A Boot-Strap Loader and Monitor for SPARC LEON2/3/FT

Les Miklosy PE
Software to Spec

The SPARC LEON family of processors offer the developer a configurable architecture for 32-bit embedded development with many options available from configurable open IP cores. For LEON all targets require an external monitor to initialize, command and control the target. Redboot - a component of the eCos real-time operating system as both a boot-strap loader and ROM monitor is ported for the first time for LEON. The Redboot monitor enhances an otherwise conventional bread-board computer into a stand-alone test bench for launching and debugging application software from the user console or debugger.

Configurable IP Cores

The LEON VHDL model developed by ESA/Gaisler Research (www.gaisler.com) conforms to the IEEE-1754 (SPARC V8) architecture and instruction set for a 32-bit processor. It is designed for embedded applications and supports many hardware features on-chip. The VHDL model maybe synthesized with most synthesis tool vendors and implemented on both FPGAs and ASICs. The LEON2 VHDL model was released under GNU_GPL license and used as the basis for commercial developments and the flight processor for early ESA satellites.

The GRLIB IP Library from Gaisler Research is a set of integrated reusable IP (intellectual property) cores designed for system-on-a-chip development. The IP cores are centered around a common high-speed bus and a plug-and-play method is used to connect IP cores without the need to modify global resources. The concept behind GRLIB is to provide a standardized vendor-independent infrastructure to utilize reusable IP cores.

SPARC

SPARC stands for Scalable Processor Architecture. SPARC is an open set of technical specifications users can license (info@sparc.org) to develop new microprocessors and other semiconductor devices based on published IEEE industry standards.

SPARC was pioneered by Sun Microsystems Inc. (now Oracle Corporation) based upon a Reduced Instruction Set Computing (RISC) at the University of California at Berkeley. The SPARC instruction set is an open industry standard, IEEE Standard 1754-1994.

The SPARC architecture is scalable in both price and function, SPARC licensees have full access to the core instruction set, allowing them to create their own processor chip version.

LEON Processors

The LEON series 2, 3, 3FT consist of 32-bit processors based on the SPARC V8 architecture featuring radiation hardened components and multi-processor support.

The LEON2 processor was developed under contract from the European Space Agency, and is now available as radiation-hardened component from Atmel. The LEON3 is based on a SPARC V8 architecture core with multi-processor support. LEON3FT is a fault tolerant version suitable for radiation-tolerant FPGA's from Actel and Xilinx. The LEON4 core is new featuring a wider internal bus and Level-2 cache. Evaluation boards for LEON3 and LEON3FT and LEON4 are available from Aeroflex/Gaisler Research.

Template Design of LEON3 IP Cores, GRLIB-1.0.7, Gaisler Research, February 2006.

