SANDIA REPORT

SAND2011-2733

Unlimited Release

Printed April 2011

Evaluation of Soft-Core Processors on a Xilinx Virtex-5 Field Programmable Gate Array

Mark W. Learn

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>

Evaluation of Soft-Core Processors on a Xilinx Virtex-5 Field Programmable Gate Array

Mark W. Learn
Sensors & Embedded Systems, 2623
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-0661

Abstract

Node-based architecture (NBA) designs for future satellite projects hold the promise of decreasing system development time and costs, size, weight, and power and positioning the laboratory to address other emerging mission opportunities quickly. Reconfigurable field programmable gate array (FPGA)-based modules will comprise the core of several of the NBA nodes. Microprocessing capabilities will be necessary with varying degrees of mission-specific performance requirements on these nodes. To enable the flexibility of these reconfigurable nodes, it is advantageous to incorporate the microprocessor into the FPGA itself, either as a hard-core processor built into the FPGA or as a soft-core processor built out of FPGA elements. This document describes the evaluation of three reconfigurable FPGA-based soft-core processors for use in future NBA systems: the MicroBlaze (uB), the open-source Leon3, and the licensed Leon3. Two standard performance benchmark applications were developed for each processor. The first, Dhrystone, is a fixed-point operation metric. The second, Whetstone, is a floating-point operation metric. Several trials were run at varying code locations, loop counts, processor speeds, and cache configurations. FPGA resource utilization was recorded for each configuration.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	9
1. GENERAL	11
1.1 Scope	11
1.2 Summary	11
1.3 Document Structure	12
1.4 References	12
1.4.1 Project References	12
2. BACKGROUND	13
2.1 MicroBlaze Soft-Core Processor	13
2.2 Leon3 Soft-Core Processor	14
2.2.1 Open-Source Core	14
2.2.2 Licensed Core	14
3. TEST SETUP	16
3.1 MicroBlaze	16
3.1.1 Hardware	16
3.1.2 Software	17
3.2 Leon3	18
3.2.1 Open-Source Core	18
3.2.2 Leon3FT Core	20
4. RESULTS	23
4.1 Virtex-5 Dhrystone	23
4.2 Virtex-5 Whetstone	25
4.3 Virtex-5 Resources	26
4.3.1 MicroBlaze	27
4.3.2 Open-Source Leon3	28
4.3.3 Leon3FT	28
5. CONCLUSION	30
APPENDIX A. PPC440 Processor	31

FIGURES

Figure 1. MicroBlaze core block diagram.	13
Figure 2. Leon3 core block diagram.	14
Figure 3. MicroBlaze block diagram.	16
Figure 4. Leon3 block diagram.	18
Figure 5. Leon3FT block diagram.	20
Figure 6. Processor compiler optimizations.	23
Figure 7. Virtex-5 Dhrystone comparison.	24
Figure 8. Virtex-5 Whetstone comparison.	25
Figure 9. Processor resource utilization.	26
Figure 10. MicroBlaze resource utilization.	27
Figure 11. Leon3 resource utilization.	28
Figure 12. Leon3FT resource utilization.	29

TABLES

Table 1. External Reference Documents.	12
Table 2. Internal Reference Documents.	12

ACRONYMS

AHB	AMBA high-speed bus
APB	AMBA peripheral bus
BRAM	block random-access memory
BSB	Base System Builder
DCM	digital clock manager
DDR	double data rate
EDAC	error detection and correction
FPGA	field programmable gate array
FPU	floating-point unit
FT	fault-tolerant
IP	intellectual property
IU	integer unit
LMB	local memory bus
LRR	least recently replaced
LRU	least recently used
LUT	lookup table
MCH_EMC	multi-channel external memory controller
MMU	memory management unit
MPMC	multi-port memory controller
NBA	node-based architecture
OTP	one-time programmable
PLB	processor local bus
RAM	random-access memory
RISC	reduced instruction set computer
SDRAM	synchronous dynamic random-access memory
SEU	single-event upset
SRAM	static random-access memory
uB	MicroBlaze
XPS	Xilinx Platform Studio

EXECUTIVE SUMMARY

Node-based architecture (NBA) designs for future satellite projects hold the promise of decreasing system development time and costs, size, weight, and power and positioning the laboratory to address other emerging mission opportunities quickly.

