Free Software for Grlib (LEON-3) debugging

Software libre para depuración de la Grlib (LEON-3)

INTI - Electrónica e Informática,

UTIC - Laboratorio de Instrumentación y Control

Authors:

Diego Brengi (brengi@inti.gob.ar)

Salvador Tropea (salvador@inti.gob.ar).

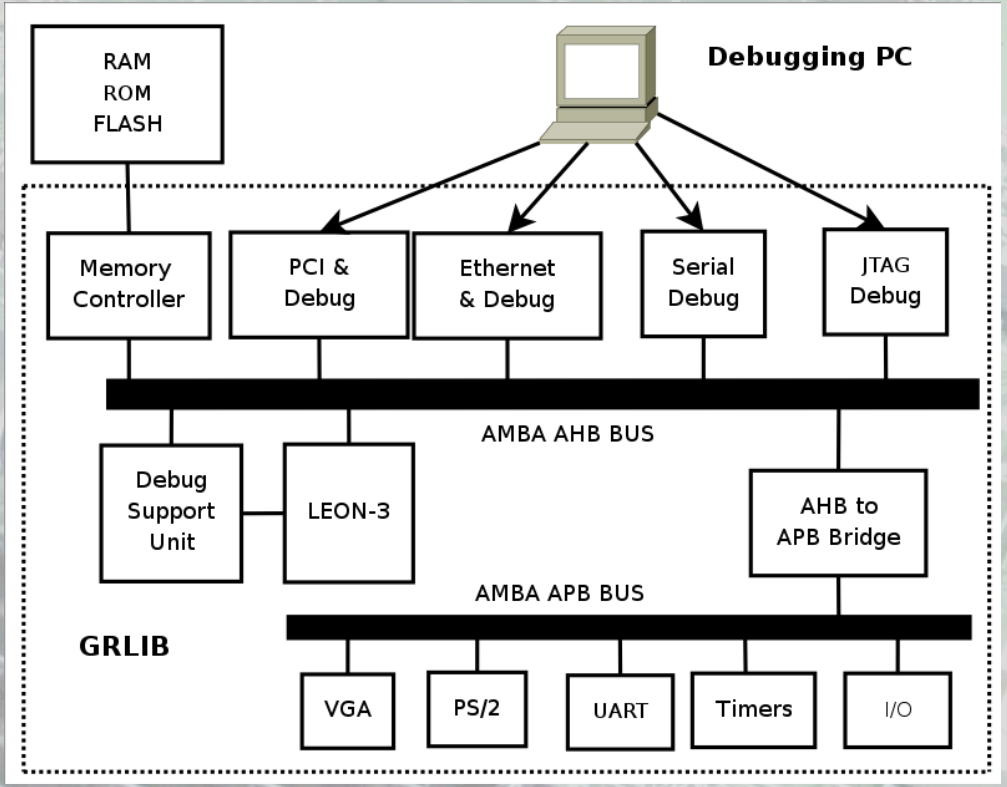
Outline:

- Grlib and LEON-3 introduction
- Grlib Debugging
- FpgaLibre LEON Monitor
- Usage examples
- Getting Flemon
- Future work

GRLIB and LEON-3

Introduction to GRLIB and LEON-3

- Grlib is a VHDL IP core library developed and maintained by Aeroflex/Gaisler.
- Main core is LEON-3, a 32-bit processor with SPARC v8 architecture.
- Uses AMBA AHB (high data traffic) and APB buses (low bandwidth).
- A lot of support cores: JTAG, memory controllers, IO ports, timers, Ethernet, PCI and more.
- Template designs for various FPGA boards.
- Registration method for IP cores helps in debugging (Plug&Play).



