Hardware-Software Extensions To A Softcore Processor For FPGA-Based Adaptive PID Control

T. Gomes, P. Garcia, S. Pinto, F. Salgado, J. Cabral, J. Monteiro, A. Tavares
Centro Algoritmi-University of Minho
{tiago.a.gomes, paulo.garcia, sandro.pinto, filipe.salgado, jorge.cabral, joao.monteiro, adriano.tavares}@algoritmi.uminho.pt

Abstract- **Embedded Systems were traditionally implemented as a microprocessor surrounded by on-board peripherals, specifically assembled for a given application. Several Commercial Off-The Shelf solutions already provide a variety of on-chip custom modules, which allow a higher performance, smaller power consumption solution for a variety of applications. The advent of Field Programmable Gate Arrays (FPGA) allowed custom chips to be designed on a per-application basis, with fine-grain control over hardware/software partitioning.**

This paper presents a case study about the integration of an adaptive control system on a softcore processor. An MRAC-PID custom hardware module was developed and implemented on FPGA, taking advantage of the extensibility capabilities of the utilized softcore. Results demonstrate how software to hardware migration can accelerate system performance and maximize application parallelism.

I. Introduction and Related Work

Automatic control systems are widely used in a variety of industrial applications and several consumer appliances (e.g., satellite antennas) [1]. Historically, these control systems have undergone several technological evolutions, ranging from pneumatic implementations to analog electronics and, more recently, digital electronics, typically implemented in software [2]. Research in the automatic control field developed techniques far more powerful than the traditional Proportional Integrative Derivative (PID) controller, such as adaptive control algorithms [3]. These adaptive controllers, such as the MIT-based Model Reference Adaptive Control (MIT-based MRAC) [4], allow automatic control systems to auto-tune their operation in function of parameter variation in the controlled system, thus enabling far more precise control than traditional techniques, requiring less human intervention. The implementation of embedded controllers in large-scale systems is typically deployed on microcontrollers, on which software executes both the adaptive algorithm and the control law [5]. These microcontrollers are connected to peripherals which sense and actuate upon the target controlled system. In this kind of implementation, the maximum actuation frequency is limited by the time required for the processor to calculate the adaptive and control algorithms.

Software to hardware migration has been proven to greatly accelerate computations, thus providing higher performance than software-only implementations [6]. Although some years ago this approach was not feasible due to economies of scale, advances in hardware configurability have enabled custom hardware designs to be used in a wide range of applications in a cost-effective fashion [7]. FPGAs now find their way to a variety of industrial systems and consumer electronics [8]. FPGAs can be used to design application specific solutions, incorporating both custom hardware and software, running on softcore (or in the case of some FPGA platforms, hardcore) processors, where hardware and software can be co-designed to provide the best tradeoffs in terms of power consumption, performance, and other metrics [9]. FPGAs have already replaced Digital Signal Processors (DSPs) in several applications, as demonstrated by the work presented in [10]. Similarly, the work presented in [11] displays the use of a FPGA to accelerate DSP computations. FPGA implementations of control systems are also becoming ubiquitous, as demonstrated in [12], [13] and [14].

This paper presents the implementation of a MRAC-PID controller as a hardware module, integrated on a softcore processor. This approach, where the processor's Instruction Set is extended to allow the insertion of custom hardware modules in a tightly-coupled fashion, allows easy partitioning of hardware and software as well as efficient software access to application specific hardware for FPGA platforms. The remainder of this paper is organized as follows: Section II gives a brief overview of the softcore processor used for extension, describes the integrated adaptive control system, and how the two systems were interconnected in a co-designed fashion. Section III presents the experimentation to validate the system execution and displays and discusses the obtained results. Finally, Section IV concludes this paper.