Reconfigurable field programmable gate array (FPGA)-based modules will comprise the core of several of the NBA nodes. Microprocessing capabilities will be necessary with varying degrees of mission-specific performance requirements on these nodes. To enable the flexibility of these reconfigurable nodes, it is advantageous to incorporate the microprocessor into the FPGA itself.

Three reconfigurable FPGA-based soft-core processors for use in future NBA systems were evaluated on the Xilinx Virtex-5 FX-70T FPGA: the MicroBlaze (uB), the open-source Leon3, and the licensed Leon3. The licensed core is also referenced as the Leon3FT because it was designed to be fault-tolerant within an Actel RTAX FPGA. The licensed Leon3 used within this evaluation was not intended to be fault-tolerant in a Xilinx device, but to use as a prototype and develop with an equivalent netlist on a reconfigurable platform rather than with the complications of developing on a one-time programmable (OTP) platform.

To evaluate the different soft-core processors, two performance benchmark applications were used. The first application used was the Dhrystone benchmark, which is used for fixed-point operation metrics. The second application used was the Whetstone benchmark, which is used for single-precision floating-point operation metrics. Several trials were run at varying code locations, loop counts, processor speeds, and cache configurations. The FPGA resource utilization for each processor was also recorded for each configuration.

The results show that enabling the cache and the floating-point unit (FPU) greatly improves processor performance. With the cache disabled, though, the Leon was more efficient than the uB while running out of external memory, but the performance was still low. With cache enabled, the performance on both processors increased by about a factor of 20. With the FPU enabled, the uB performance increased by a factor of 40, while the Leon's performance increased by a factor of about 400 to 1000.

The Leon3 uses more resources than the uB when utilizing the FPU. The FPU used within the Leon3 is designed to be a more capable hardware instantiation than the uB FPU; thus, the extra resources needed. For example, the Leon3 FPU can handle double-precision floating-point operations whereas the uB FPU cannot. With only the cache enabled, each processor utilized about the same number of resources. With the cache and FPU disabled, both processors used about the same number of resources.

1. GENERAL

1.1 Scope

The purpose of this document is to provide an evaluation of soft core processors within a Xilinx Virtex-5 FX field programmable gate array (FPGA). The intent of this report is to provide a baseline comparison on these soft-core processors without making modifications to the processors themselves and only using the tools and netlists provided. The performance and capability of each processor and the trade-offs that each provides is discussed within this document.

1.2 Summary

Node-based architecture (NBA) designs for future satellite projects hold the promise of decreasing system development time and costs, size, weight, and power and positioning the laboratory to address other emerging mission opportunities quickly. FPGA-based modules will comprise the core of several of the NBA nodes. Microprocessing capabilities will be necessary with varying degrees of mission-specific performance requirements on these nodes. To enable the flexibility of these reconfigurable nodes, it is advantageous to incorporate the microprocessor into the FPGA itself. Three reconfigurable FPGA-based soft-core processors for use in future NBA systems were evaluated on the Xilinx Virtex-5 FX-70T FPGA; the MicroBlaze (uB), the open-source Leon3, and the licensed Leon3. The licensed core is also referenced as the Leon3FT because it was designed to be fault-tolerant within an Actel RTAX FPGA. The licensed Leon3 used within this evaluation was not intended to be fault-tolerant in a Xilinx device, but to use as a prototype and develop with an equivalent netlist on a reconfigurable platform rather than with the complications of developing on an OTP platform.

To evaluate the different soft-core processors, two performance benchmark applications were used. The first application used was the Dhrystone benchmark, which is used for fixed-point operation metrics. The second application used was the Whetstone benchmark, which is used for single-precision floating-point operation metrics. Each processor was tested under varying configurations such as enabling or disabling the cache. The FPGA resource utilization for each processor was also recorded for each configuration.

The results show that enabling the cache and the floating-point unit (FPU) greatly improves processor performance. With the cache disabled, though, the Leon was more efficient than the uB while running out of external memory, but the performance was still low. With cache enabled, the performance on both processors increased by about a factor of 20. With the FPU enabled, the uB performance increased by a factor of 40, while the Leon's performance increased by a factor of about 400 to 1000.

The Leon3 uses more resources than the uB when utilizing the FPU. The FPU used within the Leon3 is designed to be a more capable hardware instantiation than the uB FPU; thus, the extra resources needed. For example, the Leon3 FPU can handle double-precision floating-point operations whereas the uB FPU cannot. With only the cache enabled, each processor utilized about the same number of resources. With the cache and FPU disabled, both processors used about the same number of resources.