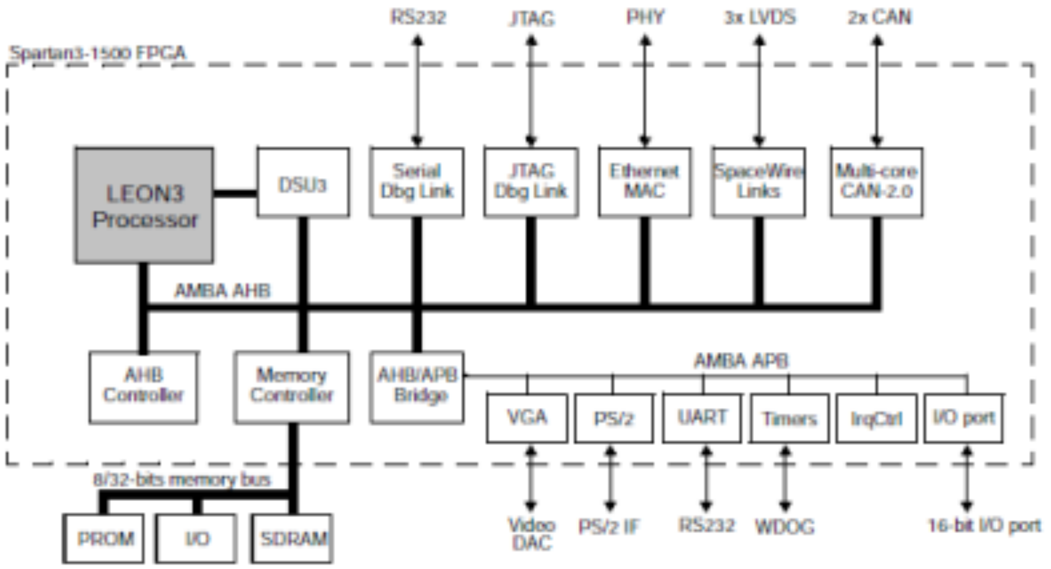


Figure 1. LEON3 template design block diagram

Testing Application Software

Software developers require hardware to execute application software when software simulators won't do or new development hardware is not available during a product development cycle. Atmel Aeroflex/Gaisler Research and Pender Electronics offer inexpensive development boards to substitute for flight hardware in the case of satellite software development around the LEON processor. Often FPGA based, these development boards offer a test platform that allows complete control over the development configuration and the hardware interfaces required by selecting a pre-packaged IP core, or building your own to specific requirements. The IP cores used in this development are available from Gaisler Research with modifications as necessary to supply the functionality for this development.

Conventional LEON Debug

LEON processor implementations on development boards are monitored by a host computer and connected by a physical connection through hardware interfaces provided on the board such as JTAG, Serial, USB, Ethernet, or 802.1 wireless. A console window displays output from the target process on the development board in it's test configuration. A second debug window provides the developer command and control of the target process. Conventionally command functionality is provided by a monitor program running on the host. For all LEON processors a proprietary monitor provided by Gaisler Research has options to configure, bring-up, execute, read memory, display registers and debug the target process. The proprietary monitor is available for purchase, an evaluation version is available for free download.

Boot-strap Dilemma

At power-up the board has no target process running and no knowledge of its hardware interfaces until the board is initialized. An external monitor cannot initialize the board without access through board interfaces first. Either the external monitor must initialize the board or the board must boot-strap itself.

External Monitor

Aeroflex/Gaisler Research provides an external monitor to accomplish both initialization and interface services for a variety of development boards configured for SPARC LEON. The Gaisler Research Grmon2 monitor has many user features including a complete debugger and interface for gdb but is a company proprietary product. A full-featured monitor is available for a fee, an evaluation version is available for free download. An external monitor makes it necessary that host computer and target are both accessible to the user and connected by a short hardware cable during power-up.

Resident Boot-Strap

A desirable alternative would be to configure the development board as a test-bench complete with boot-strap code, a monitor residing in ROM, interface initialization and a debug stub to complete a testbench target. A self-booting testbench was requested by the Data Systems Group of ESA and was the inspiration for this Redboot project. The boot-strap testbench allows load execution and debug of application programs written for eCos or RTEMS, the included debug stub allows start stop stepping and debug of the application program with gdb from a remote host over a network. The external monitor remains a high fidelity debugger for the development board, but where greater anonymity and self-booting is desirable, the testbench has advantages.

eCos Redboot

eCos is a real-time open source operating system intended for embedded systems (ecos.sourceware.org) and supported by the GNU open source development tools (www.gnu.org) under the 2002 modified GNU-GPL License. The license grants developers the right to freely copy and modify the code and distribute applications based on the original. The operating system is configurable for specific application requirements for a small hardware footprint. The eCos development system is supported by a community forum and has a broad membership in both end-user and developer arenas.

Redboot is a configurable start-up program with functionality determined during the build and distributed with eCos. Redboot complements a boot-strap layer written in assembly for initialization of the hardware. The eCos port for SPARC LEON2 was provided by Gaisler Research and Denayer Institute of the Netherlands but no longer supported. The eCos port for SPARC LEON3 was provided by Gaisler Research but no longer supported in the eCos distributions following version 2.0. Any components missing from the original eCos ports for SPARC LEON were written by the author.

Licensing

eCos and Redboot are distributed under the GNU-GPL license for open source. The SPARC Architecture is licensed through Sparc International Inc. The LEON3FT IP core is licensed through Gaisler Research. The source code for GRLIB-LEON3 is available under GNU-GPL license allowing free and unlimited use for research and education.

Application Programs

Redboot provides functionality to launch application programs from the Redboot command line and routes application I/O to a desired port. Since Redboot and the application program were built separately neither one knows the interface details nor address space held by the other. Either Redboot must pass the interface to the application program at run-time, or the two must share an interface in a common address space. eCos accomplishes the latter through a virtual address table that both Redboot and the application share at build time. Redboot provides a debug stub to further allow a remote debugger like gdb to load, start stop and step the application program. eCos or RTEMS applications can be launched from the command line, or from a Redboot boot-script launched at power-up.