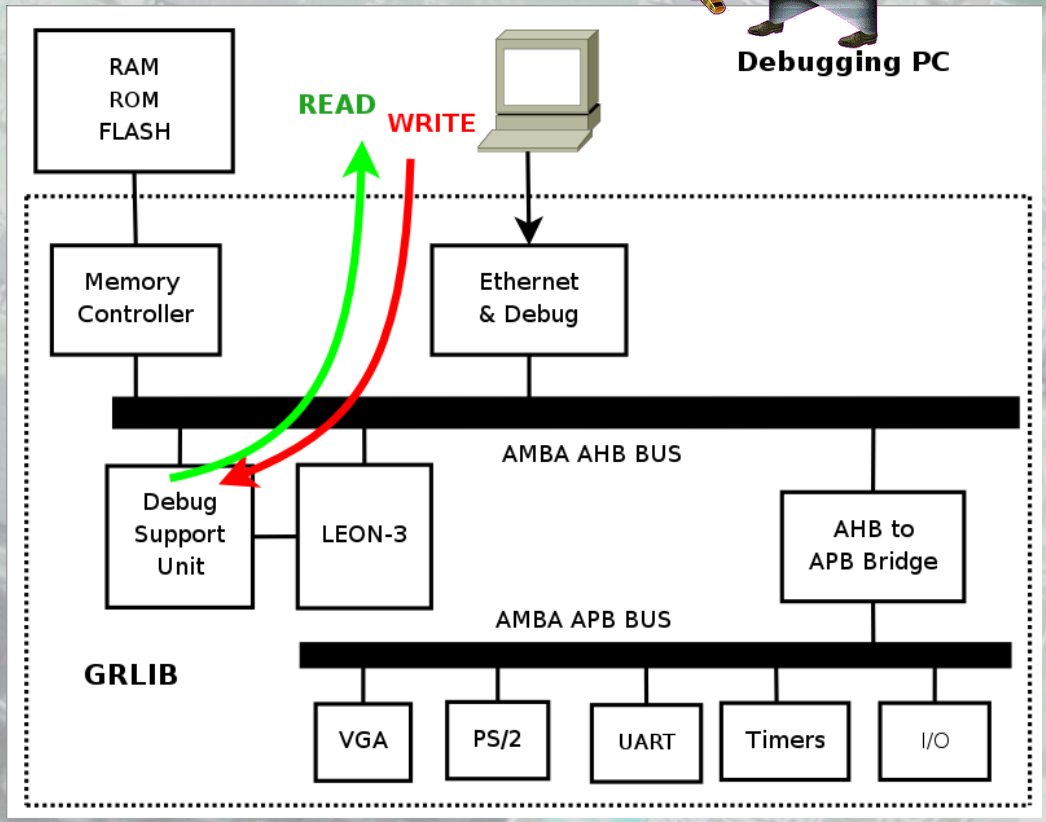
GRLIB and LEON-3

Open and royalty free

- Most parts of the Grlib are under the terms of the GPL license.
- Grlib is vendor independent (CAD tools and target technologies).
- SPARC v8 architecture is an open standard. A USD 99.0 fee gives you rights to design, manufacture and market devices conforming to this architecture (but not needed to use LEON).
- AMBA is an ARM specification. License allows you to freely use it on any type of systems. Allows selling, developing an distributing too.
- GCC (The most popular free software C compiler) has the SPARC processor as one of his target architecture.
- GNU/Linux and uCLinux runs on SPARC processors systems. Special Embedded versions like Snapgear can run on Grlib systems.
- Grlib testbench with GHDL (A free software VHDL simulator based on GCC).
- Grlib uses free tools like Bash, Make, GCC and tcl/tk.

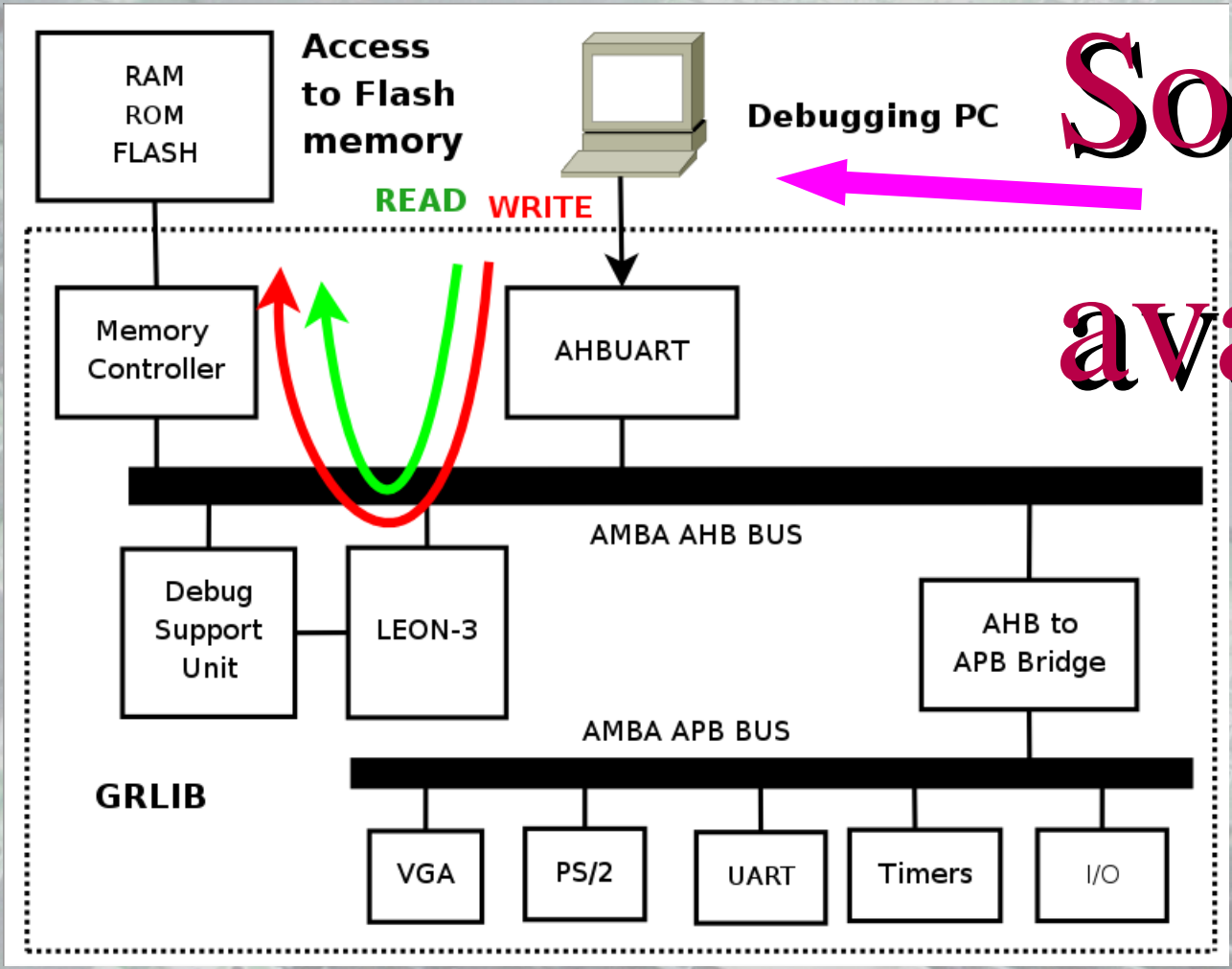
Grlib Debugging

- Grlib has special IP cores for debugging with: USB, Serial, Ethernet, PCI, JTAG and more (not all of them are GPL).
- This debug cores give you access to the internal AHB bus for read an write operations.
- There are special bus addresses with information about available IP cores.
- After reading the P&P area, you can debug the LEON-3 processor through the Debug Support Unit.