1.3 Document Structure

This document contains four major sections, with a brief description of each given below:

1. **General Information** – Outlines the purpose, summary, and contents of the evaluation study.
2. **Background** – Provides a brief description of the processors used in this study.
3. **Test Setup** – Describes the hardware and software test setup used for evaluating the soft-core processors.
4. **Results** – Provides the results of the evaluations of the soft-core processors.

1.4 References

1.4.1 Project References¹

The references identified in Table 1 were used in preparation of this document.

Table 1. External Reference Documents.

Document Number	Document Title
1.0.21	GRLIB IP Core User's Manual
1.0.21	GRLIB FT-FPGA User's Manual
1.0.21	GRLIB IP Library User's Manual
1.1.36	GRMON User's Manual
UG081 (v9.3)	MicroBlaze Processor Reference Guide
	www.gaisler.com
DS202 (v5.3)	Virtex-5 FPGA Data Sheet: DC and Switching Characteristics

Table 2. Internal Reference Documents.

Document Number	Document Title
	Leon Hardware Build
	Leon Software Build
SAND2008-6015	Soft-Core Processor Study for Node-Based Architectures

¹ Documents used are the latest version, unless otherwise specified

2. BACKGROUND

2.1 MicroBlaze Soft-Core Processor

The uB soft-core processor is a 32-bit reduced instruction set computer (RISC) designed for a Xilinx FPGA and is represented by the block diagram shown in Figure 1.

Figure 1. MicroBlaze core block diagram.

The uB Version 7.10.d was used for this test and allows selective enabling of additional functionality within the processor.

2.2 Leon3 Soft-Core Processor

Two versions of the Leon3 soft-core processor were used in this study: the open-source processor and the licensed processor. Both of these processors are described in the following sections.

2.2.1 Open-Source Core

The open-source Leon3 soft-core processor is a 32-bit processor core conforming to the IEEE-1754 (SPARC V8) architecture. The block diagram of the Leon3 is shown in Figure 2.

Figure 2. Leon3 core block diagram.

The Leon3 is customizable to generate a smaller or faster implementation. Note that Gaisler provides a high-performance FPU (GRFPU) and a light FPU (GRFPU-Lite), which uses fewer resources and has a lower performance than the GRFPU. The GRFPU-Lite is not pipelined and executes one floating-point operation at a time. Both versions of the FPU support double-precision floating point operations. The GRFPU was used for this study and only performed single-precision floating point operations using the Whetstone benchmark.

2.2.2 Licensed Core

The licensed core is also referenced as the Leon3FT because it was designed to be fault-tolerant (FT) within an Actel RTAX FPGA. The licensed Leon3 will be referenced as Leon3FT throughout the remainder of this paper. The Leon3FT used within this evaluation was not intended to be fault-tolerant in a Xilinx device, but to use as a prototype and develop with an equivalent netlist on a reconfigurable platform rather than with the complications of developing on an OTP platform. Further testing of the fault-tolerant aspects of the Leon3FT on a Xilinx FPGA will be tested in upcoming radiation tests.

The Leon3FT core is similar to the open-source version, but is built to reduce the number of single-event upsets (SEUs) within the processor when implemented in an Actel RTAX. The Leon3FT core focuses on protection of on-chip RAM used for the integer unit (IU), FPU register file, and the cache memory. Configuring the Leon3FT is not as flexible as the open-source version since a netlist is provided only by Gaisler.

Generally the IU is configurable and can be protected by four different implementations in an Actel RTAX. These are:

- Hardened flip-flops with no error checking
- 4-bit checksum per 32-bit word. Detect and corrects 1 bit per byte (4 bits per word). The pipeline is restarted on correction.
- 8-bit checksum per 32-bit word. Detected and corrects 1 bit per byte (4 bits per word). Correction is “on the fly” without pipeline restart.
- 7-bit BCH checksum per 32-bit word. Detects 2 bits and corrects 1 bit per word. The pipeline is restarted on correction.

The netlist provided by Gaisler configured the IU as an 8-bit parity protection.

The FPU is not configurable and is always protected with an 8-bit parity without pipeline restart. If the GFPU-Lite is used, then 4-bit parity is utilized.

The cache uses 4-bit parity for the tag and data. If there is an error during a cache access, then the cache line will get flushed and the failed instruction will be re-executed.

Note that within the Xilinx radiation hardened by design Virtex-5 QVR FPGA, flip-flops are also hardened, as they are in the Actel RTAX.

