LEON SPARC Processor
The past, present and future

Jiri Gaisler
Gaisler Research
Presentation outline

- Brief history of ESA processor design
- The development of LEON1 & LEON2
- LEON3 and GRLIB overview
- LEON4 and upcoming projects
What is LEON?

- LEON is a 32-bit SPARC processor, implemented as a synthesisable VHDL model.
- LEON was primarily developed for critical space applications, funded by the European Space Agency (ESA).
- The LEON VHDL model was released in open-source to improve test coverage and adoption of SPARC ISA.
- Three processor versions have so far been developed: LEON1, LEON2 and LEON3.
- Today, the LEON3 processor is part of a larger IP core library called GRLIB, making up a versatile SOC platform for both FPGA and ASIC.
ESA Processor Development

- 1989: MDC281 (1750) clone, 2.5 um CMOS/SOS, 0.5 MIPS
- 1991: MA31750, 1.5 um CMOS/SOS, 2 MIPS
- 1992: SPARC architecture selected as ESA baseline
  - In competition with MIPS, NS32, M88K, AMD29K
- 1995: 3-chip ERC32 (SPARC V7), 10 MIPS, 0.8 um
  - ISS control computer, ~ 10 missions
- 1998: single-chip ERC32 (TSC695), 15 MIPS, 0.6 um
  - Current standard processor for all ESA missions
  - Used by NASA, China, India, Israel
- 2000: First LEON1FT, 0.35 um, 50 MIPS, 0.5 W
- 2002: First LEON2FT, 0.18 um, 100 MIPS, 0.6 W
- 2004: First LEON3FT, 0.20 um, 150 MIPS, 0.4 W
- 2007: LEON3FT quad-core, 90nm, 4x500 MIPS, 3 W
Space Station Control Computer

DMS-R FT Computer
- 4 voting nodes
- ERC32@ 14 MHz
- Inmos T400 @ 10 MHz
- VME internal bus
- 2 redundant clusters
- 2 Control posts (WS)
- VxWorks-5.3 + FT
- 50 Watts
Automated transfer vehicle (ATV)

ATV ISS supply ship
- 5 tonnes supply
- Orbit re-hosting
- DMSR computer reuse
- Launch 2007
1997: ERC32 successor needed

- **ERC32 problems**
  - Proprietary design at schematic/layout level
  - Difficult to port
  - Complex interface
  - 20 MHz limit due to memory interface

- **LEON project goals**
  - European design
  - Radiation hard and SEU free
  - Standard interfaces
  - Modular
  - Portable
  - Written in VHDL
  - 100 MIPS, 20 MFLOPS
LEON1FT Demonstrator

First LEON design
- 5-stage pipeline
- 2 x 4 Kbyte caches
- Meiko FPU
- Custom on-chip bus
- PROM/SRAM control
- Full FT logic
- 30 mm², 100 Kgates
- 50 MHz, 0.5 W
- 20 MHz on Virtex-300E
- 300 Krad, SEU-free
LEON1FT Radiation test board

- Dual LEON1FT in master/checker m.
- 300 Krad
- SEU proof with FT
Louvain Cyclotron setup
LEON2FT – first flight part

- LEON1FT 5-stage pipeline with HW MUL/DIV
- Multi-way caches with LRU
- On-chip AMBA bus for modularity
- 32-bit PC133 SDRAM controller with EDAC
- 32-bit full PCI interface with DMA
- On-chip debug support unit (DSU)
- Maintained FT logic
- Targeted for 100 MHz on 0.18 um processes
- 120% performance improvement over LEON1
LEON2 block diagram and layout

UMC 0.18: 120 MHz, 0.3 W
FT logic verified through heavy-ion injection
Next step: GRLIB

- LEON2 was designed for a single function (processor), but was increasingly being used as SOC platform. A more efficient SOC platform was needed to minimize design work.

- Design goals for GRLIB IP library
  - Portability
  - CAD tool independence
  - Coherent IP interfaces
  - Uniform method for HW & SW debug
  - Rich functionality
  - Processor with MP support
LEON3/GRLIB SOC Architecture
Portability

- Process portability achieved through block wrappers
- Definition of 'virtual' components (pads, memory, MUL)
- Size, technology etc. are set through VHDL generics
- Technology specific mega-cells are instantiated with VHDL generate statements
- Modular design - new technology easily added
- Support of both ASIC and FPGA technologies
  - UMC, TSMC, Atmel, Virage, Virtual Silicon
  - Actel, Altera, Xilinx, Lattice
- 30+ template designs for common FPGA boards
CAD Tool Independence