Example: Debugging the LEON processor through Ethernet.

Grlib Debugging



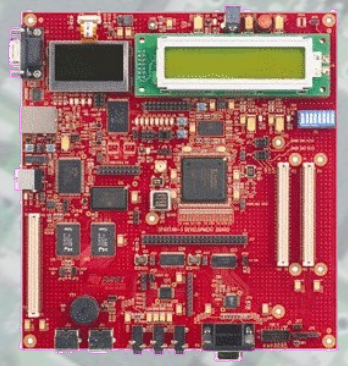
Software available?

FleMon - FPGALibre LEON Monitor

Motivation

- FleMon is a software for Grlib Debugging.
- Part of the FPGALibre project “Free an open source tools and cores for FPGA developing”. http://fpgalibre.sf.net
- It is free software (GPL License).
- It was developed because there was no other free alternative. Only evaluation (academic/limited time) or paid software available (grmon).
- First feature wanted: to write an embedded Linux image into the board's flash memory without proprietary software.

FleMon - FPGALibre LEON Monitor



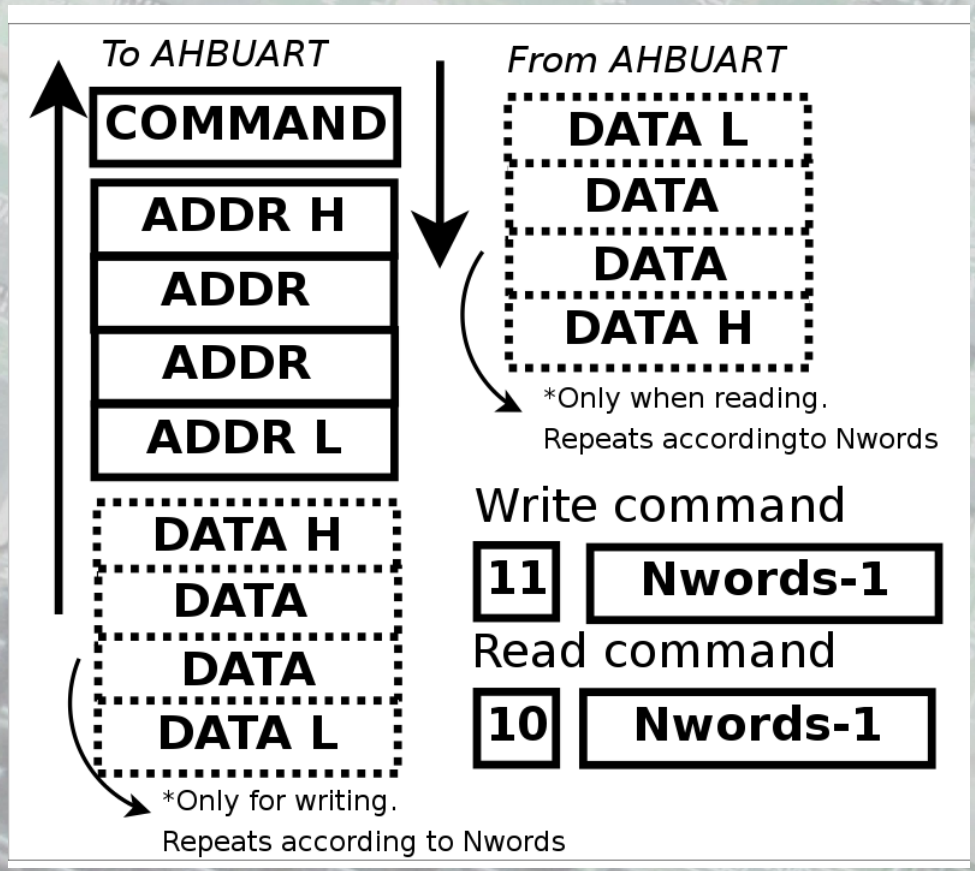
Developing

- Developed and tested with Debian GNU/Linux.
- Using C++ and compiled with GCC.
- Using libreadline and libelf.
- Setedit for source code edition.
- Interceptty and flemonspy for serial communication analysis.
- Tested on FPGA boards:
  - Pender GR-XC3S-1500.
  - AVNET Virtex 4-LX Evaluation (XC4VLX25).