2.2.2.1 Fault-Tolerant Intellectual Property

Gaisler also provides fault-tolerant intellectual property (IP) to help protect against SEUs. As stated previously, these cores are referred to as fault-tolerant (FT) to distinguish between their open-source counterparts. These IPs were designed for use within RTAX devices and a netlist is provided for designing on reconfigurable devices.

Two fault-tolerant cores that are used during this testing are the AHBRAM and the external memory controller. Both are designed to use error detection and correction (EDAC) using a 32, 7-bit BCH code.

3. TEST SETUP

The following sections describe the test setup used for evaluating the soft-core processors. All hardware and software configurations were kept as similar as possible to provide comparable benchmark scores. The Xilinx ML507 development board was used for this study, which contains a Virtex-5 FX70T FPGA.

3.1 MicroBlaze

The following sections describe the hardware and software setup used to evaluate the performance of the uB.

3.1.1 Hardware

All hardware configurations for the uB were built using the Xilinx Platform Studio (XPS) Version 10.1.03. The hardware setup was performed using the Base System Builder (BSB). The common configuration is represented by the block diagram shown in Figure 3.

Figure 3. MicroBlaze block diagram.

As shown in Figure 3, the hardware configuration for the uB was similar for each test performed. Blocks that are not highlighted in red were always enabled during testing. To test the effect on benchmark performance, the uB was configured with either an IEEE-754 single-precision FPU, 64 KB of cache, or both the cache and FPU. Note that the uB has a configurable cache size for both the instruction cache and the data cache. For this test the cache was configured as 32 KB for the data cache and 32 KB for the instruction cache.

The memory configuration for testing was also configured differently for each test. The on-chip memory using the local memory bus (LMB) is always available and was configured as 64 KB block random-access memory (BRAM) and is non-cacheable. In addition to using the LMB for on-chip testing, a 64-KB BRAM connected to the processor local bus (PLB) was also used. The BRAM is non-cacheable. For testing external memory, the DDR2 multi-port memory controller (MPMC) or the static random-access memory (SRAM) multi-channel external memory controller (MCH_EMC) were used, each of which are cacheable.

All hardware configurations that were enabled or disabled during testing are highlighted in red within Figure 3.

The frequency for each test was also adjusted to evaluate the performance of the processor. Testing was conducted using frequencies of 25, 50, 75, 100, 125, 133, and 150 MHz.

The uB frequencies higher than 125 MHz were considered non-optimal as they did not meet timing for the system. The design was still built and tested at these higher frequencies. This was necessary to capture more data points since the external memory could not be tested at frequencies lower than 50 MHz for SRAM and 75 MHz for synchronous dynamic random-access memory (SDRAM).

3.1.2 Software

To test the performance of the soft-core processor, two benchmark applications were used. The Dhrystone v2.1 application was used to test fixed-point performance while the Whetstone v1.2 application was used to test floating-point performance.

When building the executables for these applications, all code data was either placed in:

- On-chip memory using the LMB,
- On-chip memory using the PLB,
- External memory using SRAM with cache enabled or disabled, or
- External memory using SDRAM with cache enabled or disabled.

When compiling the code, compiler optimizations were used. The compiler optimizations used were the following:

- Level 1: First-level optimization. Performs jump and pop optimizations.
- Level 2: Second-level optimization. Activates all optimizations that do not involve a speed-space trade-off. This is the standard optimization level used for program deployment.
- Level 3: Highest optimization level. Adds more expensive options, which increases code size and can produce less efficient code.

3.2 Leon3

The following sections describe the hardware and software test setup used for both the open-source Leon3 and the Leon3FT.

3.2.1 Open-Source Core

3.2.1.1 Hardware

All hardware configurations for the Leon3 were built using the GRLIB implementation tool Version 1.0.2.1 provided by Gaisler. In order to synthesize the design, Synplify Pro Version 9.6.1 was used. For place and route, Xilinx ISE Version 10.1.03 was used. The common configuration is represented by the block diagram shown in Figure 4.

Figure 4. Leon3 block diagram.

As shown in Figure 4, the hardware configuration for the Leon3 was similar for each test performed. The Leon3 utilizes an AMBA high-speed bus (AHB) for IP access and an AMBA peripheral bus (APB) for register access to certain IP blocks as shown in Figure 4.

