LEON3 ML401 Template Design

Based on GRLIB, November 2007

Jiri Gaisler

Table of contents

1 Introduction ... 3

1.1 Scope ... 3
1.2 Requirements ... 3
1.3 Avnet ML401 board ... 3
1.4 Reference documents ... 4

2 LEON3 template design ... 5

2.1 Overview ... 5
2.2 LEON3 SPARC V8 processor ... 5
2.3 Memory interface ... 6
2.4 AHB status register ... 7
2.5 Timer unit ... 7
2.6 Interrupt controller ... 7
2.7 DSU UART ... 7
2.8 Application UART ... 7
2.9 DSU JTAG interface ... 7
2.10 General purpose I/O port ... 7
2.11 Ethernet ... 7
2.12 VGA controller ... 7
2.13 PS/2 keyboard and mouse interface ... 7
2.14 Clock generator ... 8
2.15 GRLIB IP Cores ... 8
2.16 Interrupts ... 8
2.17 Memory map ... 9
2.18 Signals ... 9

3 Simulation and synthesis ... 10

3.1 Design flow ... 10
3.2 Installation ... 10
3.3 Template design overview ... 10
3.4 Configuration ... 10
3.5 Simulation ... 11
3.6 Synthesis and place&route ... 12
3.7 Board re-programming ... 12

4 Software development ... 13

4.1 Tool chains ... 13
4.2 Downloading software to the target system ... 13
4.3 Flash PROM programming ... 14

1 Introduction

1.1 Scope

This document describes a LEON3 template design customized for the Avnet ML401 Virtex4 FPGA development board. The template design is intended to familiarize users with the LEON3 processor and the GRLIP IP library.

1.2 Requirements

The following hardware and software components are required in order to use and implement the Avnet ML401 LEON3 template design:

- GRLIB IP Library 1.0.17
- PC work station with Linux or Windows 2000/XP with Cygwin
- Avnet ML401 board with JTAG programming cable
- Xilinx ISE 9.2i Development software
- Synplicity Synplify 8.9 or higher (optional).

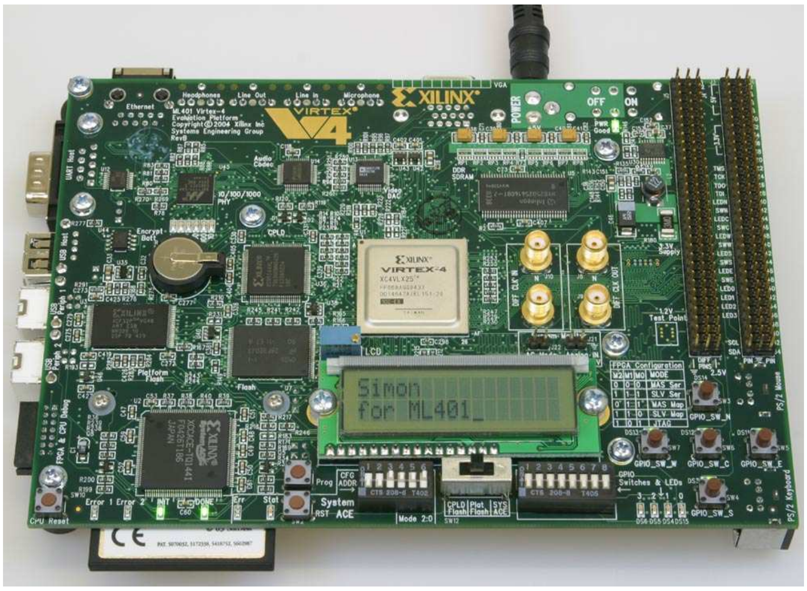
For LEON3 software development, the following tools are recommended

- BCC Bare-C LEON Cross-compiler 1.0.20
- RCC RTEMS ERC32/LEON Cross-compiler system 1.0.10
- GRMON 1.1.7

1.3 Avnet ML401 board

The ML401 board is developed by Avnet, and provides a flexible and low-cost prototype platform for LEON systems. The ML401 board has the following features:

- Xilinx Virtex4 XC4VLX25 FPGA
- 4 Mbyte FLASH PROM (2Mx32) and 1 Mbyte ZBT SSRAM (256Kx32)
- 64 Mbyte DDR RAM (16Mx32)
- One RS-232 interfaces
- USB-2.0 PHY
- 10/100/1000 Mbit/s ethernet PHY
- Two PS/2 interfaces
- VGA video DAC and 15-pin connector
- AC97 audio controller
- I2C serial PROM
- LCD Display
- JTAG interface for programming and debug
- Expansion connectors



Avnet ML401 Development Board

More information about the board can be obtained in the “ML401 Evaluation Platform User’s Guide”, available for download from http://www.xilinx.com/bvdocs/userguides/ug080.pdf .

1.4 Reference documents

The LEON3 template design is based on GRLIB, and uses the GRLIP AMBA plug&play configuration method. The following manuals should therefore be carefully studied in order to understand the design concept:

- GRLIB User’s Manual
- AMBA Specification 2.0
- GRLIB IP Core’s Manual

2 LEON3 template design

2.1 Overview

The LEON3 ML401 template design consists of the LEON3 processor and a set of IP cores connected through the AMBA AHB/APB buses (see figure below).

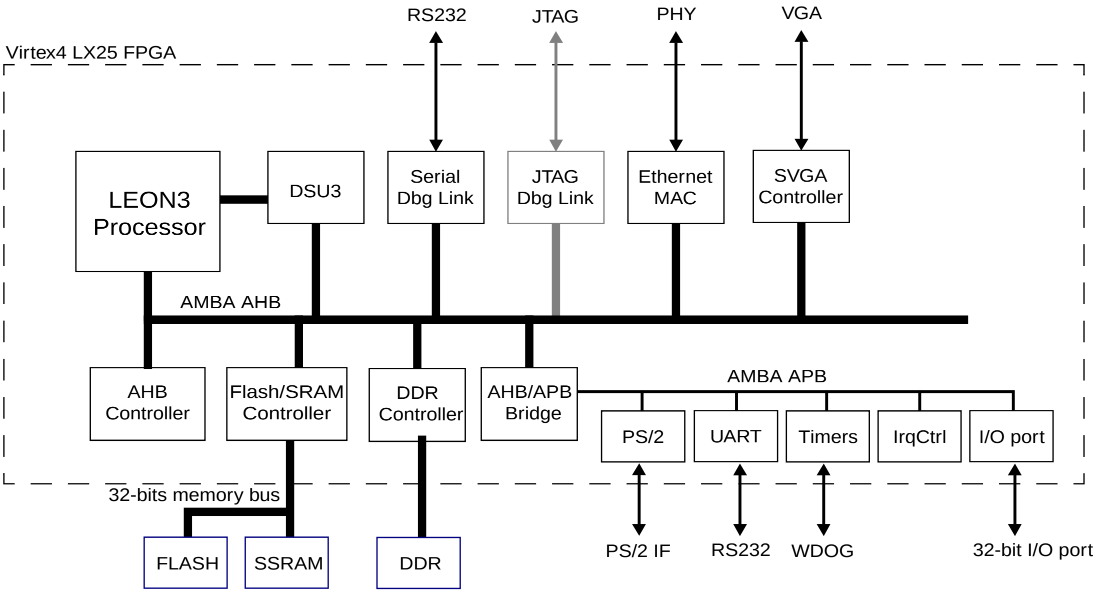


Figure 1. LEON3 template design block diagram

The design is centered around the AMBA Advanced High-Speed bus (AHB), to which the LEON3 processor and other high-bandwidth devices are connected. External FLASH PROM and SSRAM is accessed through a combined PROM/SRAM memory controller. The on-chip peripheral devices include an ethernet 10/100 Mbit MAC, serial/JTAG debug interface, one UARTs, interrupt controller, timers and an I/O port. The design is highly configurable, and the various features can be suppressed if desired.

The design is provided as part of the GRLIB IP Library, in the designs/leon3-avnet-ml401 directory. The full source code is provided under GPL open-source license.