FleMon - FPGALibre LEON Monitor

Serial Communication

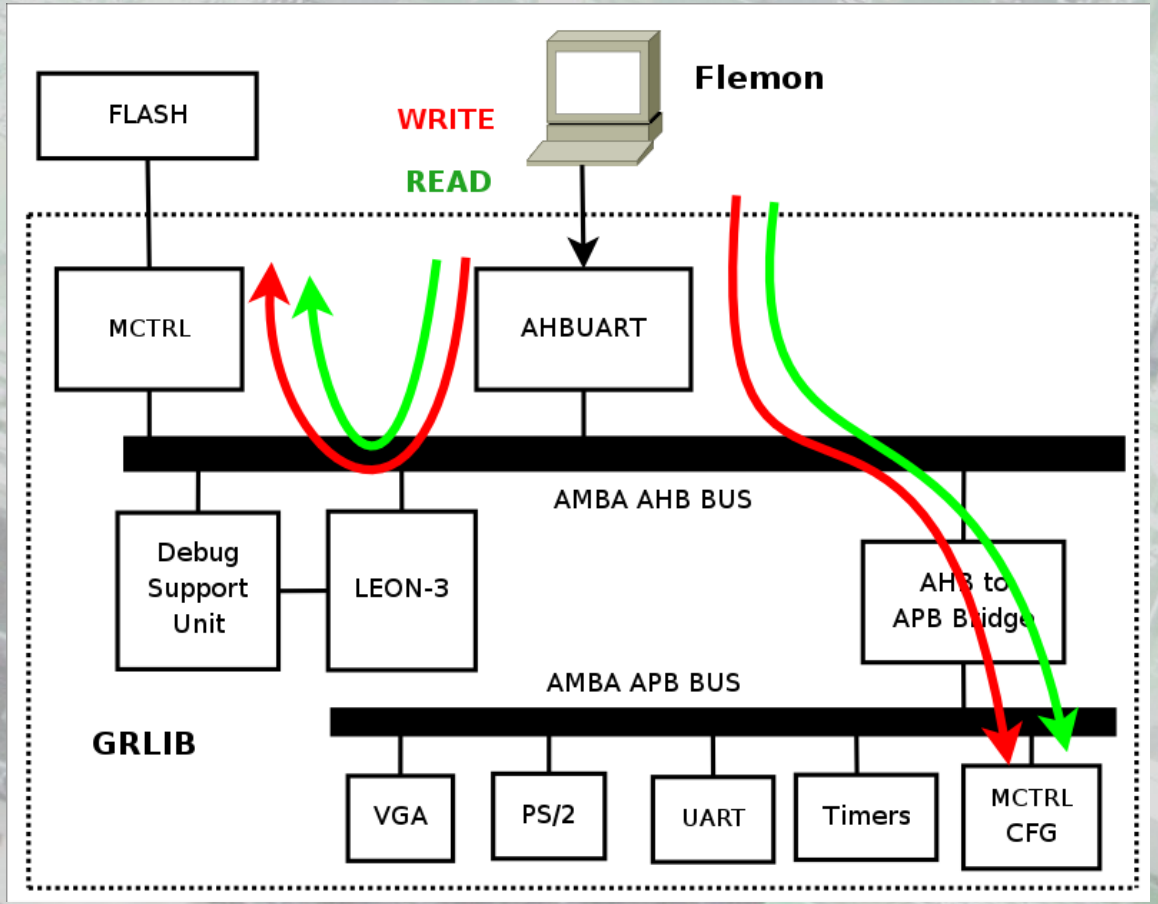
- FleMon has now only serial communication (AHBUART).
- This protocol is simple. Two types of commands: read an write.
- Payload data between 1 and 64 (32 bits words).
- Addresses an data are always 32 bits long.



FleMon - FPGALibre LEON Monitor

Discovering

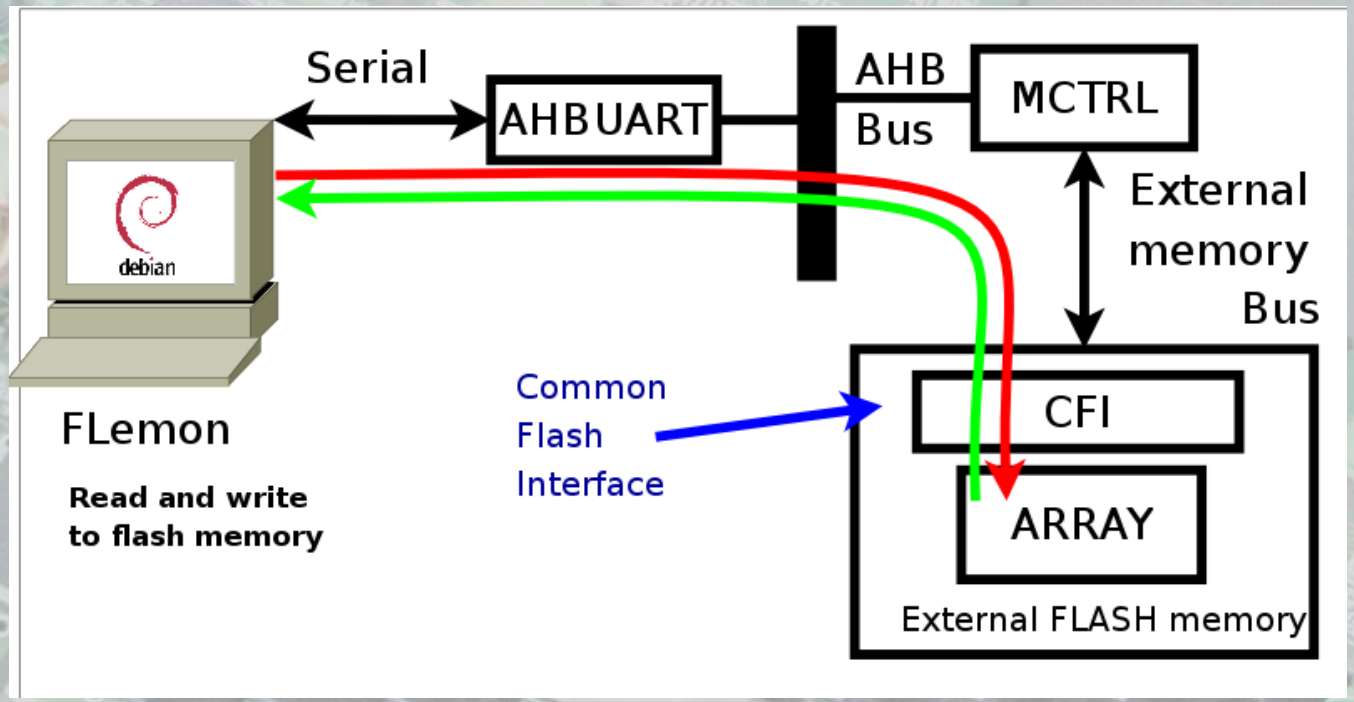
- FleMon reads Plug&Play information and then knows which IP cores are present in your design.
- If an AHB/APB bridge is detected, scans Plug&Play information present in the APB bus.
- After that, FleMon can access MCTRL (Memory controller).
- MCTRL : AHB bus for transactions and APB for register configuration.



