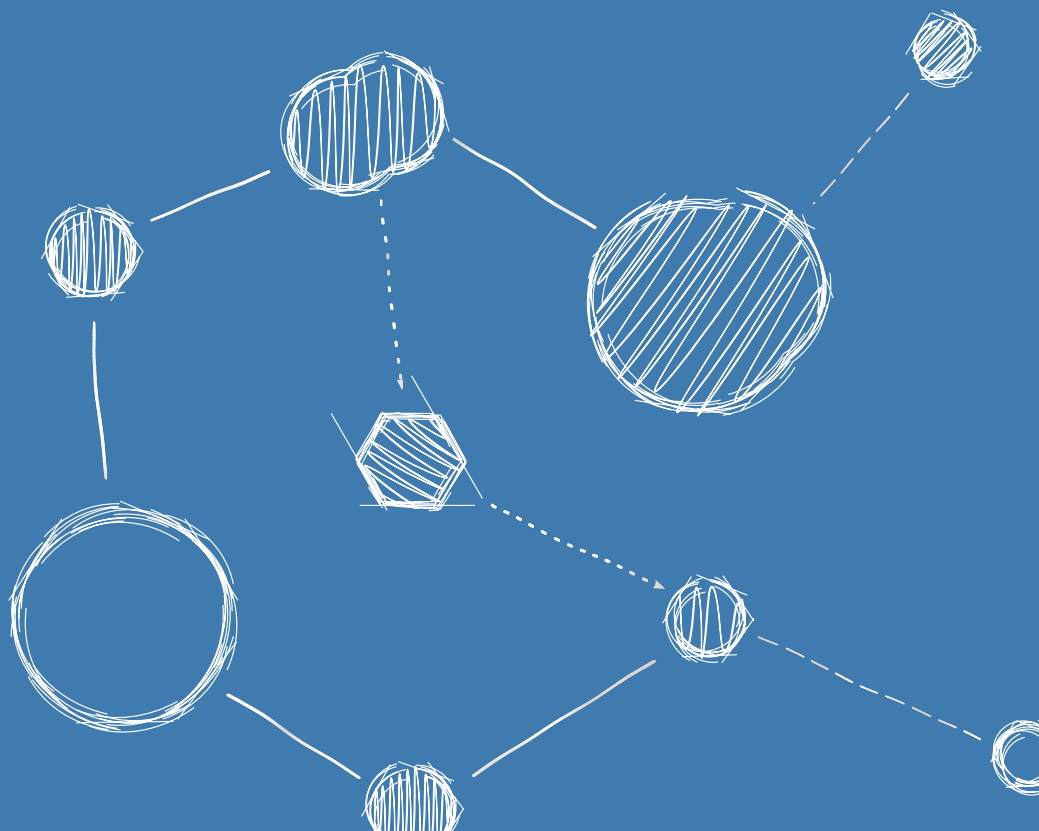


LEAN OS – AN OPERATING SYSTEM FOR THE SSDP

Test Report



Test Report

Reference: LEANOS-UVIE-TR-001

Version: Issue 1.0, June 1, 2017

Prepared by: Armin Luntzer¹

Checked by: Roland Ottensamer¹, Christian Reimers¹

Approved by: Franz Kerschbaum¹

¹ Department of Astrophysics, University of Vienna

Copyright ©2017

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Front-Cover, no Logos of the University of Vienna.

Contents

1	Introduction	4
1.1	Purpose of the Document	4
1.2	Incremental Testing	4
2	Applicable and Reference Documents	5
3	Terms, Definitions and Abbreviated Items	6
3.1	Acronyms	6
3.2	Glossary	6
4	Test Results	8
4.1	HWT-0001	8

Revision History

Revision	Date	Author(s)	Description
0.1	06.03.2017	AL	initial version
1.0	01.06.2017	FK	document approved

1. Introduction

1.1 Purpose of the Document

In this document, results of tests specified in [1] are recorded.

1.2 Incremental Testing

LeanOS is not envisaged to be implemented in one final form. Instead, it is marked with version numbers for each release. This means that testing is version-dependent and is applied in increments by executing all available tests as modifications are made. This is reflected in the dates, version numbers and cryptographic hash values of the *git* repository's HEAD listed in each individual test report.

2. Applicable and Reference Documents

- [1] *LeanOS Test Specification*. 2017.

