FPGA Based Emulation For Remote Wound Monitoring and Management

Vasileios Tsoutsouras, Sotirios Xydis, Dimitrios Soudris
Institute of Communication and Computer Systems, Greece
Contact info: {billtsou, sxydis, dsoudris}@microlab.ntua.gr

***Abstract*—Chronic wounds form a emerging hospitalization factor especially for elderly people. More than 10 million people in Europe suffer from chronic wounds, a number which is expected to grow due to the aging of the population. In order to address chronic wound management, SWAN-iCare project aims to develop a smart wearable and autonomous negative pressure device for wound monitoring and therapy. In this paper, we present a hardware-software framework for emulation, early functional prototyping and exploration of such wearable medical devices targeting to remote wound management. We analyze the requirements, the HW components and SW architecture for developing a realistic emulation platform for the specific application domain. We show that utilizing the proposed framework several architectural configurations can be explored in terms of performance and resource usage that can be further used as valuable feed-back during the design of the medical device.** [1]

***Index Terms*—wearable medical devices, HW/SW co-design, wound monitoring and management, emulation framework**

I. INTRODUCTION

Technology scaling and Improvement in electronic device manufacturing have enabled the increasing use of medical wearable devices. These devices are being (i) in close contact to the human body, (ii) autonomous while (iii) constantly monitoring various biological aspects and (iv) having the ability to react according to the state of the of their input. An emerging application domain of wearable devices is the management of patients with chronic, hard to heal wounds especially diabetic foot ulcers (DFU) and venus leg ulcer (VLU).

Thus, a wearable device constantly monitoring the status of the wound and providing information of the healing process and early identification of wound deterioration, can be proven extremely critical both for (i) improving the patients' quality of life, since patient's need for hospitalization is minimized with a reassurance that wound's condition is appropriately monitored, as well as (ii) minimizing healthcare costs, reduced hospitalization, without sacrificing the quality of treatment.

Swan-iCare [1], is an ambiguous project aiming at putting together all the necessary components to develop a system of efficient ecosystem for chronic wound management. Swan-iCare is based on the medical concept of Negative Pressure Wound Therapy (NPWT), in which negative pressure is applied on the wound to assist its healing process. In the core of Swan-iCare ecosystem, there is an embedded Smart Negative Pressure Wearable Device (SNPWD) to (i) monitor the biological parameters of the wound, (ii) combine them in order to assess the wound status status, (iii) enforce and control the negative pressure therapy and (iv) provide all these information to a Back-End clinical server for further analysis by the Healthcare Experts.

The deployment of such a multi-functional wearable devices is very complex procedure since it requires both hardware (HW) and software (SW) development, analysis and validation. Traditionally design approaches/methodologies serialize hardware and software development with the latter following the completion of the manufacturing of the first. However, such an approach is very time consuming, inducing also high recurring costs.

In this paper, we present a HW/SW framework that emulates remote wound monitoring and management wearable devices, for enabling early functional prototyping of the embedded application. The core of the HW framework is a Field Programmable Gate Array (FPGA) [2] device, extended with proper electronic equipment that implements communication, sensor data exchange, user I/F functions and regulation of (NPWT).

II. THE HW EMULATION PLATFORM

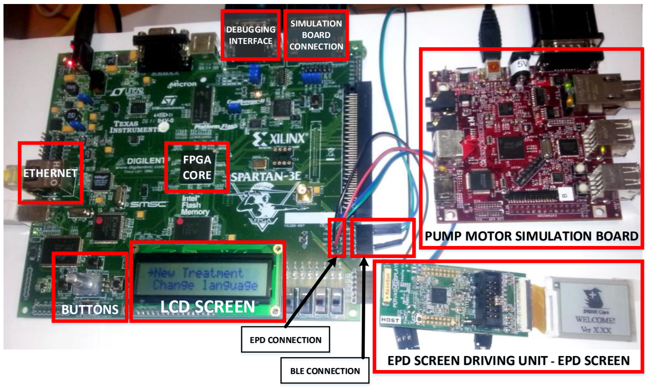

Fig. 1. The HW platform.

Figure 1 shows the HW platform assembled for emulating remote wound monitoring and management wearable devices. The basic element of the HW emulation platform is a Xilinx Spartan-III FPGA device [2]. The FPGA instantiates the control microprocessor and every interface of the peripheral devices is connected to it. We synthesize the MicroBlaze, a soft-core IP processor, provided by Xilinx. It is a RISC processor with 3-stage pipeline and clock frequency up to 50 MHz. Microblaze supports architectural parameters customization thus enabling exploration of differing design configurations to be performed, in order to tailor the design to the characteristics of the application's SW components. The on-board Ethernet module of Spartan-III has been used to implement the communication of the embedded device with clinical Back-end server. The LWIp tcp/ip protocol stack [6] was used and the processing of the packets was a task which was handled by the main processor. The server was implemented on desktop computer where the medical device uploads its information.

A Bluetooth Low Energy (BLE) module has been allocated for communicating wound sensor data with the main processor. Wireless interface has been selected instead of a wired one, since lack of wires enables the design of a more comfortable device.