II. Materials and Methods

A: M²μP: An Extensible Processor

M²μP is a 16-bits integer/32-bits floating point processor, developed for area-constrained low-power embedded applications [15]. It possesses a 4-stages pipeline, integer and (half-precision) floating point Arithmetic Logic Unit (ALU), separate L1 instruction and data caches, an extensible instruction set and up to 8 hardware-scheduled threads. The main design goal for M²μP was the scope and characteristics of its extension capabilities. Namely, its datapath can be extended with tightly-coupled Custom Computational Units (CCUs) and its instruction set can be extended with custom

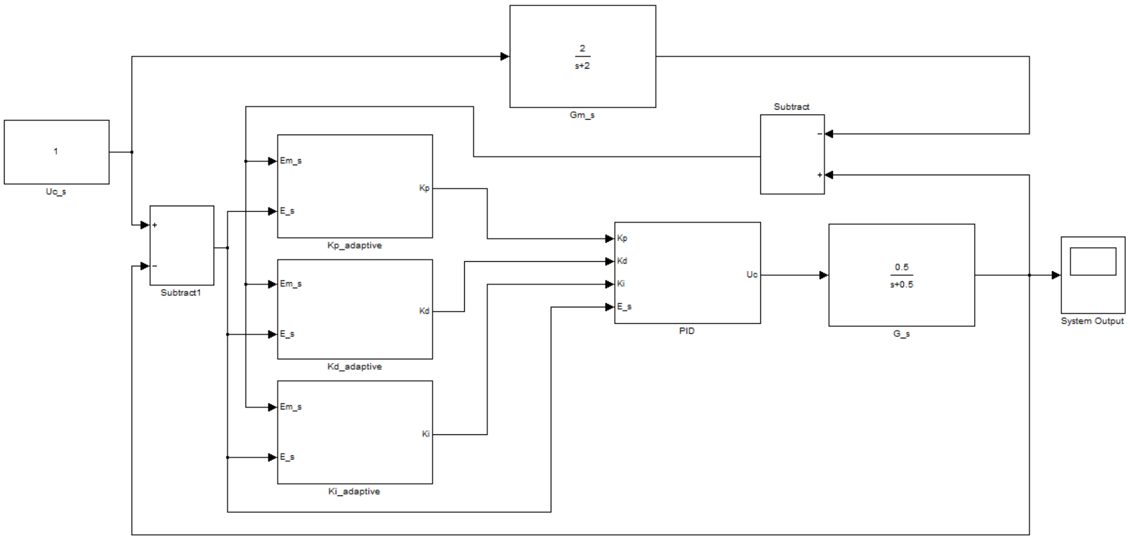

Fig. 1: MRAC-PID Controller Block Diagram

instructions to operate upon the integrated CCUs. The objectives of allowing such a tightly-coupled integration are: 1) to offer zero-latency access to CCUs to maximize performance and 2) to allow deterministic software access to CCU structures to allow their use in hard real-time applications, without the penalties imposed by the bus in co-processor implementations. Also, M²µP is a multi-threading microprocessor: the number of threads can be easily configured, (one to eight hardware threads can be operate concurrently) as well as the thread scheduling policy which can be set to Interleaved or Blocking Multi-Threading (IMT and BMT, respectively), in function of the application's requirements.

B: MRAC-PID Implementation

The MRAC-PID implementation was performed on two antagonistic approaches: in both, the PID controller was implemented as a hardware module integrated on M²µP. In the first approach, the MRAC adaptive control algorithm was implemented as software running on the softcore processor. In the second approach, the entire MRAC-PID algorithm was implemented in hardware. A block diagram of the implemented hardware system is depicted on Fig. 1.

C: System Integration and Test

The integration of the developed MRAC-PID module with the M²µP softcore processor was processed by taking advantage of M²µP's customization and extension capabilities. M²µP's Instruction Set was extended (several opcodes are reserved for extension) to incorporate special instructions to access the MRAC-PID module's internal control registers; only read/write operations for initial values are required, since the remaining functionality is completely implemented through the hardware. The module was then integrated on the processor's datapath, following a tightly-coupled approach; instead of hanging the module as a peripheral accessed through a system bus, which would cause non-determinism and latencies due to bus arbitration (i.e., loosely-coupled integration), the module was integrated directly as a datapath component; all accesses are processed internally, thus ensuring zero latency to read/write operations; writes are performed on the first pipeline stage, and reads follow the pipeline, also through the forwarding mechanisms. The M²µP is intended to be used in this fashion, thus resulting in highly area-effective application specific implementations.