FleMon - FPGALibre LEON Monitor

Using the flash memory

- Flash memory must be CFI (Common Flash Interface) compliant.
- FleMon reads CFI geometry registers to know how to program flash memory.
- After that, write and erase operations to flash can be done (using CFI commands).



FLeMon

FPGALibre LEON Monitor

- FleMon reads ELF files (Executable and Linkable Format) for SPARC.
- This is needed when writing a Linux distro image like Snapgear/Gaisler or a compiled program to flash memory.

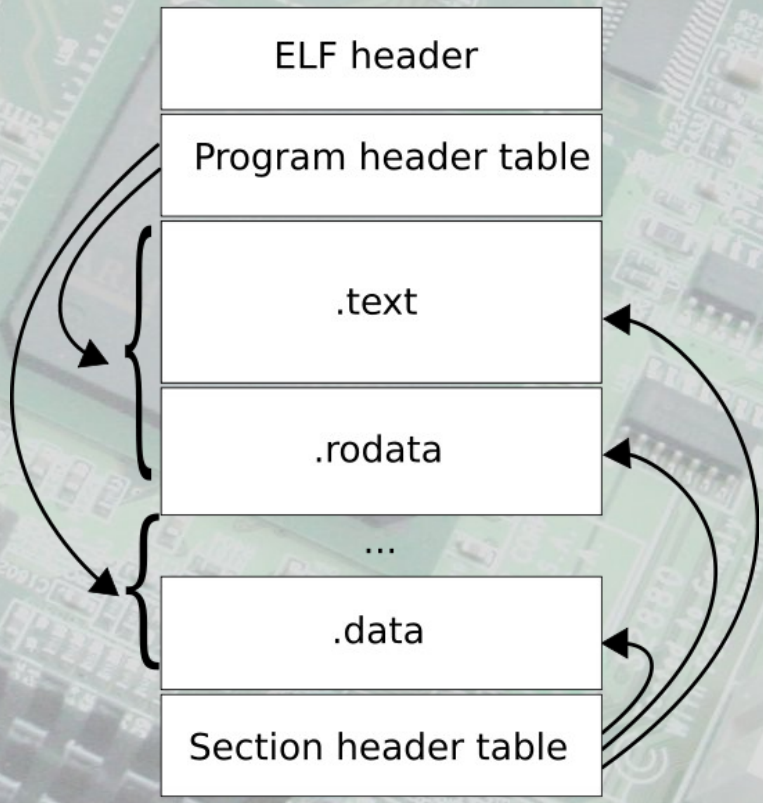


Image from:
http://en.wikipedia.org/wiki/Executable_and_Linkable_Format

FleMon - FPGALibre LEON Monitor

USAGE - Arguments

- Basic command line options.
- User can put this options in a configuration file: “flemon.cfg”.

```
diego@Demeter:~$ flemon -h
FLeMon - FPGALibre Leon Monitor $Revision: 1.37 $
Copyright (c) 2007-2009 Diego J. Brengi <brengi at inti gov ar>
Copyright (c) 2007-2009 Instituto Nacional de Tecnología Industrial

Usage:
flemon [options]

Available options:

-D, --device=DEVICE    Device for IO. Example -d /dev/ttyS0
-b, --baud=BAUDRATE    Serial baudrate
-d, --dlist=FILE       Use a custom deviceid file
-g, --plugplay         Count Plug&Play records as a core.
-h, --help             print this help and exit
-q, --quiet            reduce verbosity
-v, --verbose          increase verbosity
-V, --version          show version information and exit
```

- Type "help" to see available commands

```
FLeMon$ help
Commands:
  help            Display this text.
  ?               Synonym for `help'.
  quit            Quit using FleMon.
  exit            Quit using FleMon.
  sync            Send SYNC bytes and establish connection.
  scanh           Scan only AHB bus.
  scanp           Scan only APB bus.
  scan            Scan AHB and APB buses.
  detect          Commands: sync, scanh, scanp and fscan.
  ls              List AHB/APB detected cores. -l extended. -c collapse.
  mem             Report memory map.
  cleancores      Erase all core's information.
  fscan           Flash detection.
  finfo           Flash report.
  flock           Flash lock [all | offset].
  funlock         Flash unlock (all banks)
  fstat           Flash lock bits status.
  fcfi            Flash CFI dump.
  ferase          Flash bank erase. [offset] or 'all'.
  fdumpblock      Flash dump block. [offset|all start] [offset end].
  fwrite          Flash write a single word. [offset] [data]
  fcheck          Flash blank check. [offset start] [offset end]
  fdumpfile       Flash dump to a file. [offset|all start] [offset end].
  ffload          Flash load a plain data file. [filename]
  fferase         Erase only needeed banks for a plain data file. [filename]
  ffverify        Flash data verification (after fload). [filename]
  fxerase         Erase only needed banks for an ELF file. [filename]
  fxload          Flash load ELF file. [filename]
  fxverify        Flash ELF verification (after fxload). [filename]
```