2.2 LEON3 SPARC V8 processor

The template design is based the LEON3 SPARC V8 processor. The processor core can be extensively configured through the xconfig graphical configuration program. In the default configuration, the cache system consists or 16 + 16 Kbyte I/D cache with cache snooping enabled. The LEON3 debug support unit (DSU3) is also enabled by default, allowing downloading and debugging of programs through a serial port or ethernet.

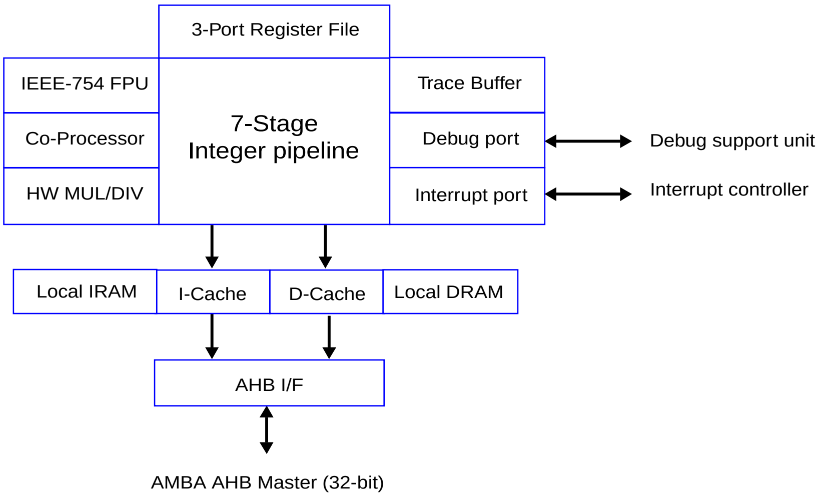


Figure 2. LEON3 processor core block diagram

For more details on the LEON3 processor, see the GRLIB IP Core's Manual.

2.3 Memory interface

The ML401 board has two memory interfaces: one for FLASH PROM and ZBT SSRAM, and one for DDR RAM. The FLASH and SSRAM are interfaced using the combined PROM/SRAM memory controller core (MCTRL). The MCTRL core is design for asynchronous RAM, but can work with ZBT SSRAM when one (1) waitstate is used. The FLASH PROMs are Intel compatible, and flash programming is fully supported through the GRMON debug monitor. The following paramters should be added to GRMON for optimal memory operation:

-ramws 1 -romws 15 -normw

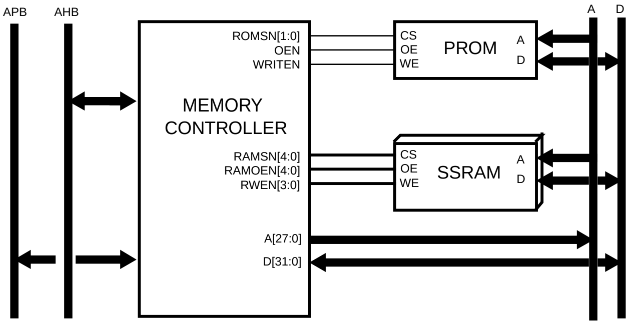


Figure 3. PROM/SSRAM Memory controller

The DDR memory is interfaced using the DDRSPA controller, which can run up to 120 MHz on the ML401 board. When enabled the DDR SDRAM will be mapped at address 0x40000000,

2.4 AHB status register

The AHB status register captures error responses on the AHB bus, and lock the failed address and active master. These values allows the software to recover from error events in the system.

2.5 Timer unit

The timer unit consists of a common scaler and up to 7 individual timers. The timers can work in periodical or on-shot mode. One of the timers can optionally be configured as a watchdog.

2.6 Interrupt controller

The interrupt controller handles up to 15 interrupts in two priority levels. The interrupt are automatically assigned and routed to the controller through the use of the GRLIB AMBA plug&play system.

2.7 DSU UART

The DSU UART is used to communicate with GRMON for application downloading and debugging. It uses the ML401 RS232 port and can operate up to 230 Kbaud.

2.8 Application UART

One standard UART can be configured in the design. The UART has configurable FIFO size, and a separate baud rate generator. The ML401 board has only one RS232 port and when the DSU UART is enabled, the nominal UART does not drive the RS232 signals. In this case, the UART is configured in loop-back mode which allows the UART data to be displayed through GRMON, using the -u option.