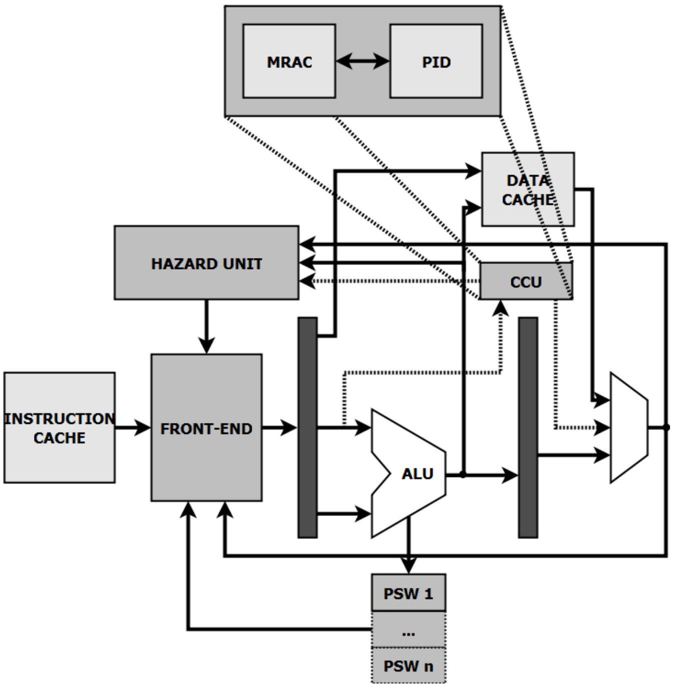

Fig. 2: M²µP and MRAC-PID Controller System Integration

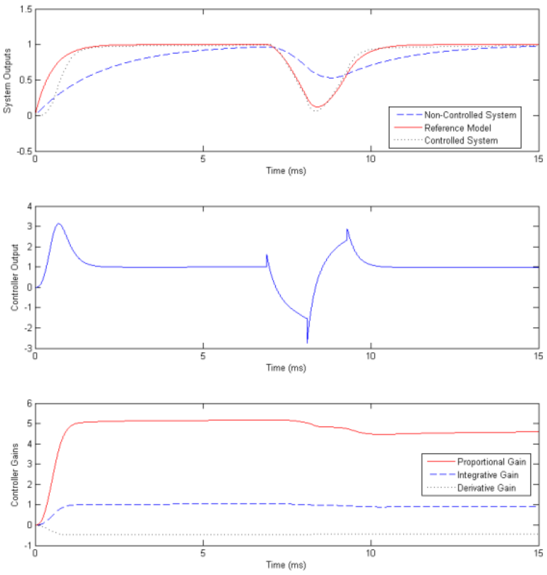

Fig. 3: Matlab Simulation Results

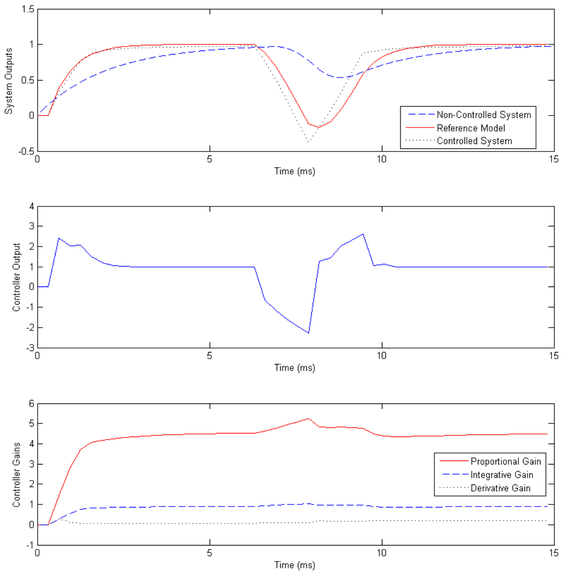

Fig. 4: ISIM Software-MRAC Simulation Results

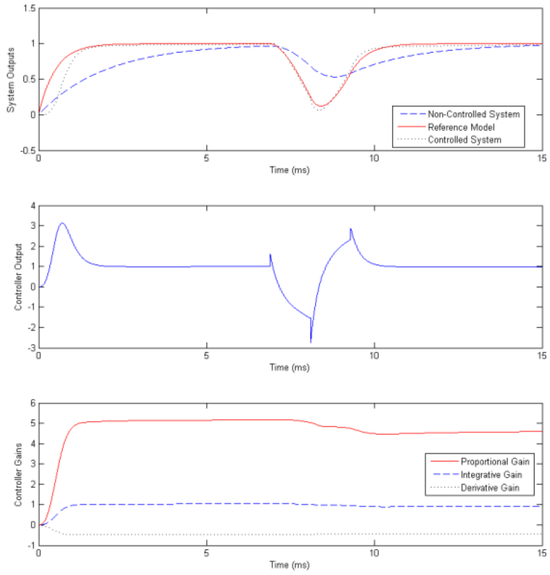

Fig. 5: ISIM Hardware-MRAC Simulation Results

The integration is depicted on Fig. 2 and was prototyped on a Xilinx Virtex 5 FPGA Development Board.

