Zaidi, S. I. H., Nabina, A., Canagarajah, C. N., & Nunez-Yanez, J. L. (2008). Evaluating dynamic partial reconfiguration in the integer pipeline of a FPGA-based opensource processor. In International Conference on Field Programmable Logic and Applications, 2008 (FPL 2008), Heidelberg. (pp. 547 - 550). Institute of Electrical and Electronics Engineers (IEEE). 10.1109/FPL.2008.4630005

EVALUATING DYNAMIC PARTIAL RECONFIGURATION IN THE INTEGER PIPELINE OF A FPGA-BASED OPENSOURCE PROCESSOR

Izhar Zaidi, Atukem Nabina, CN Canagarajah, Jose Nunez-Yanez

Department of Electrical and Electronic Engineering
University of Bristol, UK
Email: izhar.zaidi@bris.ac.uk, Atukem.Nabina@bristol.ac.uk, Nishan.Canagarajah@bristol.ac.uk, j.l.nunez-yanez@bristol.ac.uk

ABSTRACT

This work explores the potential of sharing different arithmetic hardware operators tightly coupled to the integer pipeline of the open-source LEON3 processor. The idea is to map these modules to the same silicon area saving power consumption and area utilisation. The same strategy can be used to extend the architecture of processors optimized for applications with specific energy constraints. The proposed platform serves as a guideline to illustrate gains obtained through partial reconfiguration that need to adapt to changing standards and protocols with a limited number of resources.

1. INTRODUCTION

Partial Reconfiguration has sparked significant interest and has prompted investigations on how it can be used to improve the performance of a system in different domains. [1] proposes an automotive control unit application in which less critical units can be configured at run-time saving the battery life of the vehicle. The author argues the feasibility of having a reconfigurable architecture instead of an ASIC due to decrease in the electronic product life cycle. [2] also proposes a reconfigurable instruction set processor to reduce the power consumption. Their processor is tightly coupled with four reconfigurable slices that can be configured at run-time to provide high throughput for multimedia applications with a slight increase in power consumption. [3] provides some results & graphs related to both initial configuration of the system and Partial Reconfiguration using simulation and power analysis tools, however little information is provided regarding the size of the information being loaded and the runtime system measurements due to limitation of the tool. [4] proposes a Network on Reconfigurable Chip (NoRC) that runs multiple tasks and applications simultaneously. This system is composed of tiles that can be allocated specific functionalities at runtime to best compute applications and tasks based on constraints set by the system. Such constraints could be energy budgets, time constraints and area available for logic implementation. The incentive gained from reconfiguration has led to the development of a virtual internal configuration access port (JCAP) [10] in a low cost, low power Xilinx Spartan III platform. For the purpose of Reconfiguration, an internal processor [7], [10], or a dedicated controller [12], is used in order to reprogram part of the system without interrupting the current running processes. The traditional processors used with Xilinx reconfigurable FPGA's are the IBM PowerPC hard IP core and the MicroBlaze soft core processor. In Microblaze the netlist given is in encrypted format and the source code can be obtained from Xilinx at a cost.

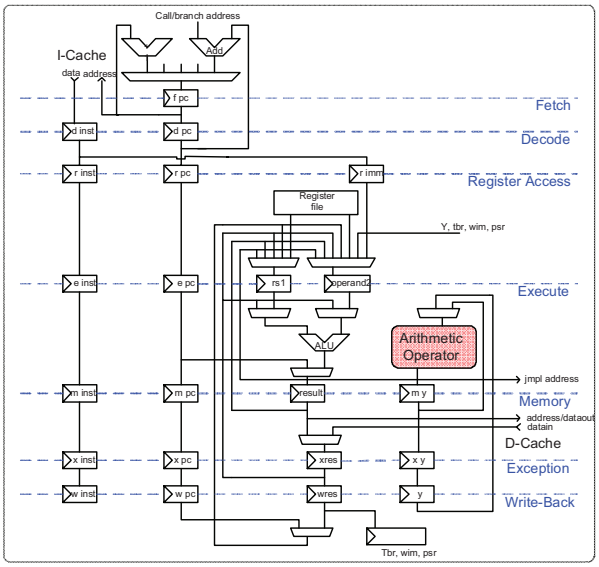
As power and energy consumption becomes more and more important factor in portable systems, it is important to be able to quantify the benefits and losses related to partial dynamic reconfiguration. This work is based on the scalable LEON3 open-source processor system from Gaisler Research. The integer pipeline was adapted and modified to allow the runtime reconfiguration of diffeent arithmetic operators including multiplier, divider Square Root, Cosine, Sine and Tangent with focus on both runtime and idle time savings on power consumption and area utilisation. The paper is organised as follows. Section 2 describes the overall system which includes the LEON3 processor and the modules used to enable Dynamic Partial Reconfiguration. Section 3 describes the procedure followed to implement this system and Section 4 describes the analysis of results. The last section gives information on the conclusion and future work.