2.9 DSU JTAG interface

The template design includes an optional JTAG-based debug interface, AHBJTAG. This allows GRMON to communicate with the board using the JTAG interface. NOTE: the Virtex4 LX25-ES devices cannot use the AHBJTAG interface due to bugs in the on-chip JTAG controller.

2.10 General purpose I/O port

A general purpose I/O port (GPIO) is provided in the design. The port size is fixed to 32 bits, and connected the the ML401 GPIO bus.

2.11 Ethernet

The template design includes one 10/100 Mbit Ethernet MAC based on the GRETH core in GRLIB. The MAC can be configured to support the GRMON debug protocol, allowing download rates of up to 50 Mbit/s. In this case, start GRMON with the -eth option.

2.12 VGA controller

A text-based video controller can optionally be enabled. The controller can display a 80x48 character screen on a 640x480 monitor.

2.13 PS/2 keyboard and mouse interface

The ML401 PS/2 keyboard interface is connected to the APBPS2 core in the LEON3 template design. The mouse interface is not used.

2.14 Clock generator

A custom clock generator is used to generate three clocks: the processor clock, the SSRAM clock and the DDR clock. The SSRAM and DDR clock feedback signals are used to synchronize the memory clocks with the internal processor clock. The clock generator can generate an arbitrary frequency by multiplying and dividing the 100 MHz input clock. The clock scaling factor is configurable through the xconfig tool.

2.15 GRLIB IP Cores

The design is based on the following IP cores from the GRLIB IP library:

Table 1. Used IP cores

Core	Function	Vendor	Device
LEON3	LEON3 SPARC V8 32-bit processor	0x01	0x003
DSU3	LEON3 Debug support unit	0x01	0x004
IRQMP	LEON3 Interrupt controller	0x01	0x00D
APBCTRL	AHB/APB Bridge	0x01	0x006
MCTRL	32-bit PROM/SRAM/SDRAM controller	0x04	0x00F
AHBSTAT	AHB failing address register	0x01	0x052
AHBUART	Serial/AHB debug interface	0x01	0x007
AHBJTAG	JTAG/AHB debug interface	0x01	0x01C
APBUART	8-bit UART with FIFO	0x01	0x00C
GPTIMER	Modular timer unit with watchdog	0x01	0x011
GRGPIO	General purpose I/O port	0x01	0x01A
GRSPW	Spacewire link	0x01	0x01F
GRETH	10/100 Mbit/s Ethernet MAC	0x01	0x01D
APBPS2	PS/2 Mouse/Keyboard interface	0x01	0x060
APBVGA	Text-based VGA controller	0x01	0x061

2.16 Interrupts

The following table indicates the interrupt assignment:

Table 2. Interrupt assignment

Core	Interrupt
APBUART	2
APBPS2	5
AHBSTAT	7
GPTIMER	8, 9
GRETH	12

See the manual of the respective core for how and when the interrupts are raised. The interrupts are all forwarded to the LEON3 processor, and controlled by the IRQMP interrupt controller.

2.17 Memory map

The memory map of the AHB bus can be seen below:

Table 3. AHB address range and bus indexes

Core	Address range	Bus Index
MCTRL	0x00000000 - 0x20000000 : FLASH PROM area 0x20000000 - 0x40000000 : I/O area (unsued) 0x40000000 - 0x80000000 : SSRAM area	0
APBCTRL	0x80000000 - 0x81000000 : APB bridge	1
DSU3	0x90000000 - 0xA0000000 : Registers	2
AHB plug&play	0xFFFFF000 - 0xFFFFFFFF : Registers	-

Access to addresses outside the ranges described above will return an AHB error response. The detailed register layout is defined in the manual for each IP core. The control registers of most on-chip peripherals are accessible via the AHB/APB bridge, which is mapped at address 0x80000000:

Table 4. APB address range and bus indexes

Core	Address range	Bus Index
MCTRL	0x80000000 - 0x80000100	0
APBUART	0x80000100 - 0x80000200	1
IRQMP	0x80000200 - 0x80000300	2
GPTIMER	0x80000300 - 0x80000400	3
APBPS2	0x80000500 - 0x80000600	5
APBVGA	0x80000600 - 0x80000700	6
AHBUART	0x80000700 - 0x80000800	7
GRGPIO	0x80000800 - 0x80000900	8
GRTEH	0x80000B00 - 0x80000C00	11
AHBSTAT	0x80000F00 - 0x80001000	15
APB plug&play	0x800FF000 - 0x80100000	-

The address of the on-chip peripherals is defined through the AMBA plug&play configuration, and can be changed by editing the top level design (leon3mp.vhd).