The blocks shown in Figure 4 that are not highlighted in red were always enabled during testing. To test the effect on benchmark performance, the Leon3 was configured with either an IEEE-754 high-performance FPU, 64 KB of cache, or both the cache and FPU. Note that the Leon3 has a configurable cache size for both the instruction cache and the data cache. For this test the cache

was configured as a 4×8 KB instruction cache and data cache with 32 bytes/line. The replacement algorithm used was the least recently used (LRU) replacement algorithm.

The AHBRAM was used for on-chip memory and is cacheable. The AHBRAM was configured as 64 KB. For testing external memory the 8-bit external memory controller was used for SRAM, which is cacheable. The DDR2SPA external memory controller was used for the DDR2 memory and is also cacheable.

All hardware configurations that were enabled or disabled during testing are highlighted in red within Figure 4.

The frequency for each test was also adjusted to evaluate the performance of the processor. Testing was conducted using frequencies of 50, 80, and 100 MHz.

These frequencies were chosen to closely match that of the uB frequencies. Since the Leon design is implemented on the ML507 board, the core is clocked from the 100 MHz on-board clock. This clock connects to a digital clock manager (DCM) in the design, and the DCM output clocks the majority of the Leon logic. We can reduce the clock rate of the DCM, but cannot go below 32 MHz due to the constraints of the Xilinx DCM on an FX70T part. The DCM on the FX70T part is also capable of a maximum speed of 140 MHz, but the Leon design would not operate above 100 MHz. Modifications could have been made to utilize faster speeds, but the intent of this testing was to utilize the processor as is without making modifications to the design code itself in order to get a baseline case.

3.2.1.2 Software

To test the performance of the soft-core processor, two benchmark applications were used. The Dhrystone v2.1 application was used to test fixed-point performance while the Whetstone v1.2 application was used to test floating-point performance.

When building the executables for these applications, all code data was either placed in:

- On-chip memory using the AHBRAM with cache enabled or disabled,
- 8-bit external memory using SRAM with cache enabled or disabled, or
- External memory using SDRAM with cache enabled or disabled.

When compiling the code, compiler optimizations were used. The compiler optimizations used were the following:

- Level 1: First-level optimization. Performs jump and pop optimizations.
- Level 2: Second-level optimization. Activates all optimizations that do not involve a speed-space trade-off. This is the standard optimization level used for program deployment.
- Level 3: Highest optimization level. Adds more expensive options, which increases code size and can produce less efficient code.

All code compiled used either the v8 compiler switch, which allows the use of multiply/divide instructions, or the soft-float compiler switch to allow the support of floating point operations. The soft-float compiler switch is used to disable generation of floating point instructions. When the FPU is enabled, the soft-float compiler switch was not used.

3.2.2 Leon3FT Core

All hardware configurations for the Leon3FT were built using the GRLIB implementation tool Version 1.0.2.1 provided by Gaisler. In order to synthesize the design, Synplify Pro Version 9.6.1 was used. For place and route, Xilinx ISE Version 10.1.03 was used. The common configuration is represented by the block diagram shown in Figure 5.

Figure 5. Leon3FT block diagram.

algorithm. The provided Leon3FT netlist configures the processor with a memory management unit (MMU) and the IU was protected using the 8-bit parity option.

The memory configuration for testing was configured differently for each test. The FT AHBram was used for on-chip memory and was configured as 64 KB. Note that this memory is cacheable.

For testing external memory the 8-bit FT external memory controller was used for SRAM and is cacheable. EDAC with pipelining was enabled on the FT memory controller. Though enabled in the hardware configuration, EDAC could not be enabled through software due to the size of the SRAM on the ML507 board. The size of the SRAM on the ML507 board is only 36 bits, 32 data bits and 4 check bits. EDACing on the FT memory controller requires 7 check bits.

The DDR2SPA external memory controller was used for the DDR2 memory and is also cacheable.

All hardware configurations that were enabled or disabled during testing are highlighted in red within Figure 4.

The frequency for each test was also adjusted to evaluate the performance of the processor. Testing was conducted using frequencies of 50, 80, and 100 MHz.

These frequencies were chosen to closely match that of the uB frequencies. Since the Leon3FT design is implemented on the ML507 board, the core is clocked from the 100 MHz on-board clock. This clock connects to a DCM in the design, and the DCM output clocks the majority of the Leon3FT logic. We can reduce the clock rate of the DCM, but cannot go below 32 MHz due to the constraints of the Xilinx DCM on an FX70T part. The DCM on the FX70T part is also capable of a maximum speed of 140 MHz, but the Leon design would not operate above 100 MHz. Modifications could have been made to utilize faster speeds, but the intent of this testing was to utilize the processor as is without making modifications to the design code itself in order to get a baseline case.