2. SYSTEM DESCRIPTION AND TEST CONDITONS

The LEON3 micro architecture consists of a 7-stage pipeline with separate instruction and data caches. Its supports the full SPARC V8 instruction set [5]. The LEON3 system is provided with a number of generic modules which can be used to test and debug the entire system. These modules interconnect through the AMBA (Advanced Microprocessor Bus Architecture) bus system.

978-1-4244-1961-6/08/$25.00 ©2008 IEEE.

Fig. 1 Arithmetic module in integer unit pipeline



The multiplier divider (Mul/Div) unit within the integer pipeline of the processor was modified to provide the implementation of extra mathematical operations at runtime as required by a particular application. The Mul/Div unit must be enabled to conform to SPARC V8 specification. The different mathematical functionalities that were implemented are Multiplier, Divider, Sine/Cosine, Tangent and Square-Root. The multiplier module selected for this work implements 32x32 bit multiplication [6]. The selected divider module performs signed/unsigned 64-bit by 32-bit division taking 36 clock cycles and leaving no remainder. It implements the radix-2 non-restoring iterative division algorithm [6]. The Sine/Cosine, Tangent and Square-Root modules selected for this work were taken from Xilinx Core generator based CORDIC modules and an interface is designed to provide appropriate interfacing to the integer pipeline as shown in Fig.1.

ICAP (Internal Configuration Access Port) is the primitive used to provide access to the fabric of Xilinx FPGAs. It provides functionalities such as the ability to read and write to its internal registers and also to read and write to the configuration memory.

The ML402 Virtex-4 VSX 35 board was modified to allow the FPGA core voltage to be user controlled and to allow the measurement of voltage and current. This modification only allows the control of the internal core voltage and does not affect the I/O port voltage. This was achieved by disconnecting the voltage regulator providing power to the core of the FPGA and making the voltage terminals of the FPGA core accessible through unused pins on the GPIO port of the board.

Table 1. Resource Utilisation

	Slice utilisation	Size of the partial bitstream (KB)	% FPGA Utilisation
Leon3 Static	8819	-	57
Blank	-	43	0
Multiplier	315	51	2
Divider	431	54	2
Square Root	438	59	2
Sin/Cos	713	62	4
Arc Tan	760	62	4

The Keithley 2420 source meter was then used to provide voltage to the FPGA internal core and current and time information was measured to obtain the power used by the internal core of the device.

A program referred to as the "Keithley source meter control program" was written to allow control of the settings of the Keithley source meter as well as data extraction from the source meter to a host computer. The data that was extracted was voltage, current and time interval between measurements. This information was automatically stored in a spreadsheet. From these results power and energy information was computed.

There were 6 test cases that were used. Test case 1 was the complete LEON3 system with the ICAP hardware system and a partial reconfiguration region; however the reconfigurable region was left empty. Test case 2 was a replica of Test case 1 with the multiplier instantiated in the configurable region and similarly divider, square root, Sin/Cos, Arc Tan were configured in the reconfigurable region.

Enough time was allowed for the current to stabilise then 300 sequential measurements of voltage, current and time were taken for each case. The voltage and current information was averaged and used to calculate the static power consumption of the device. For these tests, no application program was run on the LEON3 processor and the operational frequency of the system for all test cases was 50 MHz. Another test was also carried out in order to measure power consumption of the system at run-time. This includes the power consumption by emulating the arithmetic operators using software or a dedicated hardware. Energy used during partial reconfiguration is also measured. This test was carried out by designing an application program to be run on the LEON3 processor using either algorithm or hardware. The bitstreams used for partial reconfiguration were appended.

