DEVELOPING RTEMS SMP FOR LEON3/LEON4 MULTI-PROCESSOR DEVICES

Flight Software Workshop 2013
12/12/2013

Daniel Hellström
www.aeroflex.com/Gaisler

Presentation does not contain U.S. Export controlled information (aka ITAR)
Overview

- ESA Activity – “Development Environment for Future LEON Multi-core”

- Collaboration between
  - Aeroflex Gaisler
  - OAR
  - ASTRIUM

- Background
  - Hardware
  - From an ESA point of view

- Activity Overview

- RTEMS SMP development

- Parallel Programming Models

- Demonstrator
Processor evolution - European perspective

Processor evolution:

- 1989: MDC281 (MIL-STD-1750) clone, 2.5 um CMOS/SOS, 0.5 MIPS
- 1991: MA31750 (MIL-STD-1750), 1.5 um CMOS/SOS, 2 MIPS
- 1990-92: SPARC V7 architecture selected as ESA baseline
- 1995: Three-chip ERC32 (TCS691/2/3), 10 MIPS
- 1997: Single-chip ERC32 (TSC695), 15 MIPS
- 1997: SPARC V8 LEON VHDL development begins
- 2000: First LEON1FT, 0.35 um, 50 MIPS
- 2002: First LEON2FT, 0.18 um, 100 MIPS
- 2004: First LEON3FT, 0.20 um, 125 MIPS
- 2009: First dual-core LEON3FT, 180 nm, 250 MIPS
- 2010: First LEON4
- 2012: First quad-core LEON4FT, 45 nm
LEON multi-core architecture evolution

- LEON architecture evolution from a multi-core view:
  - First LEON3 multi-core designs
  - Improved L1 D-cache snooping
  - CAS instruction defined by SPARCv9
  - Ability to share one GRFPU between two CPUs
  - GR712RC first dual-core LEON3FT
  - Improved bus architecture for better multi-master bus utilization
  - Wider AMBA bus 64-bit/128-bit
  - Introduction of L2-cache
  - Improved IRQ Controller
  - I/O-MMU for protection against faulty DMA
  - Improvements in other IP-Cores to support multi-core, such as GRPCI2
  - Other improvements..
  - NGMP - quad-core LEON4FT
GR712RC dual-core LEON3FT processor

- 2x LEON3FT fault-tolerant processor
  - 32 KiB cache with 4 parity bits per word
  - 7-bit SEC/DED BCH EDAC on register file
  - SPARC SRMMU with 32 TLB entries
  - Separate GRFPU per LEON3FT core
  - Branch prediction
  - SPARC V9 CAS instruction
  - On-the-fly error correction of store data
  - Inactive CPU/FPU automatically clock-gated

- On-chip SRAM, 192 KiB with 7-bit SEC/DED BCH EDAC

- 8/32-bit PROM/SRAM/SDRAM controller with BCH and Reed-Solomon EDAC
- 6x GRSPW2 SpaceWire Nodes with RMAP (200 Mbps)
- Redundant 1553B BC/RT/MT
- 2x CAN-2.0B
- 10/100 Mbit Ethernet MAC (RMII PHY interface)
- I2C, SPI (suitable for legacy interfaces)
- CCSDC TC Decoder (10 Mbps)
- CCSDC TM Encoder (50 Mbps)
GR712RC boards – EM and prototype

- USB/JTAG Debug interface
- 10/100 Ethernet
- 6x SpW
- 8 Mbyte SRAM
- Access to all I/O
- 8 Mbyte FLASH
- 256 Mbyte SDRAM
- 2xCAN
- 2x RS232
- MIL-1553 A/B
- T0/100 Ethernet
- USB/JTAG Debug interface
LEON4/4FT – going further

▼ LEON4 processor core
- In January 2010, the fourth version of the LEON processor was released
- Static branch prediction added to pipeline
- 64-bit or 128-bit path to AMBA AHB interface
- Optional level-2 cache
- Higher performance possible (1.7 DMIPS/MHz)
- Tuned to higher clocking rates
- Quad-core/processor solutions feasible