3.2.2.2 Software

To test the performance of the soft-core processor, two benchmark applications were used. The Dhrystone v2.1 application was used to test fixed-point performance while the Whetstone v1.2 application was used to test floating-point performance.

When building the executables for these applications, all code data was either placed in:

- On-chip memory using the FT AHBram with FT cache enabled or disabled,
- 8-bit FT external memory using SRAM with FT cache enabled or disabled, or
- External memory using DDR2-SDRAM with FT cache enabled or disabled.

When compiling the code, compiler optimizations were used. The compiler optimizations used were the following:

- Level 1: First-level optimization. Performs jump and pop optimizations.
- Level 2: Second-level optimization. Activates all optimizations that do not involve a speed-space trade-off. This is the standard optimization level used for program deployment.
- Level 3: Highest optimization level. Adds more expensive options, which increases code size and can produce less efficient code.

All code compiled used either the v8 compiler switch, which allows the use of multiply/divide instructions, or the soft-float compiler switch to allow the support of floating point operations. The soft-float compiler switch is used to disable generation of floating point instructions. When the FPU is enabled, the soft-float compiler switch was not used.

The Leon3 utilizes the sparc-elf-gcc compiler.

4. RESULTS

As discussed in the previous sections, compiler optimization levels 1 through 3 were used to test performance of each processor. Figure 6 shows the speed-up seen when utilizing a certain compiler optimization.

Figure 6. Processor compiler optimizations.

Figure 6 only shows the processors running code out of SRAM with both the cache and FPU enabled. The intent of the figure is to demonstrate how much of a speed-up the compiler optimizations contributed to each processor. As seen in Figure 6, the uB sees a larger speed-up than the Leon3 when utilizing optimization level 3. The Leon, on the other hand, only sees a slight speed-up with each optimization level when compared to the slowest runtime of the uB. One of the reasons that there is a difference between the processors in terms of compiler optimizations is because of the compiler used for each processor.

Since compiler optimization level 2 is the standard optimization level used for program deployment, the following results shown are for this level only.

Though the FT memory controller used for SRAM on the Leon3FT was enabled in the hardware configuration, the EDACing on the memory controller could not be enabled in software due to the SRAM used on the ML507 board, as mentioned in Section 3.2.2.1.

4.1 Virtex-5 Dhrystone

The performance of each processor running the Dhrystone application is shown in Figure 7. The Dhrystone application measures fixed-point operations and is mainly used to test the performance of either using cache or no cache. The chart represents the performance for each processor at 100 MHz. The chart is normalized to the slowest performance measurement, which was the DDR2 memory on the uB. Part of the reason that the DDR2 may have performed slower

than the SRAM could be due to the type of memory controller used. The DDR2 memory controller used on the uB is different than the SRAM memory controller. The DDR2 memory had a performance measurement of 3.77 DMIPS and was normalized to one. Each memory unit tested was then compared against that value to show the speed-up that was seen when utilizing a certain configuration.

Figure 7. Virtex-5 Dhrystone comparison.

Figure 7 shows that at 100 MHz each processor has a slight performance increase over the DDR2 memory on the uB. When comparing on-chip memory the uB performs much better than the Leon3 while utilizing the LMB, but once cache is enabled on the Leon3 the performance is about the same. As mentioned in Section 3.1.1, the uB on-chip memory is not cacheable.

With cache disabled, the Leon3's performance is slightly better when running out of off-chip memory compared to the uB. Once cache is enabled, both processors perform about the same with a speed-up factor of about 20.

4.2 Virtex-5 Whetstone

The performance of each processor running the Whetstone application is shown in Figure 8. The Whetstone application measures single-precision floating point operations on each processor. Both processors ran the Whetstone application with either the FPU disabled or enabled. As with the Dhrystone chart, the Whetstone chart is normalized to the slowest benchmark, which was again the DDR2 memory on the uB. The DDR2 performance was at 0.057 WMIPS and was normalized to one. The results presented are also for a processor speed of 100 MHz.

Figure 8. Virtex-5 Whetstone comparison.