Table 2. Power analysis

	Static Power (mW)	Average Runtime Power (mW)
Leon3 Static & Blank	604	749 (Div) 746 (Mul)
Leon3 Static & Multiplier	608	709
Leon3 Static & Divider	608	662
Leon3 Static & Square Root	609	757
Leon3 Static & Sin/Cos	613	720
Leon3 Static & Arc Tan	614	727

The application program starts with the initialisation process of ICAP and the timer module. Then, assuming that the reconfigurable module is initially blank, the application starts the timer and runs the partial reconfiguration function which transfers the required bitstream from the memory to the ICAP hardware system. The presence of the required module is then validated by running an appropriate instruction in assembly language and the number of clock cycles that was taken for the partial reconfiguration to be completed is noted. The power consumption of the LEON3 system during reconfiguration is taken by implementing a reconfiguration of different modules in the application program and by taking appropriate number of readings during this process and plotting it on the graph. This method of power measurement provides the accurate values for this particular implementation.

3. IMPLEMENTATION

To build an experimental setup for partial reconfiguration, the Mul/Div unit from the LEON3 core was moved to the top-level hierarchy of the system following the guidelines for partial reconfiguration [7] and a new top-level file was introduced which contains the static design, which was LEON3 itself, and a dynamic design containing the arithmetic operators

For self-reconfiguration, the ICAP hardware source code provided by Xilinx were modified and attached to LEON3 APB bus. The source code is written and supported for Xilinx PowerPC 405, MicroBlaze processor and OPB bus architecture and needs to be tailored accordingly for the AMBA bus architecture. The work started with introducing some address space for the ICAP hardware system as a slave on the APB bus, the wrapper was created to bridge

Table 3. Energy consumed during Reconfiguration

	Clock cycles to PDR	Average Power during PDR (mW)	PDR duration (mSec)	PDR Energy (mJ)
Multiplier	1510896	712	30.21	21.5
Divider	1602864	712	32.06	22.8
Square Root	1691568	712	33.83	24.1
Sin /Cos	1875552	712	37.51	26.7
Arc Tan	1820064	712	36.40	25.9
Blank	1266624	712	25.33	18.0

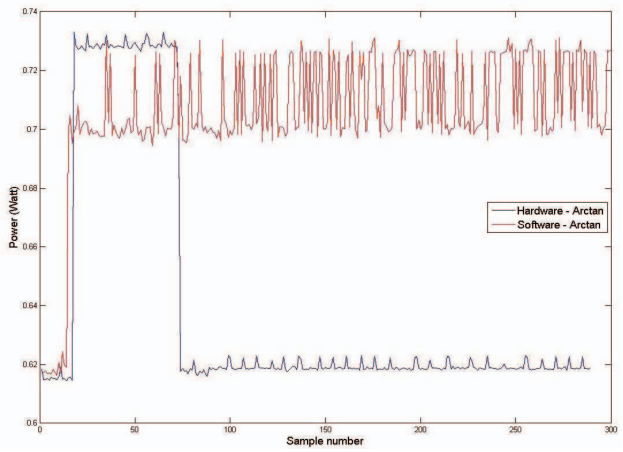
the signals of the AMBA bus to the OPB interface of the ICAP source code. The ICAP hardware system requires 4 clock cycles to complete a read/write cycle however a read/write transaction to the APB bus by default requires 2 clock cycles. Therefore, two extra cycle delays were introduced in APB to conform to ICAP hardware system signalling protocol. To allow communication between the processor and the partial module, bus macros which are predefined macros are used to constrain routing.