FleMon - FPGALibre LEON Monitor

USAGE – Initial detection

- Interactive command line with autocomplete for commands and filenames (TAB key and arrows for command history)
- Using libreadline.
- The "detect" command does the following:
  - Sends "sync" bytes to uart.
  - Scans AHB bus: "scanh".
  - Scans APB bus: "scanp".
  - Reads CFI info: "fscan".

```
FLeMon$ detect
Check address 80000000: 0x000000FF
Bridge found..
MCFG1 = 0x000008FF
MCTRL mode: 8 bit  flash on D[31:24]
Prom Waitstates:  read=15   write=15
Prom write enable: YES
Flash is CFI compliant.
CFI Vendor: 0x89 Device: 0x17  INTEL 64 Mbit
Command set: 0x0001 - Intel/Sharp Extended.
--------CFI Geometry info---------
Device size: 8388608 bytes
Erase block regions: 1
Erase blocks size in region 1: 131072 bytes
(y=512)
Erase blocks in region 1: 64 blocks
Total Size: 8388608 bytes
FLeMon$
```

```
VENDOR=0x01="Gaisler Research"
 0x002="LEON2 Debug Support Unit"
 0x003="LEON3 SPARC V8 Processor"
 0x004="LEON3 Debug Support Unit"
 0x005="10/100 Mbit Ethernet MAC"
 0x006="AHB/APB Bridge with Plug&Play"
 [...]
 0x012="32-bit target-only PCI interface"
 0x015="32-bit PCI Trace Buffer"
 0x016="DMA Controller for PCI MTF"

VENDOR=0x02="Pender Electronic Design"

VENDOR=0x04="European Space Agency"
 0x002="LEON2 SPARC V8 Processor"
 0x003="LEON2 Peripheral Bus"
 0x005="LEON2 Interrupt Ctrl"
 0x006="LEON2 Timer Unit"
```

- Each type of core is associated with a unique VendorID/DeviceID number (Grlib guidelines).
- Instead of numbers, it's nice to see descriptions for better and quick human interpretation.
- FleMon has this information inside a plain text file called “devices”.
- The user can freely change this file, or add new VendorID/DeviceID descriptions.

FleMon - FPGALibre LEON Monitor

USAGE – Listing

- Listing detected IP cores in the system.

```
FLeMon$ ls
============== DETECTED SYSTEM CORES ==============
01- LEON3 SPARC V8 Processor                        Gaisler Research
02- AHB Debug UART                                  Gaisler Research
03- JTAG/AHB Debug Interface                        Gaisler Research
04- SVGA video frame buffer                         Gaisler Research
05- GR Ethernet MAC                                 Gaisler Research
06- USB 2.0 Debug Communication Link                Gaisler Research
07- IDE/ATA Controller                              Gaisler Research
08- 8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl    European Space Agency
09- AHB/APB Bridge with Plug&Play                   Gaisler Research
10- LEON3 Debug Support Unit                        Gaisler Research
11- IDE/ATA Controller                              Gaisler Research
12- APB BUS Plug&Play info (inferred)               Gaisler Research
13- 8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl    European Space Agency
14- Generic APB UART                                Gaisler Research
15- LEON3 Multi process Interrupt Ctrl              Gaisler Research
16- Modular Timer Unit                              Gaisler Research
17- APB Keyboard PS/2 interface                     Gaisler Research
18- APB Keyboard PS/2 interface                     Gaisler Research
19- SVGA video frame buffer                         Gaisler Research
20- AHB Debug UART                                  Gaisler Research
21- General purpose I/O port                        Gaisler Research
22- GR Ethernet MAC                                 Gaisler Research
FLeMon$
```

FleMon - FPGALibre LEON Monitor

USAGE – Address report

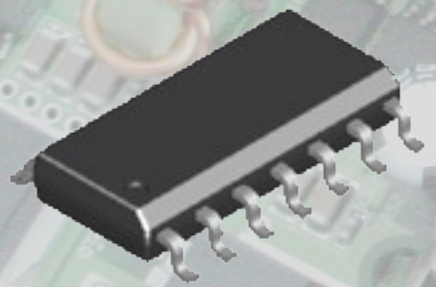
- Bus addresses report.

```
FLeMon$ mem
ahb mem:  00000000-20000000  8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl
ahb mem:  20000000-40000000  8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl
ahb mem:  40000000-80000000  8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl
ahb mem: [80000000-80100000] AHB/APB Bridge with Plug&Play
apb i/o:  80000000-80000100  8/16/32bit PROM/IO/SRAM/SDRAM LEON2 Mem Ctrl
apb i/o:  80000100-80000200  Generic APB UART
apb i/o:  80000200-80000300  LEON3 Multi proccess Interrupt Ctrl
apb i/o:  80000300-80000400  Modular Timer Unit
apb i/o:  80000400-80000500  APB Keyboard PS/2 interface
apb i/o:  80000500-80000600  APB Keyboard PS/2 interface
apb i/o:  80000600-80000700  SVGA video frame buffer
apb i/o:  80000700-80000800  AHB Debug UART
apb i/o:  80000800-80000900  General purpose I/O port
apb i/o:  80000B00-80000C00  GR Ethernet MAC
apb i/o:  800FF000-800FF080  APB BUS Plug&Play info (inferred)
ahb mem:  90000000-A0000000  LEON3 Debug Support Unit
ahb i/o:  FFFA0000-FFFA0100  IDE/ATA Controller
FLeMon$
```