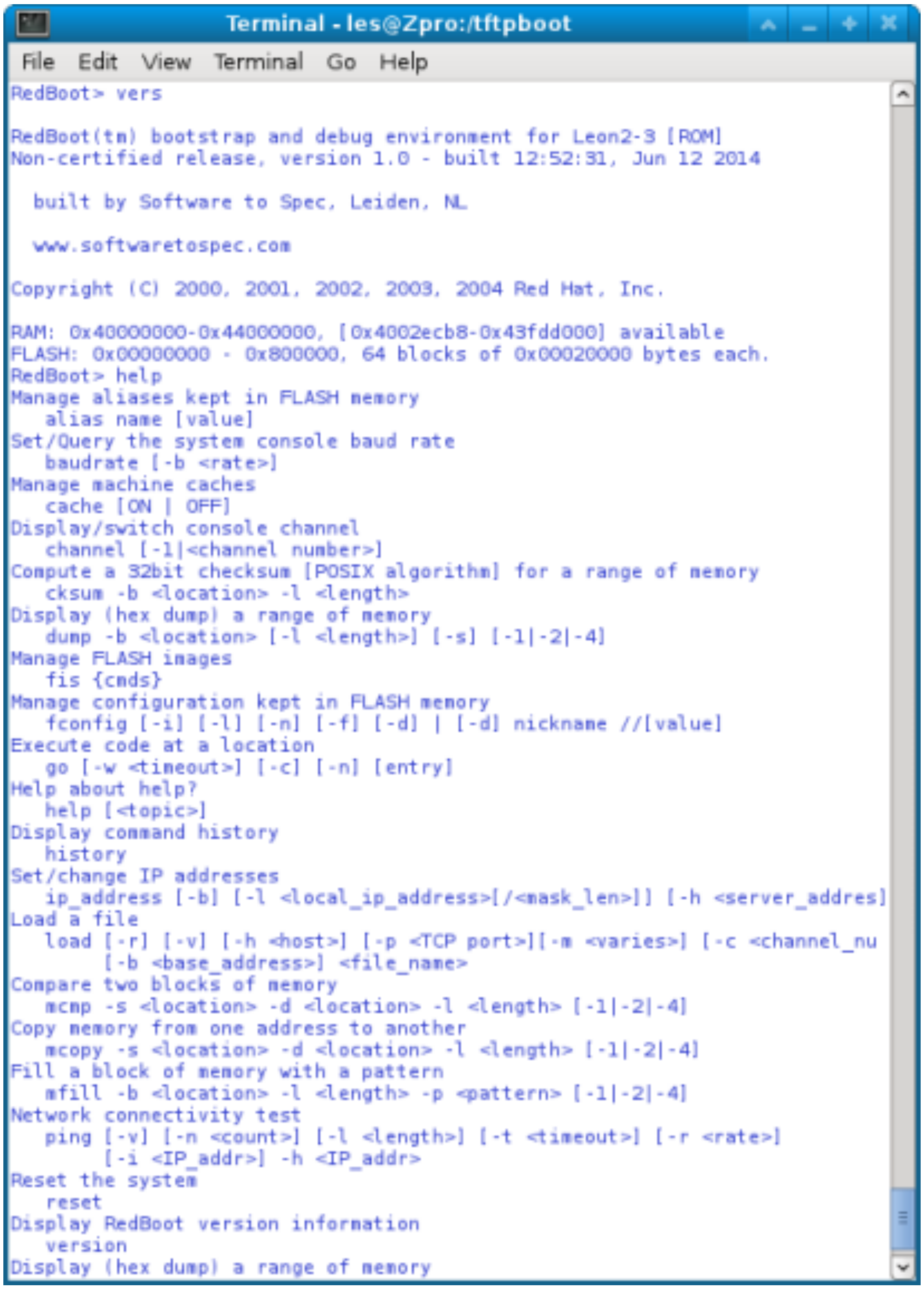
Interfaces

The eCos source code supplies a hardware abstraction layer that supports physical I/O for many development boards including:

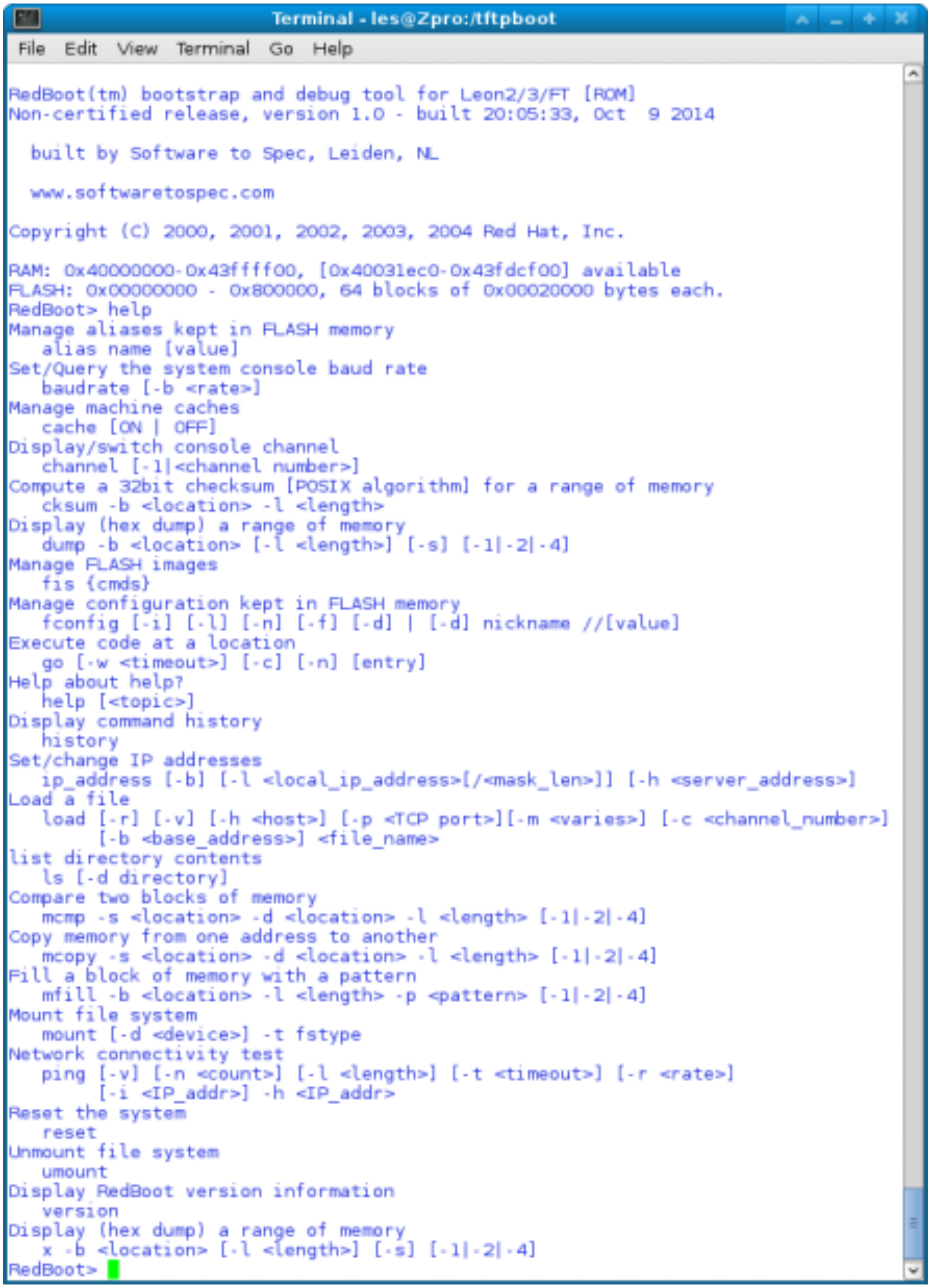
- Serial
- PS2
- Ethernet 10/100T
- JTAG
- Video DAC (VGA)
- USB
- Firewire IEEE 1394
- Spacewire IEEE 1355
- CAN

Redboot Functionality

Redboot provides considerable functionality to configure the testbench and inspect the target memory. The following are screen-shots of the Redboot menu systems for LEON2 and LEON3 boot-strap.