▼ LEON4FT processor core
- Fault-tolerant version of the standard LEON4
- New system-on-chip architecture developed in parallel:
  ▼ Time-and-space partitioning
  ▼ Separation between processor and peripheral buses
  ▼ IOMMU
- Major impact on peripheral IP cores which were updated to support new architecture
LEON4 system-on-chip architecture

- GRFPU
- LEON4FT
- L1 Cache

CPU bus 128-bit

- DMA Masters
- IOMMU

Level 2 Cache

Memory bus 128-bit

- Scrubber

EDAC

Memory controller

- DDR2
- SDRAM

to PROM, SRAM, PCI and low-speed slaves...
Next Generation Microprocessor (NGMP)

- Quad-core LEON4FT with two shared FPUs
- 128-bit L1 caches connected to 128-bit AHB bus
- 256+ KiB L2 cache, 256-bit cache line, 4-ways
- 64-bit DDR2-800/SDR-PC100 SDRAM memory I/F
- 32 MiB on-chip DRAM (if feasible)
- 8-port SpaceWire router with 4 internal ports

- 32-bit, 66 MHz PCI interface
- 2x 10/100/1000 Mbit Ethernet
- 4x High-Speed Serial Links
- Debug links: Ethernet, JTAG, USB, SpaceWire RMAP target
- 16x GPIO, SPI master/slave, MIL-STD-1553B, 2x UART

www.aeroflex.com/Gaisler
NGMP performance and status

- On ESA’s microprocessor roadmap
- Designed for DSM CMOS technology
- Radiation hard library targeted
- IEEE-1754 SPARC V8 compliant 32-bit processor
  - 7-stage pipeline, multi-processor support
  - Separate multi-set L1 caches with LRU/LRR/RND
  - 64-bit single-clock load/store operation
  - 64-bit register file with BCH
  - 128-bit AHB bus interface
  - Write combining in store buffer
  - Branch prediction
  - CAS support
  - Performance counters
  - On-chip debug support unit with trace buffer
- Software compatible with LEON2FT/LEON3FT
- Compliant with SPARC V8
- Prototypes available based on Xilinx FPGA boards
- Functional Prototype in progress based on eASIC Nextreme-2
- Functional Prototype evaluation board available since 2012

Performance:
- 1.7 DMIPS/MHz
- 0.6 Whetstone MFLOPS/MHz
- 0.35 SPECINT/MHz,
- 0.25 SPECFP/MHz
- 2.1 CoreMark/MHz (comparable to ARM11)
ESA Activity - Background

- ESA has been driving LEON and now NGMP development forward

- Several studies/workshops been performed on how to use multi-core and NGMP in space applications. For example
  - SIDMS
  - Multi-core OS Benchmark
  - ADCSS

- Earlier work shows that space applications on multi-core systems needs
  - SMP RTOS
  - Analysis tools
  - Parallelization help
ESA Activity – Overview

▼ Started Q3 2013 – work in progress

▼ Task1 – RTEMS SMP development
- Analyze existing RTEMS SMP
- Extend RTEMS SMP
- Test and Verify on target hardware: GR712RC and NGMP
- Commit changes to RTEMS mainline

▼ Task2 – Parallelization library/techniques support
- Analyze existing MP Parallelization libraries/techniques
- Highlight libraries suitable for LEON space applications
- Select and port libraries to RTEMS SMP on LEON targets

▼ Task3 – Demonstrator application
- Port existing space application
- Parallelize application to support LEON multi-core targets on the RTEMS SMP OS
- Use above software - selected library(ies) and RTEMS
- Demonstrate and performance measurements
ESA Activity – Collaboration

▼ Task1 – RTEMS SMP development
– OAR: RTEMS SMP extensions
– Aeroflex Gaisler: SPARC/LEON architectural parts

▼ Task2 – Parallelization library support
– ASTRIUM: Analyze existing MP Parallelization libraries/techniques
– Aeroflex Gaisler: Implement Library(ies) for RTEMS SMP on LEON targets

▼ Task3 – Demonstrator application
– ASTRIUM
ESA Activity – Hardware targets

- GR712RC Development board
  - Dual-core LEON3FT

- GR-CPCI-LEON4-N2X (NGMP)
  - Quad-core LEON4
  - Functional-prototype at 200MHz
RTEMS SMP development