FleMon - FPGALibre LEON Monitor

USAGE – Flash memory commands

Basic operations were implemented:



- Lock and unlock memory blocks:***fstatus, funlock and flock.***
- Erase memory blocks:***ferase, fferase and fxerase.***
- Write a single word: ***fwrite.***
- Read flash memory:***fdumpblock and fdumpfile.***
- Blank check:***fcheck***
- Loading operations:***ffload and fxload.***
- Verify operations:***ffverify and fxverify.***
- Debug:***finfo and fcfi***

FleMon - FPGALibre LEON Monitor

USAGE – Erasing Flash

- The *"fxerase"* takes an ELF file as an argument and erase only the blocks needed.

```
FLeMon$ fxerase /tmp/grlib_linux_pender/image.flashbz
Erasing flash memory...
  Opening file [/tmp/grlib_linux_pender/image.flashbz]
   ELF 32-bit Executable Sparc
Size:0002257036bytes Off:0x00010000 addr:0x00000000 .text  <-This!
Size:0000000000bytes Off:0x0023B100 addr:0x4031B100 .data
Size:0000033848bytes Off:0x0023B100 addr:0x4031B100 .bss
Size:0000000054bytes Off:0x0023B100 addr:0x00000000 .comment
Size:0000000000bytes Off:0x0023B136 addr:0x00000000 .note.GNU-stack
Size:0000000069bytes Off:0x0023B136 addr:0x00000000 .shstrtab
Size:0000001680bytes Off:0x0023B2E4 addr:0x00000000 .symtab
Size:0000001165bytes Off:0x0023B974 addr:0x00000000 .strtab
Erased address 0+0. Erase time: 1.03 seg. Total:1.03
Erased address 0+131072. Erase time: 1.04 seg. Total:2.07
Erased address 0+262144. Erase time: 1.03 seg. Total:3.10
Erased address 0+393216. Erase time: 1.03 seg. Total:4.14
...
Erased address 0+1966080. Erase time: 1.04 seg. Total:16.55
Erased address 0+2097152. Erase time: 1.03 seg. Total:17.58
Erased address 0+2228224. Erase time: 1.03 seg. Total:18.61
Total time: 18.61 seg.
```

FleMon - FPGALibre LEON Monitor

USAGE – Writing a Snapgear Linux image to flash memory

- The "fxload" command takes an ELF file as an argument and write it to flash memory.

```
FLeMon$ fxload /tmp/grlib_linux_pender/image.flashbz
  Flemon ELF Flash loader
  Opening file [/tmp/grlib_linux_pender/image.flashbz]
   ELF 32-bit Executable Sparc
Size:0002257036bytes Off:0x00010000 addr:0x00000000 .text  <-This!
Size:0000000000bytes Off:0x0023B100 addr:0x4031B100 .data
Size:0000033848bytes Off:0x0023B100 addr:0x4031B100 .bss
Size:0000000054bytes Off:0x0023B100 addr:0x00000000 .comment
Size:0000000000bytes Off:0x0023B136 addr:0x00000000 .note.GNU-stack
Size:0000000069bytes Off:0x0023B136 addr:0x00000000 .shstrtab
Size:0000001680bytes Off:0x0023B2E4 addr:0x00000000 .symtab
Size:0000001165bytes Off:0x0023B974 addr:0x00000000 .strtab
File: [/tmp/grlib_linux_pender/image.flashbz]  Offset[0x00010000-65536d]
Lenght[0x22708C-2257036d] pad=0
32/2257036 (0%) Time: 0.01 segs. Write/Skip: W
22624/2257036 (1%) Time: 6.12 segs. Write/Skip: W
45216/2257036 (2%) Time: 12.17 segs. Write/Skip: W
...
2214048/2257036 (98%) Time: 601.45 segs. Write/Skip: W
2236640/2257036 (99%) Time: 607.46 segs. Write/Skip: W

  Total bytes: 2257036 Time: 10.219484 min Kbits/seg: 28.76
```

FleMon - FPGALibre LEON Monitor

USAGE – Verifying

- The "fxverify" command takes an ELF file as an argument and compare it against flash memory.
- Abort at first error.