System validation was performed by implementing the MRAC-PID system: (1) in Matlab in the Laplace domain; (2) in Matlab, implementing equivalent discrete (time domain) system and validating the equivalent implementation; (3) performing simulation of the software implementation of the MRAC algorithm, with the PID controller implemented in hardware and; (4) performing simulation of the complete hardware implementation using Xilinx's ISE and comparing the outcomes.

III. Results and Discussion

Figs. 3, 4 and 5 depict the simulation results on: Matlab simulation (Fig. 3), Xilinx ISIM simulation using software implementation of the MRAC algorithm (Fig. 4) and Xilinx ISIM simulation using hardware implementation of the MRAC algorithm (Fig. 5).

The adaptation frequency for the Matlab model was chosen in function of the maximum possible frequency, obtained from the hardware implementation. As can be observed, the software implementation of the MRAC algorithm allows a smaller adaptation frequency, thus yielding poorer results than the hardware implementation. Additionally, in the hardware implementation, the softcore processor is completely free to compute any additional software; in the software implementation, if the processor should execute any additional functionality, the adaption frequency would be further decreased. Results clearly show how hardware implementations offer higher performance as well as increased flexibility by allowing explicit parallelism between functionalities, including between hardware and software. While with traditional technologies this approach was not feasible, the widespread use of FPGAs made approaches like this highly attractive. Although the integration of the MRAC hardware implementation increases the occupied area, this is not a design constraint (for the presented case study) since the softcore processor and the hardware PID controller occupy just a small portion of the available FPGA fabric. Fig. 6 displays the results obtained from simulation comparing the

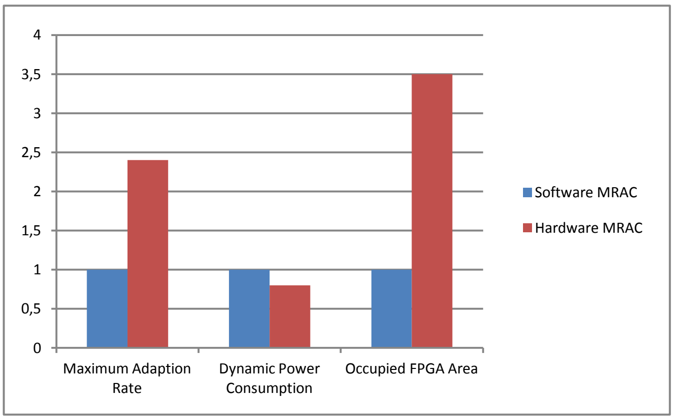

Fig. 6: Simulation Results for Software/Hardware MRAC implementations

software and hardware implementations of the MRAC controller.

IV. Conclusions

This paper presented the integration of a custom hardware module onto a softcore processor's datapath, performing an adaptive control algorithm. How hardware migration of software functionalities can greatly accelerate performance was demonstrated, as well as how parallelism can be more easily achieved through hardware/software co-design. The advantage of using extensible/customizable processors was demonstrated, especially in the case of applications which use FPGAs as solutions; in these cases, hybrid (hardware plus software) designs more often than not are capable of providing the optimum solution.

Future work will focus on expanding the extension/customization capabilities, especially regarding custom hardware integration. At this point, custom hardware requires addition of custom instructions, so software must be changed accordingly in order to utilize such modules. Research will focus on novel methods to seamlessly integrate hardware (migrated from software) without changes to other software, thus providing transparency to applications/system software to utilize functionalities: access will be provided by a set of Application Programming Interfaces which encapsulate the underlying implementation. The goal is to obtain a completely customizable system, where changes to one sub-system do not require changes to others, in order to realize application specific solutions with minimum effort, balancing the design choices to meet several constraints.