2.18 Signals

The mapping of the signals to the FPGA pins is provided in the *boards/avnet-ml401-xc4vlx25/leon3mp.ucf* file. The following on-board switches are use:

DIP Switch 1:DSUEN - Enables the DSU unit

FPGA RST:Processor reset

Push-button south:DSU break input

3 Simulation and synthesis

3.1 Design flow

Configuring and implementing the LEON3 template design on the ML401 board is done in three basic steps:

- Configuration of the design using xconfig
- Simulation of design and test bench (optional)
- Synthesis and place&route

The template design is based on the GRLIB IP library, and all implementation step are described in detailed in the ‘GRLIB User’s Manual’. *The following sections will summarize these steps, but will not provide a exhaustive description*.

3.2 Installation

The template design is distributed together with the GRLIP IP library. The library is provided as a gzipped tar file, which should be extracted as follows:

```
tar xzf grlib-1.0.17.tar.gz
```

The will create a directory called grlib-eval-1.0.17, containing all IP cores an template designs. On windows hosts, the extraction and all further steps should be made inside a Cygwin shell.

3.3 Template design overview

The template design is located in grlib-1.0.17/designs/leon3-avnet-ml410, and is based on three files:

- *config.vhd* - a VHDL package containing design configuration parameters. Automatically generated by the xconfig GUI tool.
- *leon3mp.vhd* - contains the top level entity and instantiates all on-chip IP cores. It uses config.vhd to configure the instantiated IP cores.
- *testbench.vhd* - test bench with external memory, emulating parts of the ML401 board.

Each core in the template design is configurable using VHDL generics. The value of these generics is assigned from the constants declared in config.vhd, created with the xconfig GUI tool.

3.4 Configuration

Configuration of the template design is done by issuing the ‘make xconfig’ command in the design directory. This will launch the xconfig GUI tool. When the configuration is saved and xconfig is exited, the config.vhd is automatically updated with the selected configuration:

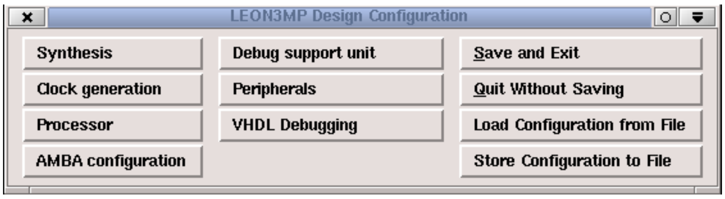


Figure 4. Xconfig GUI

3.5 Simulation

The template design can be simulated in a test bench that emulates the prototype board. The test bench includes external FLASH PROM which is pre-loaded with a test program. The test program will execute on the LEON3 processor, and test various functionality in the design. The test program will print diagnostics on the simulator console during the execution.

The following command should be give to compile and simulate the template design and test bench:

```
make vsim
vsim testbench
```