- GRLIB supports all major CAD tools
- Robust coding style – 'least common denominator'
- Automatic generation of synthesis and simulation scripts
- Supported tools:
  - Mentor, Cadence, Synopsys, Aldec
  - Synplify, ISE, Quartus, ISPLever, Designer
- New CAD tool easily added but needs thorough testing
GRLIB Plug & play

- Unique method to quickly assemble a complex SOC design
- PCI-style plug&play support for AMBA configuration:
  - Device, vendor & version identification
  - Cacheability, pre-fetch information
  - Address and interrupt configuration
- Up to four memory BARs per slave
- Configuration set through VHDL generics
- Plug&play information routed in sideband signals, accessible as small ROM at dedicated address
- Fully compatible with AMBA-2.0
VSIM 1> run
# LEON3 Demonstration design
# GRLIB Version 1.0
# Target technology: inferred, memory library: inferred
# ahbctrl: AHB arbiter/multiplexer rev 1
# ahbctrl: Common I/O area at 0xffff0000, 1 Mbyte
# ahbctrl: Configuration area at 0xffffffff, 4 kbyte
# ahbctrl: mst0: Gaisler Research Leon3 SPARC V8 Processor
# ahbctrl: slv0: Gaisler Research PROM/SRAM/SDRAM Controller
# ahbctrl: memory at 0x00000000, size 16 Mbyte, cacheable, prefetch
# ahbctrl: memory at 0x40000000, size 16 Mbyte, cacheable, prefetch
# ahbctrl: slv1: Gaisler Research AHB/APB Bridge
# ahbctrl: memory at 0x80000000, size 16 Mbyte
# apbmst: APB Bridge at 0x80000000 rev 1
# apbmst: slv1: Gaisler Research Generic UART
# apbmst: I/O ports at 0x80000100, size 256 byte
# apbmst: slv2: Gaisler Research Multi-processor Interrupt Ctrl.
# apbmst: I/O ports at 0x80000200, size 256 byte
# greth5: GRETH 10/100 Mbit ethernet MAC, rev 0, irq 12
# apbictrl: Multi-processor Interrupt Controller rev 1, #cpu 2
# apbuart1: Generic UART rev 1, irq 3
# leon3_0: LEON3 SPARC V8 processor rev 0
# leon3_0: icache 1*2 kbyte, dcache 1*1 kbyte

# cpu0: 0x00000000 flush 0x0000
# cpu0: 0x00000004 sethi %hi(0x00001000), %g1 [0x00001000]
# cpu0: 0x00000008 or %g1, 0x00c0, %g1 [0x000010c0]
# cpu0: 0x0000000c mov %g1, %psr
# cpu0: 0x00000010 mov 0, %wim
GRLIB debugging

- Debugging of hardware and software is done through debug interfaces connected to the AHB bus.
- A number of interfaces are available: JTAG, PCI, RS232, USB, Ethernet, Spacewire
- Trace-buffer for instructions, AHB or PCI can be enabled
- On-chip logic analyzer available as IP core
- Software debugging handled through GRMON debug tool
LEON3/GRLIB Debug interfaces

LEON3

Debug Unit

JTAG

RS232

GPIO

IRQ

TIMER

UART

APB

MCTRL

PCI

ETH

USB

SPW

PROM/SDRAM

32-bit PCI

PHY

PHY

LVDS
GRLIB Open-Source Cores

- LEON3 32-bit SPARC V8 Processor
- AMBA AHB Controller/Arbiter & AHB/APB Bridge
- PROM, SRAM, SDRAM, DDR controllers
- 10/100 Mbit Ethernet MAC
- 32-bit PCI Bridge with optional DMA and FIFO
- CAN-2.0 with FIFO
- UART, Timers, Interrupt controller, GPIO, CLK/RST gen.
- JTAG/TAP controllers
- SVGA frame buffer
- IDE interface for disks and CF
GRLIB Commercial Cores

- Fully pipelined IEEE-754 FPU (singe/double)
- Low-area single-issue IEEE-754 FPU (singe/double)
- AHB/AHB bridge with prefetch and FIFO
- USB-2.0 device controller with DMA
- 1G Ethernet MAC with UDP/TCP off-loading
- Fault-Tolerant LEON3FT for Military and Space app.
- Memory controllers with ECC (BCH & Reed-Solomon)
- MIL-STD-1553 BC/RT/BM interfaces with DMA
- Spacewire 200 Mbit/s serial link with DMA
- Only commercially available (free netlists for evaluation)
LEON3 SPARC V8 Processor

- 7-stage pipeline, multi-processor support
- Separate multi-way caches with LRU/LRR/RND
- Highly configurable:
  - Way size 1-256 Kbyte, 1-4 ways, LRU/LRR/Random
  - Hardware MUL/DIV/MAC options, FPU, MMU, Co-Proc.
  - Pipeline optimization for specific target technologies
- On-chip debug support unit with trace buffer
- 250/400 MHz on 0.18/0.13 um, 250/400 MIPS, 25 Kgates
- 125 MHz on Virtex2pro FPGA, 3500 LUT
- Fault-tolerance by design for space applications
LEON3 Configuration GUI
LEON3 on-chip instruction trace