3. Terms, Definitions and Abbreviated Items

3.1 Acronyms

CPU	Central Processing Unit
DSP	Digital Signal Processor
ILP	Instruction Level Parallelism
MPPB	Massively Parallel Processor Breadboarding system
NoC	Network On Chip
SoC	System On Chip
VLIW	Very Long Instruction Word

3.2 Glossary

Central Processing Unit (**CPU**)

The Central Processing Unit is the electronic circuitry that interprets instructions of a computer program and performs control logic, arithmetic, and input/output operations specified by the instructions. It maintains high-level control of peripheral components, such as memory and other devices.

Digital Signal Processor (**DSP**)

A Digital Signal Processor is a specialised processor with its architecture targeting the operational needs of digital signal processing.

Instruction Level Parallelism (**ILP**)

Instruction-level parallelism (ILP) is a measure of how many instructions in a computer program can be executed simultaneously by the [CPU](#).

LEON2

The LEON2 is a synthesisable VHDL model of a 32-bit processor compliant with the SPARC V8 architecture. It is highly configurable and particularly suitable for [System On Chip \(SoC\)](#) designs. Its source code is available under the GNU LGPL license

Massively Parallel Processor Breadboarding system (**MPPB**)

The Massively Parallel Processor Breadboarding system is a proof-of-concept design for a space-hardened, fault-tolerant multi-DSP system with various subsystems to build a powerful digital signal processing system with a high data throughput. Its distinguishing fea-

tures are the [Network On Chip \(Network On Chip \(NoC\)\)](#) and the [Xentium DSPs](#) controlled by a [LEON2](#) processor. It was developed under ESA contract 21986 by Recore Systems B.V.

Network On Chip (NoC)

A Network On Chip is a communication system on an integrated circuit that applies (packet based) networking to on-chip communication. It offers improvements over more conventional bus interconnects and is more scalable and power efficient in complex [System On Chip \(SoC\)](#) designs.

System On Chip (SoC)

A System On Chip is an integrated circuit that combines all components of a computer or other electronic system into a single chip.

Very Long Instruction Word (VLIW)

Very Long Instruction Word is a processor architecture design concept that exploits [Instruction Level Parallelism \(ILP\)](#). This approach allows higher performance at a smaller silicon footprint compared to serialised instruction processors, as no instruction re-ordering logic to exploit superscalar capabilities of the processor must be integrated on the chip, but requires either code to be tuned manually or a very sophisticated compiler to exploit the full potential of the processor.

Xentium

The Xentium is a high performance [Very Long Instruction Word \(VLIW\) DSP](#) core. It operates 10 parallel execution slots supporting 32/40 bit scalar and two 16-bit element vector operations.

4. Test Results

4.1 HWT-0001

Date May 24th, 2017

Version 0.1

Hash 1077b932fd894ec212aa45741aacacf8ce808ab3

Test Description

This test verifies the implementation of the [Xentium](#) Processing Network implementation on the [MPPB](#) by generating a series of processing tasks and running them through the network. This test also represents an integrated test of LeanOS.

Overview of the Procedure

This test is executed on a [MPPB](#) v2.x hardware model. Once the test environment is set up, the build configuration is prepared and the test itself is executed and evaluated automatically by starting the *bash* script *run_test.sh*. It may be found in the *tools/testing/hwtests/xen_proc_net* subdirectory of the LeanOS source tree.

Test Setup

The following preparations must be made in order to carry out the test:

- checkout of the LeanOS source tree
- [MPPB/Xentium](#) toolchain location in shell *PATH*
- *grmon* ver. 1.x configured and in shell *PATH*
- [MPPB](#) v2.x connected to PC via DEBUG UART and powered on