A typical simulation log can be seen below.

```
$ vsim testbench

VSIM 1> run
# LEON3 Avnet ML401 (Virtex4 LX25) Demonstration design
# GRLIB Version 1.0.5
# Target technology: virtex4 ,  memory library: virtex4
# ahbctrl: mst0: Gaisler Research        Leon3 SPARC V8 Processor
# ahbctrl: mst1: Gaisler Research        AHB Debug UART
# ahbctrl: mst2: Gaisler Research        JTAG Debug Link
# ahbctrl: mst3: Gaisler Research        GR Ethernet MAC
# ahbctrl: slv0: European Space Agency   Leon2 Memory Controller
# ahbctrl:       memory at 0x00000000, size 512 Mbyte, cacheable, prefetch
# ahbctrl:       memory at 0x20000000, size 512 Mbyte
# ahbctrl:       memory at 0x40000000, size 1024 Mbyte, cacheable, prefetch
# ahbctrl: slv1: Gaisler Research        AHB/APB Bridge
# ahbctrl:       memory at 0x80000000, size 1 Mbyte
# ahbctrl: slv2: Gaisler Research        Leon3 Debug Support Unit
# ahbctrl:       memory at 0x90000000, size 256 Mbyte
# ahbctrl: AHB arbiter/multiplexer rev 1
# ahbctrl: Common I/O area disabled
# ahbctrl: Configuration area at 0xfffff000, 4 kbyte
# apbctrl: APB Bridge at 0x80000000 rev 1
# apbctrl: slv0: European Space Agency   Leon2 Memory Controller
# apbctrl:       I/O ports at 0x80000000, size 256 byte
# apbctrl: slv1: Gaisler Research        Generic UART
# apbctrl:       I/O ports at 0x80000100, size 256 byte
# apbctrl: slv2: Gaisler Research        Multi-processor Interrupt Ctrl.
# apbctrl:       I/O ports at 0x80000200, size 256 byte
# apbctrl: slv3: Gaisler Research        Modular Timer Unit
# apbctrl:       I/O ports at 0x80000300, size 256 byte
# apbctrl: slv7: Gaisler Research        AHB Debug UART
# apbctrl:       I/O ports at 0x80000700, size 256 byte
# apbctrl: slv8: Gaisler Research        General Purpose I/O port
# apbctrl:       I/O ports at 0x80000800, size 256 byte
# apbctrl: slv11: Gaisler Research        GR Ethernet MAC
# apbctrl:       I/O ports at 0x80000b00, size 256 byte
# apbctrl: slv15: Gaisler Research        AHB Status Register
# apbctrl:       I/O ports at 0x80000f00, size 256 byte
# greth3: 10/100 Mbit Ethernet MAC rev 03, EDCL 1, buffer 8 kbyte
# ahbstat15: AHB status unit rev 0, irq 7
# grgpio8: 32-bit GPIO Unit rev 0
# gptimer3: GR Timer Unit rev 0, 8-bit scaler, 2 32-bit timers, irq 8
# irqmp: Multi-processor Interrupt Controller rev 3, #cpu 1
# apbuart1: Generic UART rev 1, fifo 8, irq 2
# ahbjtag AHB Debug JTAG rev 0
# ahbuart7: AHB Debug UART rev 0
# dsu3_2: LEON3 Debug support unit + AHB Trace Buffer, 2 kbytes
# leon3_0: LEON3 SPARC V8 processor rev 0
# leon3_0: icache 1*16 kbyte, dcache 1*16 kbyte
# clkgen_ml401: virtex-4 ddr/ssram clock generator, version 1
# clkgen_ml401: Frequency 100000 KHz, DCM divisor 2/2
VSIM 2>
```

The test program executed by the test bench consists of two parts, a simple prom boot loader (prom.S) and the test program itself (systest.c). Both parts can be re-compiled using the ‘make soft’ command. This requires that the BCC tool-chain is installed on the host computer.

3.6 Synthesis and place&route

The template design can be synthesized with either Synplify-8.9 or ISE-9.2i. Synthesis can be done in batch or interactively. To use synplify in batch mode, use the command:

```
make synplify
```

To use synplify interactively, use :

```
make scripts
synplify leon3mp_synplify.prj
```

The corresponding command for ISE are:

```
make ise-map
```

or

```
make scripts
ise leon3mp.ise
```

To perform place&route for a netlist generated with synplify, use:

```
make ise-synp
```

For a netlist generated with XST, use:

```
make ise
```

In both cases, the final programming file will be called ‘leon3mp.bit’. See the GRLIB User’s Manual chapter 3 for details on simulation and synthesis script files.

The maximum frequency is approximately 100 MHz, depending on design configuration and used synthesis tool.

3.7 Board re-programming

The ML401 FPGA configuration PROM (XF32S) can be programmed from the shell window with the following command:

```
make ise-prog-prom
```

To program only the FPGA, use:

```
make ise-prog-fpga
```

For interactive programming, use Xilinx Impact software. See the ML401 Manual for details on which configuration PROM to specify.