<table>
<thead>
<tr>
<th>time</th>
<th>address</th>
<th>instruction</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>120828287</td>
<td>400096c0</td>
<td>sethi %hi(0x40013800), %o0</td>
<td>[40013800]</td>
</tr>
<tr>
<td>120828294</td>
<td>400096c4</td>
<td>ldd [%o0 + 0x220], %f2</td>
<td>[3ff00000 00000000]</td>
</tr>
<tr>
<td>120828304</td>
<td>400096c8</td>
<td>fcmped %f0, %f2</td>
<td>[3ff00000]</td>
</tr>
<tr>
<td>120828314</td>
<td>400096cc</td>
<td>nop</td>
<td>[00000000]</td>
</tr>
<tr>
<td>120828315</td>
<td>400096d0</td>
<td>fbule 0x40009754</td>
<td>[00000000]</td>
</tr>
<tr>
<td>120828316</td>
<td>400096d4</td>
<td>sethi %hi(0x40013800), %o0</td>
<td>[40013800]</td>
</tr>
<tr>
<td>120828320</td>
<td>40009754</td>
<td>ldd [%fp - 0x38], %f0</td>
<td>[bfe8ab1d 4daa6a20]</td>
</tr>
<tr>
<td>120828325</td>
<td>40009758</td>
<td>ret</td>
<td>[40009758]</td>
</tr>
<tr>
<td>120828328</td>
<td>4000975c</td>
<td>restore</td>
<td>[00000000]</td>
</tr>
<tr>
<td>120828337</td>
<td>40004578</td>
<td>ba,a 0x400045d4</td>
<td></td>
</tr>
</tbody>
</table>
AMBA AHB trace

<table>
<thead>
<tr>
<th>time</th>
<th>address</th>
<th>type</th>
<th>data</th>
<th>trans</th>
<th>size</th>
<th>burst</th>
<th>mst</th>
<th>lock</th>
<th>resp</th>
</tr>
</thead>
<tbody>
<tr>
<td>120828324</td>
<td>40004578</td>
<td>read</td>
<td>30800017</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828326</td>
<td>4000457c</td>
<td>read</td>
<td>d21221b8</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828330</td>
<td>90000000</td>
<td>write</td>
<td>000045f9</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828334</td>
<td>400045d4</td>
<td>read</td>
<td>81c7e008</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828336</td>
<td>400045d8</td>
<td>read</td>
<td>81e80000</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828338</td>
<td>400045dc</td>
<td>read</td>
<td>9de3bf90</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828344</td>
<td>40006c08</td>
<td>read</td>
<td>c13fbfd0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828346</td>
<td>40006c0c</td>
<td>read</td>
<td>40000928</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>120828349</td>
<td>90000000</td>
<td>read</td>
<td>000055f9</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Mixed instruction/AHB trace

<table>
<thead>
<tr>
<th>TIME</th>
<th>ADDRESS</th>
<th>OPERATION</th>
<th>ADDRESS/DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>120828317</td>
<td>40009754</td>
<td>ahb read, mst=0, size=2</td>
<td>[4000975c 81e80000]</td>
</tr>
<tr>
<td>120828320</td>
<td>40009754</td>
<td>ldd [%fp - 0x38], %f0</td>
<td>[bfe8ab1d 4daa6a20]</td>
</tr>
<tr>
<td>120828324</td>
<td></td>
<td>ahb read, mst=0, size=2</td>
<td>[40004578 30800017]</td>
</tr>
<tr>
<td>120828325</td>
<td>40009758</td>
<td>ret</td>
<td>[40009758]</td>
</tr>
<tr>
<td>120828326</td>
<td></td>
<td>ahb read, mst=0, size=2</td>
<td>[4000457c d21221b8]</td>
</tr>
<tr>
<td>120828328</td>
<td>4000975c</td>
<td>restore</td>
<td>[00000000]</td>
</tr>
<tr>
<td>120828330</td>
<td></td>
<td>ahb write, mst=1, size=2</td>
<td>[90000000 000045f9]</td>
</tr>
<tr>
<td>120828334</td>
<td></td>
<td>ahb read, mst=0, size=2</td>
<td>[400045d4 81c7e008]</td>
</tr>
<tr>
<td>120828336</td>
<td></td>
<td>ahb read, mst=0, size=2</td>
<td>[400045d8 81e80000]</td>
</tr>
<tr>
<td>120828337</td>
<td>40004578</td>
<td>ba,a 0x400045d4</td>
<td>[00000000]</td>
</tr>
</tbody>
</table>
LEON3 Multi-Processor support