V. Acknowledgments

This work is supported by FEDER through COMPETE and national funds through FCT Foundation for Science and Technology in the framework of the project FCOMP-01-0124-FEDER-022674.

References

[1] Bennett, S.; , "Development of the PID controller," *Control Systems, IEEE* , vol.13, no.6, pp.58-62, 64-5, Dec 1993

[2] Meenakshi, M.; , "Microprocessor Based Digital PID Controller for Speed Control of D.C. Motor," *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on* , vol., no., pp.960-965, 16-18 July 2008

[3] Wang Wu; Bai Zheng-min; , "Performance Analysis of an Improved Single Neuron Adaptive PID Control," *Intelligent Information Technology and Security Informatics (IITSI), 2010 Third International Symposium on* , vol., no., pp.22-25, 2-4 April 2010

[4] Kungwalrut, P.; Thumma, M.; Tipsuwanporn, V.; Numsomran, A.; Boonsrimuang, P.; , "Design MRAC PID control for fan and plate process," SICE Annual Conference (SICE), 2011 Proceedings of , vol., no., pp.2944-2948, 13-18 Sept. 2011

[5] Ai Xiong; Yongkun Fan; , "Application of a PID Controller using MRAC Techniques for Control of the DC Electromotor Drive," Mechatronics and Automation, 2007. ICMA 2007. International Conference on , vol., no., pp.2616-2621, 5-8 Aug. 2007

[6] Chaves, R.; Kuzmanov, G.; Sousa, L.; Vassiliadis, S.; , "Cost-Efficient SHA Hardware Accelerators," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on , vol.16, no.8, pp.999-1008, Aug. 2008

[7] Sirowy, S.; Yonghui Wu; Lonardi, S.; Vahid, F.; , "Two-Level Microprocessor-Accelerator Partitioning," Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE '07 , vol., no., pp.1-6, 16-20 April 2007

[8] Saponara, S.; L'Insalata, N.; Tosti, F.; Fanucci, L.; Terreni, P.; , "Analysis of FPGA Solutions for Baseband Processing in Multi Carrier Communication Standards for Consumer Applications," Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on , vol., no., pp.1-2, 9-13 Jan. 2008

[9] Salgado, F.; Garcia, P.; Gomes, T.; Cabral, J.; Monteiro, J.; Tavares, A.; Ekpanyapong, M.; , "Exploring metrics tradeoffs in a multithreading extensible processor," Industrial Electronics (ISIE), 2012 IEEE International Symposium on , vol., no., pp.1375-1380, 28-31 May 2012

[10] Diggikar, A.B.; Ardhapurkar, S.S.; , "Design and implementation of adaptive filtering algorithm for Noise Cancellation in speech signal on FPGA," Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on , vol., no., pp.766-771, 21-22 March 2012

[11] Jingmeng Liu; Shangfeng Li; Xingming Wu; Baicheng Chen; , "Design of self-adaptive fuzzy-pid controller based on DSP and FPGA for rapid thermal processing," Industrial Electronics and Applications (ICIEA), 2011 6th IEEE Conference on , vol., no., pp.1649-1653, 21-23 June 2011

[12] Li Huirong; Zhang Xuefeng; Pang Lijuan; Tong Yanwei; , "Design and Implementation on the Temperature Control System of Roaster Based on FPGA and Fuzzy-PID Technology," Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on , vol.1, no., pp.396-399, 28-29 March 2011

[13] Ying-Shieh Kung; Chia-Sheng Chen; Kiing-Ing Wong; Ming-Hung Tsai; , "Development of a FPGA-based control IC for PMSM drive with adaptive fuzzy control," Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE , vol., no., pp. 6 pp., 6-10 Nov. 2005

[14] del Campo, I.; Martinez, M.V.; Echanobe, J.; Basterretxea, K.; Doctor, F.; , "A hardware/software embedded agent for real-time control of ambient-intelligence environments," Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on , vol., no., pp.1-8, 10-15 June 2012

[15] Salgado, F.; Garcia, P.; Gomes, T.; Cabral, J.; Mendes, J.; Ekpanyapong, M.; Tavares, A.; , "A customizable processor architecture for a design space exploration framework," Industrial Technology (ICIT), 2012 IEEE International Conference on , vol., no., pp.129-133, 19-21 March 2012