A pre-compiled bit file is provided in the bitfiles directory. To load this template design in the FPGA or configuration PROM, do one of the following commands:

make ise-prog-prom-ref

make ise-prog-fpga-ref

The PROM programming command programs the on-board plarform flash (XF32S). The load the FPGA from this prom, set SW12 to ‘platform flash and the CFG ADDR DIP switches to ‘000’.

4 Software development

4.1 Tool chains

The LEON3 processor is supported by several software tool chains:

- Bare-C cross-compiler system (BCC)
- RTEMS cross-compiler system (RCC)
- Snapgear embedded linux
- eCos real-time kernel

All these tool chains and associated documentation can be downloaded from www.gaisler.com.

4.2 Downloading software to the target system

LEON3 has an on-chip debug support unit (DSU) which greatly simplifies the debugging of software on a target system. The DSU provides full access to all processor registers and system memory, and also includes instruction and data trace buffers. Downloading and debugging of software is done using the GRMON debug monitor, a tool that runs on the host computer and communicates with the target through either serial, JTAG or ethernet interfaces. A typical debug session is shown below:

```
$ grmon -ramws 1 -romws 15 -normw -u

 GRMON LEON debug monitor v1.1.7

 Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
 For latest updates, go to http://www.gaisler.com/
 Comments or bug-reports to support@gaisler.com

 using port /dev/ttyS0 @ 115200 baud

 initialising .............
 detected frequency: 100 MHz

 Component                            Vendor
 LEON3 SPARC V8 Processor             Gaisler Research
 AHB Debug UART                       Gaisler Research
 AHB Debug JTAG TAP                   Gaisler Research
 GR Ethernet MAC                      Gaisler Research
 LEON2 Memory Controller              European Space Agency
 AHB/APB Bridge                       Gaisler Research
 LEON3 Debug Support Unit             Gaisler Research
 Generic APB UART                     Gaisler Research
 Multi-processor Interrupt Ctrl       Gaisler Research
 Modular Timer Unit                   Gaisler Research
 General pupose I/O port              Gaisler Research
 GR Ethernet MAC                      Gaisler Research
 AHB status register                  Gaisler Research

 Use command ’info sys’ to print a detailed report of attached cores

grlib> lo ~/examples/soft/v8/dhry.exe
section: .text at 0x40000000, size 47664 bytes
section: .data at 0x4000ba30, size 2408 bytes
total size: 50072 bytes (92.6 kbit/s)
read 178 symbols
entry point: 0x40000000
grlib> run
Execution starts, 400000 runs through Dhrystone
Microseconds for one run through Dhrystone:    6.3
Dhrystones per Second:                      157894.7

Dhrystones MIPS      :                        89.9
```

```
Program exited normally.
grlib> q
```

Please refer to the GRMON User’s Manual for a description of the GRMON operations.

4.3 Flash PROM programming

The ML401 board has two 32 Mbit (2Mx16) Intel flash PROMs, providing a total of 4 Mbyte of storage. A PROM image is typically created with the sparc-elf-mkprom utility provided with the BCC tool chain. The suitable mkprom parameters for the ML401 board are:

sparc-elf-mkprom -romws 15 -freq 100 -ramws 1 -ramsize 1024 -msoft-float -baud 38400

Note that the -freq option should reflect the selected processor frequency, which depends on the clock generator settings.

Once the PROM image has been created, the on-board flash PROM can be programmed through GRMON. The procedure is described in the GRMON manual, below is the required GRMON command sequence:

```
flash enable
flash unlock all
flash erase all
flash load prom.out
flash lock all
```

Information furnished by Gaisler Research is believed to be accurate and reliable.

However, no responsibility is assumed by Gaisler Research for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

No license is granted by implication or otherwise under any patent or patent rights of Gaisler Research.

Gaisler Research AB
Första Långgatan 19
413 27 Göteborg
Sweden

tel +46 31 7758650
fax +46 31 421407
sales@gaisler.com
www.gaisler.com

All information is provided as is. There is no warranty that it is correct or suitable for any purpose, neither implicit nor explicit.