- LEON3 core + caches = 1.5 mm² on 0.13 process,
- Multi-processor systems feasible without area problems
- LEON3 MP support:
  - Processor enumeration
  - Write-through caches, snooping on data cache
  - Multi-processor DSU and interrupt controller
- AMBA: Round-robin AHB arbiter for fair bus utilization
- Asymmetric configuration possible
- 4-processor system fits on XC2V3000 FPGA @ 80 MHz
LEON3 Multi-Processor arch.
ESA/GR LEON3FT GINA processor
LEON3 Software Tools

- LEON3 is supported by a number of open-source kernels:
  - eCos, RTEMS, Pthreads, uCLinux, Linux-2.6
  - GCC-3.4.5 and associated tools
  - Debugging is based on GDB/DDD and GRMON
  - Eclipse IDE available for RTEMS and Pthreads
- LEON3 VxWorks 5.4 & 6.3 BSP
- GRMON hardware debug monitor for all kernels/compilers
- TSIM behavioral simulator (1 CPU, 20 MIPS)
- GRSIM behavioral simulator (multi-CPU, 10 MIPS)
- Windows and LINUX hosts
GRMON debug monitor

- Debug monitor with drivers for each specific IP core
- Loadable modules allow IP vendors to provide own drivers
- Communicates with target hardware through many interfaces: serial, JTAG, Ethernet, PCI, USB, Spacewire
- Can display all on-chip memory and registers
- Handles the LEON3 trace buffers
- ICE-like debug control without additional hardware
- Can be connected to GDB for source-level debugging
TSIM LEON3 Simulator

- Emulates one LEON3 processor, all on-chip peripherals + any amount of memory (PROM, SRAM, SDRAM).
- Processor can be configured with respect to cache size and organization, FPU and MUL/DIV options.
- Includes symbol handling and built-in trace buffer
- High performance: 15 MIPS+
- GDB interface
- Loadable modules to model any additional on- or off-chip functionality
- Library version for integration into larger frameworks
- New in 2.0.7: emulation of MMU
Sample LEON implementations

- Nemerix NJ1030 GPS receiver (CH)
- Infrant IT3102/7 Network storage processor (US)
- G2 Microsystems G2C501 Wi-Fi asset tracker
- Atmel AT697 embedded controller (FR, rad-hard)
- AGGA-3 Galileo/GPS receiver (D)
- AiSeek AI accelerator (Israel)
- Orbita S698 embedded controller (China)
- MPEG4 decoder (France)
- Printer engine (Aus.)
- Digital Radio (NZ)
- Aeroflex UT699 rad-hard controller
Sample LEON Layouts

Atmel AT697

A-STAR

NJ1030

LEON2FT UMC 0.18

LEON3FT IHP 0.25
Processor projects in 2007

- LEON3 with physical cache snooping
  - Duplication of tags (physical/virtual), 4 Kbyte way size
  - Porting of linux-2.6 SMP
- Integration of Sun Niagara (S1) into GRLIB framework
  - S1/AHB bridge + software drivers for Linux
- Integration of Xilinx V2/V4 PPC405 into GRLIB
  - PPC/AHB bridge with endian conversion
  - Software drivers for RTEMS, eCos and Linux
- Integration of Microblaze (maybe) ...
IP core projects in 2007

- USB-2.0 host controller
  - Being verified, release Q1 2007 (GPL)
- USB-2.0 device controller release in GPL
- IDE controller with DMA33 (Q1-2007)
- Virtex-5 port of GRLIB tech mapping
- DDR2 controller for ML501 board
Gaisler Research has received funding to develop LEON4, for both Aerospace and commercial applications:

- Non-blocking pipeline
- Instruction streaming buffer
- FPU controller with instruction FIFO
- DMA in data cache controller
- 64-bit AHB interface
- Dual issue pipeline (IU/FPU instruction)
- Clock gating to preserve power
LEON3 Development boards

- LEON3/GRLIB system can easily be prototype on many low-cost FPGA boards:
  - Spartan3-1500 from PENDER ($895)
  - Virtex4-LX25 from Xilinx ($495)
  - Cyclone2/Stratix2 from Altera ($695)
  - LatticeEC-C33 from Gleichmann ($450)
  - Digilent XUP or Spartan3-1600E
  - Many others ...

- Supported by free tools: ISE Webpack, Quartus Web Edition
- 30+ template designs with FPGA config files freely available
Further information

- LEON3 information and GRLIB source code is at:
  - http://www.gaisler.com
- Suitable FPGA boards:
  - http://www.pender.ch
  - http://www.ger-fae.com/
  - http://www.digilentinc.com/