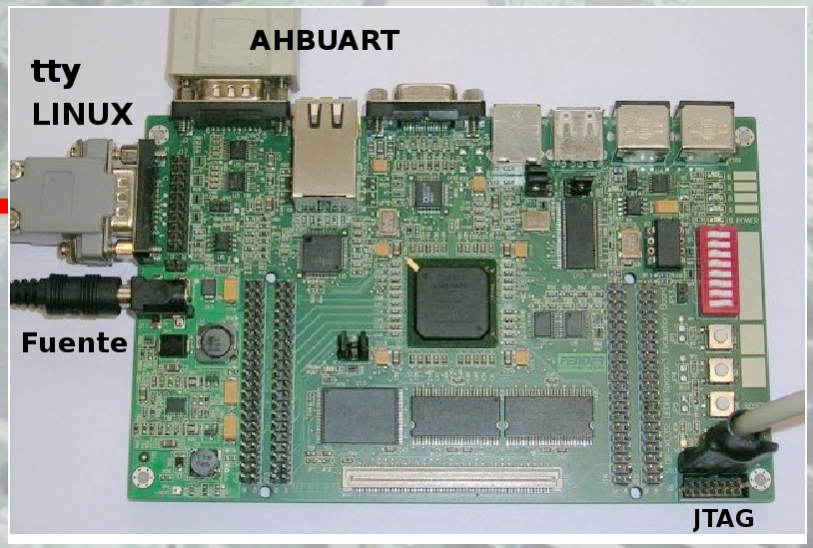
```
FLeMon$ fxverify /tmp/grlib_linux_pender/image.flashbz
  File verification
  Opening file [/tmp/grlib_linux_pender/image.flashbz]
   ELF 32-bit Executable Sparc
Size:0002257036bytes Off:0x00010000 addr:0x00000000 .text  <-This!
Size:0000000000bytes Off:0x0023B100 addr:0x4031B100 .data
Size:0000033848bytes Off:0x0023B100 addr:0x4031B100 .bss
Size:0000000054bytes Off:0x0023B100 addr:0x00000000 .comment
Size:0000000000bytes Off:0x0023B136 addr:0x00000000 .note.GNU-stack
Size:0000000069bytes Off:0x0023B136 addr:0x00000000 .shstrtab
Size:0000001680bytes Off:0x0023B2E4 addr:0x00000000 .symtab
Size:0000001165bytes Off:0x0023B974 addr:0x00000000 .strtab
File: [/tmp/grlib_linux_pender/image.flashbz]  Offset[0x00010000-65536d]
Length[0x22708C-2257036d] pad=0
32/2257036 (0%) Time: 0.01 segs. Write/Skip: V
22624/2257036 (1%) Time: 5.41 segs. Write/Skip: V
45216/2257036 (2%) Time: 10.88 segs. Write/Skip: V
...
2214048/2257036 (98%) Time: 531.02 segs. Write/Skip: V
2236640/2257036 (99%) Time: 536.42 segs. Write/Skip: V

  Total bytes: 2257036 Time: 9.022266 min Kbits/seg: 32.57
```

FleMon - FPGALibre LEON Monitor

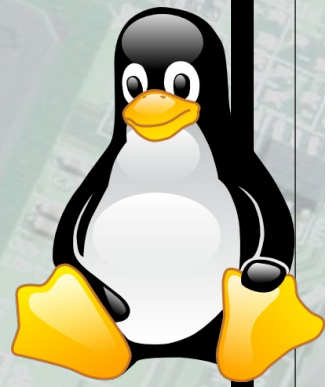
Booting Linux!!!

minicom

38400 8N1




```
decompress_kernel(to: 40000000,freemem:40323538,freemem_end:43ffdf38)
output_data:40000000,free_mem_ptr:40323538,free_mem_ptr_end:43ffdf38
  - Inputbuffer [ptr: 3168, sz: 223f24]
.....................................................................
.........................
done [sz:0x31b100], booting the kernel.
Booting Linux...
PROMLIB: Sun Boot Prom Version 0 Revision 0
Linux version 2.6.21.1 (daniel@neptune) (gcc version 3.4.4) #81 Thu Dec
6 10:27:29 CET 2007
ARCH: LEON       Vendors          Slaves
Ahb masters:
 0( 1:  3| 0):   VENDOR_GAISLER    GAISLER_LEON3
 1( 1:  7| 0):   VENDOR_GAISLER  GAISLER_AHBUART
 2( 1: 1c| 0):   VENDOR_GAISLER  GAISLER_AHBJTAG
 3( 1: 63| 0):   VENDOR_GAISLER     GAISLER_SVGA
 4( 1: 1d| 0):   VENDOR_GAISLER   GAISLER_ETHMAC
 5( 1: 22| 0):   VENDOR_GAISLER Unknown device 22
[...]
TCP: Hash tables configured (established 2048 bind 2048)
TCP reno registered
io scheduler noop registered
io scheduler cfq registered (default)
grlib apbuart: 1 serial driver(s) at [0x80000100(irq 2)]
grlib apbuart: system frequency: 40000 khz, baud rates: 38400 38400
ttyS0 at MMIO 0x80000100 (irq = 2) is a Leon
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
Probing GRETH Ethernet Core at 0x80000b00
Auto negotiation timed out. Selecting default config
10/100 GRETH Ethermac at [0x80000b00] irq 12. Running 100 Mbps full
duplex
[...]
Sash command shell (version 1.1.1)
/>
```



Command shell

FleMon - FPGALibre LEON Monitor

Getting FleMon

FleMon is available at :

http://fpgalibre.sourceforge.net
http://fpgalibre.sf.net

Source code and Debian package (unofficial).

FleMon - FPGALibre LEON Monitor

Future work (or call for contributions)

FleMon is experimental software and is still in development.

- Add Ethernet debug communication (faster flash memory access). Jtag and PCI are also missing.
- Add specific IP core reports. Each IP core has configuration registers. There is a lot of information to show. Some kind of easy and open mechanism is needed to allow users adding core's reports.
- Test FleMon in more Hardware/Grlib combinations.
- Add processor debug capabilities using the DSU unit, trace buffer, etc. Add GDB support for that.
- Clean source code and put some functionalities inside a library.

Permanent Staff:

- Tropea, Salvador E. <salvador@inti.gob.ar>
- Brengi, Diego J. <brengi@inti.gob.ar>
- Rodrigo Melo <rmelo@inti.gob.ar>
- Darío Farías <dario@inti.gob.ar>

¡Thanks!