Figure 8 shows that with both cache and the FPU disabled, the performance for each processor was poor. However, using the on-chip memory does have a larger increase in performance over external memory, especially when using the LMB BRAM on the uB. Once the FPU is enabled, the Leon performance increases by a factor of about 400 whereas the uB only speeds up by a factor of about 40. The same can be seen with off-chip memory with the FPU enabled. The Leon3 performance increases by a factor of about 1000 whereas the uB still only speeds up by a factor of about 40. When running out of off-chip memory, the cache was also enabled when the FPU was enabled for both processors.

The FPU unit within the Leon3 performs much better than the FPU within the uB. This could be due to the larger amount of resources, namely lookup tables (LUTs), which the Leon3 FPU utilizes when compared to the uB. Also, the Leon3 FPU utilizes more instructions than the uB, which in turn would require fewer operations to perform a floating-point operation.

4.3 Virtex-5 Resources

Figure 9 shows the utilization for each soft-core processor with both the FPU enabled and cache enabled. Note that the utilization results are not in relation to a normalized reference point, but with respect to chip resources. Also note that the LMB BRAM and the PLB BRAM on the uB are non-cacheable.

Figure 9. Processor resource utilization.

Figure 9 represents a scenario in which each processor is configured with both the cache and an FPU to achieve the best performance possible. Under these circumstances, the Leon utilizes many more resources than the uB. Thus, there is a trade-off of performance versus resource utilization when using the Leon over the uB.

The following sections show a breakdown of resources utilized by each soft-core processor when certain configurations are used.

4.3.1 MicroBlaze

Figure 10 shows the resources consumed for each uB configuration, i.e., FPU enabled or disabled.

Figure 10. MicroBlaze resource utilization.

As shown in Figure 10, the uB consumes fairly low resources when utilizing various configurations. Overall, utilizing external memory with the external memory controllers increases resources, especially when utilizing the MPMC for the DDR memory. As expected, enabling cache increases BRAM utilization since cache is built with BRAMs. Enabling the FPU, on the other hand, had little effect on the increase in resource utilization.

Overall, the majority of the uB resource consumption comes from enabling the cache and using the DDR MPMC.

4.3.2 Open-Source Leon3

Figure 11 shows the resources consumed by the open-source Leon3 during testing.

Figure 11. Leon3 resource utilization.

As seen in Figure 11, the large increase in resource utilization is due to enabling the FPU. Looking at the number of LUTs used in Figure 11, with cache enabled, the resources slightly increase, but as soon as the FPU is enabled the resources increase by almost a factor of three. The large amounts of LUTs used within the V5 with the FPU enabled could be part of the reason for the large speed-up seen on the Leon3 over the uB on floating-point operations. Without the FPU enabled the Leon is almost comparable to the uB in terms of resources utilized. Also, since the high-performance FPU was utilized, many more resources were utilized. Gaisler provides a smaller FPU that uses fewer resources than the high-performance FPU.

4.3.3 Leon3FT

The resources utilized when enabling the FT components on the Leon3 are shown in Figure 12. The resources utilized are per the netlist provided during this testing. As mentioned previously, the Leon3FT is not as configurable as the open-source Leon3. For example, disabling the FPU results in enabling the FPU-Lite, cache size was fixed, the MMU was enabled, and FT protection for the IU was set at 8-bit parity protection.

Figure 12. Leon3FT resource utilization.

As seen in Figure 12, resource utilization increased slightly when FT components were enabled. Regardless of which FT component is enabled, the resource utilization for LUTs increased by an additional 10% of available chip resources whereas registers increased by an additional 5% of available chip resources. Also, as seen in Figure 12, the BRAM usage decreases by 10% of available chip resources once the FT portion is enabled when utilizing the on-chip memory or the SRAM memory. It appears that enabling the FT portions of the Leon3 increases LUT utilization by about 10% of available chip resources and decreases BRAM utilization by about 10% of available chip resources.

Overall the Leon3FT does not show a large resource increase when enabling the FPU, which is partly due to the fact that the FPU-lite is already enabled. Therefore, there is only a slight increase in resources going from the FPU-lite to the high-performance FPU.

5. CONCLUSION

The purpose of this report was to show the performance for each processor while running both a Dhrystone application, which benchmarks the fixed-point operations of the processor, and the Whetstone application, which benchmarks single-precision floating point operations. Both processors demonstrated an increased performance when enabling the FPU and cache, which is expected. With cache enabled, the processors both achieved about a 20x speed-up. On the other hand, with the FPU enabled, the Leon3 performed much better than the uB, achieving a speed-up of about 400x to 1000x. This is much faster than the uB, which only saw about a 40x speed-up.

