controlling acquisition and analysis constitute the most complex and least inspectable components. Short of achieving the "perfect" system, much can be done to reduce the amount and complexity of hardware and software, thereby simplifying inspection. The measures proposed here permit the removal of superfluous functionality using open source tools. As such they offer an evolutionary path away from the current practice of simply adapting laboratory-based instruments to this peculiar application. Upcoming systems may achieve significant improvement in inspectability at low cost by pursuing these mature technologies.

REFERENCE

[1] D. G. Langner, T. Gosnell, D. W. MacArthur, N. J. Nicholas, R. Whiteson, and J. Wolford, *Progress Towards Criteria For A Second-Generation Prototype Inspection System With Information Barrier For The Trilateral Initiative,* 41st Annual Meeting of the INMM, New Orleans, 2000

[2] Larry Avens, James E. Doyle, and Mark Mullen, *The Fissile Material Transparency Technology Demonstration*, 42nd Annual Meeting of the INMM, Indian Wells, 2001

[3] Dean J. Mitchell and Keith M. Tolk, *Trusted Radiation Attribute Demonstration System,* 41st Annual Meeting of the INMM, New Orleans, 2000

[4] *eCos Home Page*, http://sources.redhat.com/ecos/

[5] *eCos Software License*, http://sources.redhat.com/ecos/license.html

[6] *Gaisler Research Home Page*, http://www.gaisler.com/

[7] *Lesser GNU Public License*, http://www.gaisler.com/doc/COPYING.LGPL

[8] *TSIM SPARC Simulator Software*, http://www.gaisler.com/tsim.html

University of California
Lawrence Livermore National Laboratory
Technical Information Department
Livermore, CA 94551