4. ANALYSIS AND RESULTS

Table 1, 2 and 3 represent the result from the experiments. Table 1 illustrates that instead of having the hardware resources residual in the system unutilised, area savings can be made by time multiplexing the modules. This constitutes a base for maximum utilisation of available resources in a portable system. In Table 2 the power analysis of our experiment are shown. The static design with no module in the reconfiguration region dissipates about 604mW without any program in operation. As compared to the one with any of the arithmetic module instantiated, it saves approximately 4-10mW. The run-time power readings for soft multiplication and division were also noted in table 2 and similar result is shown in Fig. 2. This depicts the comparison of power consumption between having software algorithms versus the hardware implementation. The over all gain with this system is having a complete area/power efficient system with full hardware acceleration advantages and capable of delivering the required results in a given time/energy budget.

The number of clock cycles required to reconfigure a specific arithmetic module is shown in Table 3. The reconfiguration of the Arc Tan module as an example consumes an average power of 712mW and takes approximately 36.40mSec to reprogram a 62kb of bitstream which comes out to be a total of 25.9mJ. This calculation gives the clear idea of energy consumption and time required to perform complete reconfiguration of the module.

Fig. 2. Hardware vs Software emulation of Arc Tan



5. CONCLUSION AND FUTURE WORK:

In this paper we have shown how Partial Reconfiguration can be used to obtain power and area saving in the inter pipeline of the LEON3 processor. We have also shown how power can be saved whilst using Partial Dynamic Reconfiguration by replacing a blank bitstream with an inactive module. The trade-off between the time and energy required for reconfiguring a particular module compared to the power utilised using software algorithm can be easily visualised with the help of these results.

The future work will include further investigation of power saving using dynamic voltage scaling [12] along with partial reconfiguration. These results can be carried forward as a model for a Dynamically Reconfigurable NoC-based SoC which has a requirement of running multiple new or existing applications simultaneously. The other area to explore is the possibility of having a dedicated reconfigurable controller so that the processor can offload this task to this IP.

References

[1] Michael Ullmann, Michael Hubner, Bjorn Grimm, Jurgen Becker. "An FPGA Run-Time System for Dynamical On-Demand Reconfiguration." 0-7695-2132-0/04 (IPDPS'04) IEEE.

[2] Francisco Barat, Murali Jayapala, , Tom Vander Aa, Rudy Lauwereins, Geert Deconinck, and Henk Corporaal "Low Power Coarse-Grained Reconfigurable Instruction Set Processor" FPL 2003, LNCS 2778, pp. 230–239, 2003.Springer-Verlag Berlin Heidelberg 2003

[3] Jurgen Becker, Michael Huebner, Michael Ullmann "Power Estimation and Power Measurement of Xilinx Virtex FPGAs: Trade-offs and Limitations." 07695-2009-/03 (SBCCI'03) IEEE

[4] Mohammad Hosseinabady and Jose Nunez-Yanez "Fault-Tolerant Dynamically Reconfigurable NoC-based SoC." (To be published ASAP'08)

[5] Gaisler Research LEON3 processor description.

[6] GRLIB IP Library User's Manual Version 1.0.14 Jiri Gaisler, Sandi Habnic, Edvin Catovic, Gaisler Research, 2006.

[7] PlanAhead as Platform for partial Reconfiguration http://www.xilinx.com/publications/xcellonline/xcell_55/xc_pdf/xc_prmethod55.pdf

[8] Xilinx "Virtex-4 Configuration Guide, ug071" January 12, 2007.

[9] Xiaofeng Wu and Tanya Vladimirova "A Self-reconfigurable System-on-Chip Architecture for Satellite On-Board Computer Maintenance" C. Jesshope and C. Egan (Eds.): ACSAC 2006, LNCS 4186, pp. 552 – 558, 2006. Springer-Verlag Berlin Heidelberg 2006.

[10] K.Paulsson, M.Hubner, G. Auer, M. Dreschmann, L. Chen, J. Becker "Implementation of a Virtual Internal Configuration Access Port (JCAP) for Enabling Partial Self-Reconfiguration on Xilinx Spartan III FPGA's" 10.1109/FPL.2007.4380671.

[11] Mateusz Majer "An FPGA based Dynamically Reconfigurable Platform: From Concept to Realization" 10.1109/FPL.2006.311364

[12] Jose Luis Nunez-Yanez, Vassilios Chouliaras. Jiri Gaisler "Dynamic Voltage Scaling in a FPGA-Based System-On-Chip" 10.1109/FPL.2007.4380689.