Terminal - les@Zpro:/tftpboot
File Edit View Terminal Go Help

```
RedBoot> vers

RedBoot(tm) bootstrap and debug environment for Leon2-3 [ROM]
Non-certified release, version 1.0 - built 12:52:31, Jun 12 2014

  built by Software to Spec, Leiden, NL

  www.softwaretospec.com

Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x40000000-0x44000000, [0x4002ecb8-0x43fdd000] available
FLASH: 0x00000000 - 0x800000, 64 blocks of 0x00020000 bytes each.
RedBoot> help
Manage aliases kept in FLASH memory
   alias name [value]
Set/Query the system console baud rate
   baudrate [-b <rate>]
Manage machine caches
   cache [ON | OFF]
Display/switch console channel
   channel [-1|<channel number>]
Compute a 32bit checksum [POSIX algorithm] for a range of memory
   cksum -b <location> -l <length>
Display (hex dump) a range of memory
   dump -b <location> [-l <length>] [-s] [-1|-2|-4]
Manage FLASH images
   fis {cmds}
Manage configuration kept in FLASH memory
   fconfig [-i] [-l] [-n] [-f] [-d] | [-d] nickname //[value]
Execute code at a location
   go [-w <timeout>] [-c] [-n] [entry]
Help about help?
   help [<topic>]
Display command history
   history
Set/change IP addresses
   ip_address [-b] [-l <local_ip_address>[/<mask_len>]] [-h <server_addres]
Load a file
   load [-r] [-v] [-h <host>] [-p <TCP port>][-m <varies>] [-c <channel_nu
        [-b <base_address>] <file_name>
Compare two blocks of memory
   mcmp -s <location> -d <location> -l <length> [-1|-2|-4]
Copy memory from one address to another
   mcopy -s <location> -d <location> -l <length> [-1|-2|-4]
Fill a block of memory with a pattern
   mfill -b <location> -l <length> -p <pattern> [-1|-2|-4]
Network connectivity test
   ping [-v] [-n <count>] [-l <length>] [-t <timeout>] [-r <rate>]
        [-i <IP_addr>] -h <IP_addr>
Reset the system
   reset
Display RedBoot version information
   version
Display (hex dump) a range of memory
```


Terminal - les@Zpro:/tftpboot
File Edit View Terminal Go Help

```
RedBoot(tm) bootstrap and debug tool for Leon2/3/FT [ROM]
Non-certified release, version 1.0 - built 20:05:33, Oct  9 2014

  built by Software to Spec, Leiden, NL

  www.softwaretospec.com

Copyright (C) 2000, 2001, 2002, 2003, 2004 Red Hat, Inc.

RAM: 0x40000000-0x43ffff00, [0x40031ec0-0x43fdcf00] available
FLASH: 0x00000000 - 0x800000, 64 blocks of 0x00020000 bytes each.
RedBoot> help
Manage aliases kept in FLASH memory
   alias name [value]
Set/Query the system console baud rate
   baudrate [-b <rate>]
Manage machine caches
   cache [ON | OFF]
Display/switch console channel
   channel [-1|<channel number>]
Compute a 32bit checksum [POSIX algorithm] for a range of memory
   cksum -b <location> -l <length>
Display (hex dump) a range of memory
   dump -b <location> [-l <length>] [-s] [-1|-2|-4]
Manage FLASH images
   fis {cmds}
Manage configuration kept in FLASH memory
   fconfig [-i] [-l] [-n] [-f] [-d] | [-d] nickname //[value]
Execute code at a location
   go [-w <timeout>] [-c] [-n] [entry]
Help about help?
   help [<topic>]
Display command history
   history
Set/change IP addresses
   ip_address [-b] [-l <local_ip_address>[/<mask_len>]] [-h <server_address>]
Load a file
   load [-r] [-v] [-h <host>] [-p <TCP port>][-m <varies>] [-c <channel_number>]
        [-b <base_address>] <file_name>
list directory contents
   ls [-d directory]
Compare two blocks of memory
   mcmp -s <location> -d <location> -l <length> [-1|-2|-4]
Copy memory from one address to another
   mcopy -s <location> -d <location> -l <length> [-1|-2|-4]
Fill a block of memory with a pattern
   mfill -b <location> -l <length> -p <pattern> [-1|-2|-4]
Mount file system
   mount [-d <device>] -t fstype
Network connectivity test
   ping [-v] [-n <count>] [-l <length>] [-t <timeout>] [-r <rate>]
        [-i <IP_addr>] -h <IP_addr>
Reset the system
   reset
Unmount file system
   umount
Display RedBoot version information
   version
Display (hex dump) a range of memory
   x -b <location> [-l <length>] [-s] [-1|-2|-4]
RedBoot>
```

Third Party Contributors

eCos is compatible with third party contributors sharing the open-source GNU_GPL license, some of these capabilities are included in the current build:

POSIX
*ROM Monitors: Redboot, Cygon, GDB Stub
File Systems
PCI Support
*USB Support
*Networking: OpenBSD, FreeBSD, IwIP, Threads, DNS
GoAhead Embedded WebServer
Symmetric Multi-Processing Support

Conclusion

This adaptation of Redboot as a ROM monitor and boot-strap for SPARC LEON shows promise as the first stand-alone monitor for the SPARC LEON family of embedded processors for managing a LEON remote target from a host computer. The SPARC LEON IP cores and the boot-strap are both highly configurable and extensible where the details cannot be covered here. This project remains a development in-progress undergoing verification and performance testing. For further information about SPARC LEON contact the vendors, for information about the LEON boot-strap development contact the author at Software to Spec.