- Analyze existing RTEMS SMP – identify problems/solutions related to SMP support and LEON

- CPU Affinity support in RTEMS Scheduler
  - Pluggable Scheduler interface extensions
  - Extend Simple SMP Scheduler with affinity support
  - Introduce CPU Affinity User APIs to POSIX threads and classic RTEMS task

- Atomic operations layer
  - SPARC/LEON CAS instruction support
  - Use the atomic layer in code implementations

- Trace library with SMP support
  - Extend existing RTEMS capture engine
  - Add SMP support using lockless methods

- Port RTEMS SMP to LEON GR712RC and NGMP

- Support basic LEON peripherals such as IRQCtrl, Timer and UART for SMP

- Test and verification
RTEMS Scheduler CPU Affinity support

- **Scheduler Affinity**
  - Control which CPU a task executing on
  - Affinity is changable during run-time

- **Example use cases of Scheduler Affinity**
  - Porting single-core code, possible to mimick a uni-processor system for parts of the application
  - Avoiding SMP locking effects over common resources. Limit a particular resource to a specific CPU (non-preemption)
  - Performance and real-time performance. Fine-grained control over scheduling, L1-cache hit rate improvements
  - Load balancing
  - Debugging
Parallel Programming Models

- ASTRIMUM - Work in progress,
- Parallel programming techniques are being studied,
  - Cilk, Go
  - OpenMP
  - Message Passing Interface (MPI)

- Early analysis – MPI the way to go
  - Not compiler dependent
  - Full control of execution
  - Data transmissions
  - Synchronization
  - Hardware architecture independent
Demonstrator – GAIA VPU

- ASTRIUM – GAIA Video Processing Unit (VPU) application (launch December 2013)
- Evaluating RTEMS-SMP for multi-core platform projects
- Existing VxWorks single-core PowerPC application
- Port to RTEMS
- Parallelize application by using RTEMS SMP and parallelization software
- Support dual-core (GR712RC) and quad-core (NGMP)
- Existing application extensively validated and performance detailed instrumented which will prove to be a very good reference application for LEON multi-core design
Thank you all involved in the project and ESA!
Thank you for listening!

The End
GR712RC specifications

- Manufactured on Tower Jazz 180 nm CMOS technology
- RadSafe radiation hard library:
  - TID: 300 krad(Si)
  - SEL: LET > 118 MeV/cm²/mg at +125°C
  - SEU: Proven SEU tolerance (RADECS 2011 proceedings)
- Software compatible with LEON2FT
- Compliant with SPARC V8
- Performance:
  - 125 DMIPS/100 MFLOPS peak @ 100 MHz (per core)
  - 50 MFLOPS Whetstone @ 100 MHz (per core)
- Power consumption:
  - 1.5 W @ 100 MHz, one CPU/FPU under full load
  - 0.5 W @ 100 MHz, both CPUs in standby
- Supply voltages, 1.8V core, 3.3V I/O
- Temperature range –55°C to +125°C
- CQFP-240, 0.5 mm pitch, 32 mm x 32 mm
- Screening and qualification as per MIL-STD-883 Class Level S
- Samples and development boards available
- Qualified parts available since 2013
Working with open-source

- Work targets mainline development
- Set baseline
- Problems introduced
- What happens when someone else breaks the system
- Failure - fall back to branch from mainline for the activity
- Sources not accepted – important to develop according to RTEMS standards and common goals
- Initial development to fix existing problems
- Verification against
- Findings during the project, to be exported to public Wiki

- CPU Affinity support in RTEMS Scheduler
  - Pluggable Scheduler interface extensions
  - Extend Simple SMP Scheduler with affinity support
  - Introduce CPU Affinity User APIs to POSIX threads and classic RTEMS task

- Atomic operations layer
  - SPARC/LEON CAS instruction support
  - Use the atomic layer in code implementations
References

- ADCSS – ESA Workshop on Avionics Data, Control and Software Systems (ADCSS)
- Multicore OS benchmarks
- SIDMS – System Impact of Distributed Multicore Systems