Resource utilization for each processor was also included within this report. By not utilizing the FPU and cache, the Leon3 and uB processors were very comparable on the amount of resources used. The uB saw larger resource utilization when using the DDR2 MPMC. When enabling the cache, the uB saw an increase in BRAM usage, which was expected. The Leon3, on the other hand, saw a large increase in resource utilization once the FPU was enabled. This could partly account for the large performance increase when running the Whetstone application.

Performance is usually required when running a processor, which means utilizing the FPU and cache. With both these enabled, the Leon3 performed much better than the uB, but utilized many more resources than that processor.

Overall, both processors have their own trade-offs when used. The Leon3 appears to perform much faster than the uB when utilizing the FPU, but uses more resources than the uB. Also, the Leon3 FPU provides the ability to use both single-precision and double-precision floating-point operations whereas the uB FPU can only perform single-precision. Both processors, though, performed similarly when using cache and both were comparable in resources with cache enabled.

APPENDIX A. PPC440 PROCESSOR

A.1 PPC440 Results

Benchmarking of the hard-core PPC440 processor was also conducted to show a comparison of performance and resource utilization to the soft-core processors.

A.1.1 Benchmarking Results

The benchmarking results shown in Figures A-1 and A-2 are for a processor speed of 100 MHz. Note that this significantly constraints the performance of the processor, which is more efficient at higher speeds, but due to the limitations of the soft-core processors, this speed is chosen for comparison.

Figure A-1. PPC440 Dhrystone comparison.

As seen in Figure A-1, the PPC440 had a speed-up of about 35x with cache enabled, giving a better performance than the uB and Leon3 in which performance sped up by a factor of 20. Note that the DDR2 memory was not utilized with the PPC440 since at 100 MHz it could not be used with the memory controller.

When utilizing the floating-point unit (FPU) within the PPC440, Figure A-2 shows that the PPC440 performs much better than the uB, but still performs slower than the Leon3 with the FPU enabled.

Figure A-2. PPC440 Whetstone comparison.

As seen in Figure A-2, the PPC440 achieves a speed-up of about 250x to 350x whereas the uB only achieved a speed up of about 40x. The Leon3, on the other hand, still outperformed both the uB and PPC440. Once again, keep in mind that the FPU used with the PPC440 is constrained at 100 MHz for this test.

A.1.2 Utilization Results

The utilization results for the PPC440 are compared against the uB and Leon3 and are shown in A-3. Note that these results are for both the cache and FPU enabled. Also note that the DDR2 is shown within the utilization results.

Figure A-3. PPC440 resource utilization comparison.

Figure A-3 shows that the PPC440 utilizes fewer resources than the Leon3 and uB, which is expected since extra resources are not required to generate the hard-core PPC440, unlike the soft-core processors that require extra resources. The PPC440 design using static random-access memory (SRAM) with cache does not use block random-access memory (BRAM) to instantiate the cache. The design may be utilizing LUT-RAM to build the cache. The DDR2, on the other hand, does appear to use BRAM for the cache.

DISTRIBUTION

1	MS0503	James W. Daniels	5337
1	MS0503	Dominic A. Perea	5337
1	MS0503	Mythi M. To	5337
1	MS0503	Christopher K. Wojahn	5337
1	MS0503	Donald E. Tolsch	5339
1	MS0513	Richard D. Hunt	5336
1	MS0521	Ed E. Boucheron	2617
1	MS0661	Daniel E. Gallegos	2623
1	MS0661	Mark W. Learn	2623
1	MS0661	Aaron D. Niese	2623
1	MS0672	Brian P. Van Leeuwen	5628
1	MS0966	Veronica Chavez-Soto	5733
1	MS0971	Ethan L. Blansett	5735
1	MS0980	Jay F. Jakubczak	5710
1	MS0980	Matt P. Napier	5571
1	MS0982	Jaime Gomez	5732
1	MS0982	Dan J. Kral	5732
1	MS0986	David M. Bullington	2664
1	MS0986	Jonathon W. Donaldson	2664
1	MS0986	David Heine	2664
1	MS0986	Jeffrey L. Kalb	2664
1	MS0986	David S. Lee	2664
1	MS0986	J. (Heidi) Ruffner	2664
1	MS1027	John M. Eldridge	5632
1	MS1243	Raymond H. Byrne	5535
1	MS0359	D. Chavez, LDRD Office	1911
1	MS0899	Technical Library	9536 (<i>electronic copy</i>)