Test Log

```
./run_test.sh
CLEAN    .tmp_versions
HOSTCC   scripts/basic/fixdep
HOSTCC   scripts/kconfig/conf.o
HOSTCC   scripts/kconfig/zconf.tab.o
HOSTLD   scripts/kconfig/conf
```

```
scripts/kconfig/conf --silentoldconfig Kconfig
CHK      include/generated/version.h
CC       scripts/mod/empty.o
HOSTCC   scripts/mod/mk_elfconfig
MKELF    scripts/mod/elfconfig.h
HOSTCC   scripts/mod/modpost.o
HOSTLD   scripts/mod/modpost
CC       init/main.o
CC       init/xentium_demo.o
CC       init/modules-image.o
LD       init/built-in.o
CC       arch/sparc/kernel/setup.o
CC       arch/sparc/kernel/init.o
CC       arch/sparc/kernel/page.o
CC       arch/sparc/kernel/bootmem.o
CC       arch/sparc/kernel/mm.o
CC       arch/sparc/kernel/elf.o
CC       arch/sparc/kernel/module.o
CC       arch/sparc/kernel/traps.o
CC       arch/sparc/kernel/stacktrace.o
CC       arch/sparc/kernel/stack.o
AS       arch/sparc/kernel/traps/data_access_exception_trap.o
CC       arch/sparc/kernel/traps/data_access_exception.o
CC       arch/sparc/kernel/irq.o
LD       arch/sparc/kernel/built-in.o
LD       arch/sparc/mm/built-in.o
LD       arch/sparc/built-in.o
CC       kernel/kmem.o
CC       kernel/ksym.o
CC       kernel/printk.o
CC       kernel/bitmap.o
CC       kernel/module.o
CC       kernel/noc_dma.o
CC       kernel/irq.o
CC       kernel/xentium.o
LD       kernel/built-in.o
LD       lib/built-in.o
CC       lib/ar.o
CC       lib/chunk.o
CC       lib/data_proc_net.o
CC       lib/data_proc_task.o
CC       lib/data_proc_tracker.o
```

```
CC      lib/elf.o
CC      lib/mm.o
CC      lib/page.o
CC      lib/string.o
AR      lib/lib.a
LD      dsp/xentium/built-in.o
HOSTCC  dsp/xentium/kernel/dummy/xen_dummy.o
HOSTCC  dsp/xentium/lib/xen_printf.o
HOSTCC  dsp/xentium/lib/xen.o
HOSTCC  dsp/xentium/lib/dma.o
HOSTCC  dsp/xentium/lib/kmem.o
HOSTCC  dsp/xentium/../../lib/data_proc_task.o
HOSTLD  dsp/xentium/xen_dummy.xen
HOSTCC  dsp/xentium/kernel/rampfit/xen_rampfit.o
HOSTLD  dsp/xentium/xen_rampfit.xen
HOSTCC  dsp/xentium/kernel/deglitch/xen_deglitch.o
HOSTLD  dsp/xentium/xen_deglitch.xen
HOSTCC  dsp/xentium/kernel/stack/xen_stack.o
HOSTLD  dsp/xentium/xen_stack.xen
LD      dsp/built-in.o
LD      samples/noc_dma/built-in.o
HOSTCC  samples/noc_dma/noc_dma_demo.o
HOSTLD  samples/noc_dma/noc_dma_demo
LD      samples/proc_chain/built-in.o
HOSTCC  samples/proc_chain/proc_chain_demo.o
HOSTLD  samples/proc_chain/proc_chain_demo
LD      samples/built-in.o
LD      leanos.o
make[1]: Nothing to be done for 'leanos.o'.
GEN      .version
KSYM     .tmp_kallsyms1.o
KSYM     .tmp_kallsyms2.o
LD      leanos
SYSMAP   System.map
Building modules, stage 2.
MODPOST 0 modules
Embedding module image, stage 3.
LD      leanos.o
make[1]: Nothing to be done for 'leanos.o'.
GEN      .version
KSYM     .tmp_kallsyms1.o
KSYM     .tmp_kallsyms2.o
```

LD leanos
SYSMAP System.map

GRMON LEON debug monitor v1.1.61 professional version

Copyright (C) 2004-2011 Aeroflex Gaisler - 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: 50 MHz

Component	Vendor
LEON2 Memory Controller	European Space Agency
LEON2 AHB Status & Failing Addr	European Space Agency
LEON2 SPARC V8 processor	European Space Agency
LEON2 Write Protection	European Space Agency
LEON2 Configuration register	European Space Agency
LEON2 Timer Unit	European Space Agency
LEON2 UART	European Space Agency
LEON2 UART	European Space Agency
LEON2 Interrupt Ctrl	European Space Agency
LEON2 I/O port	European Space Agency
AHB Debug UART	Gaisler Research
LEON2 Debug Support Unit	Gaisler Research

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

```
grlib> load leanos
section: .text at 0x40000000, size 96592 bytes
downloading: 96592
section: .data at 0x40017950, size 81572 bytes
downloading: 81572
total size: 178164 bytes (88.9 kbit/s)
read 394 symbols
entry point: 0x40000000
grlib> run
MAIN: Loading module image
MAIN: Looked up symbol printk at 0x400032e0
MAIN: Looked up file noc_dma.ko at 0x0
```

```
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT:      19740352
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT:      19996224
XEN_OUT:      349184
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT: releasing empty task
XEN_OUT:      19996224
XEN_OUT:      349184
Xentium demo SUCCESSFUL

Program exited normally.
grib> quit
./run_test.sh  9.92s user 0.41s system 35% cpu 28.921 total
```

Test Results

The test procedure was executed **successfully**.