Regarding to the user I/F, an ePaper Display (EPD) [4] was connected through a UART interface. The EPD shows information about the status of the wearable device and messages about the evolution of the treatment. We used the 1.44" EPD panel that satisfies the need for easy to read messages.

---

[1] This work has been partially supported by the EC funded project Swan-iCare (FP7-ICT, Project No. 317894)

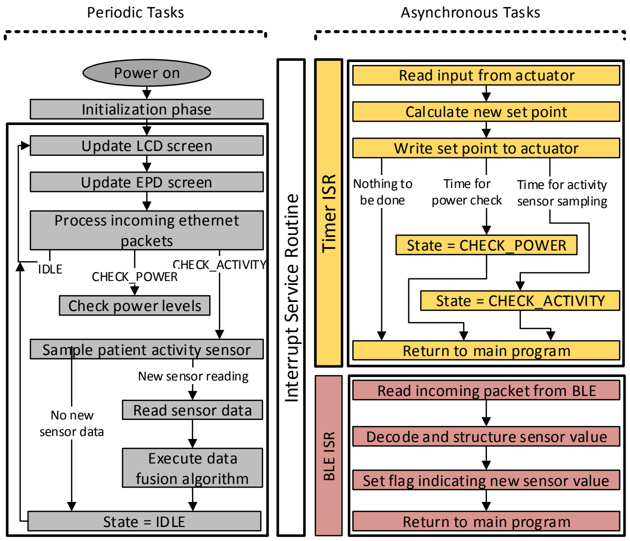

Fig. 2. Task interaction of the embedded SW application.

In the proposed HW platform, the real-time control of the pump is performed through the speed regulation of the motor driving the pump. A model of the pump in the time-domain was created in Matlab using nominal values from an actual DC motor and then a PI controller was configured to function as the controller of this motor. PI control was used in order to ensure the stability of the of the control by sacrificing settling time. This choice was made on the premise that it is important to ensure that the pump will not momentarily assert great pressure on the wound which could probably damage the tissue on and around the wound. To achieve real time simulation of both the motor behavior and the controller response, the model of the motor was implemented on a Beagle development board [5] external to the FPGA.

III. THE EMBEDDED SW APPLICATION

The embedded SW application of the SNPWD medical device will be responsible for the control of all the systems on the device. Fig. 2 depicts an abstract view of its internal tasks and their interaction. A subset of these tasks is periodic (left part of Figure 2) and need to be executed in predefined intervals. To achieve that, a timer is used, set to expire on the interval of the most time-critical of part of the code, in our case the pump control task. This ensures that pump control, the highest priority task of our design, will be executed on time no matter what the state of the main program is.

The same timer triggers other periodic tasks like power management or activity sensor sampling which should be executed in different interval compared to pump control. Other tasks, like input from BLE which is essentially input from in-wound sensors, are executed only when such data are present and thus falling into the category of event-driven tasks. In this case, the interrupting handling function sets appropriate flags to indicate presence of new data, which will be collected and analysed by the main part of the software only when other tasks of higher priority are complete.

A SW sensor data fusion engine is integrated with the SW application for evaluating and combining the data sampled by the various sensors. The fusion engine can generate either alarms, related to the detection of mechanical malfunctions, or warnings related to the detection of medical related critical situations. We devised a general fusion engine module that can be customized across differing treatment and therapy scenarios. For the medical decision making, classification algorithms used where Neural Networks (NN), Support Vector Machines (SVM) and Decision Trees (D. Trees) [3].

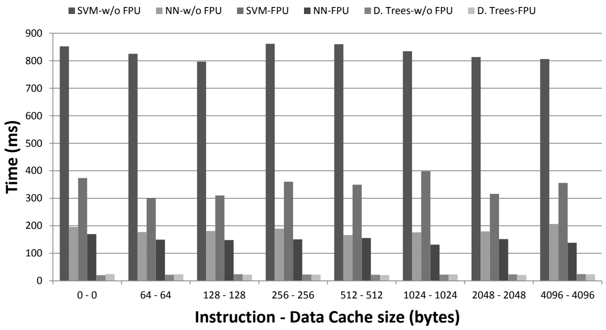

Fig. 3. Architectural configurations impact on performance.

IV. EXPERIMENTAL EVALUATION: EXPLORING THE IMPACT OF DESIGN ALTERNATIVES

In this section, we utilize the proposed HW/SW framework to analyse the impact of differing architectural decisions on the timing and resource usage. We focus our timing analysis on the data fusion engine that forms the heaviest computational component of the system. Specifically, we explore architectural decisions regarding to (i) the memory architecture, i.e. instruction and data cache system configuration, and (ii) the inclusion/exclusion of the a Floating Point Unit (FPU), across embedded application instances with differing machine learning algorithms for the data fusion engine, i.e. NN, SVM and D. Trees. Such analysis can be further used as feedback to the hardware design team, to customize the design of the medical device.

Figure 3 depicts the impact of cache size (instruction and data) and FPU allocation on the performance of the fusion engine. As shown the D. Trees forms the most efficient decision regarding to performance. The existence of an FPU in the microprocessor reduces the execution time of the algorithms operating on floating point data, like SVM and NN. Decision trees are not affected since their code structure is based on branch instructions, which are not requiring complex FP operations to benefit from the FPU. In contrast to the FPU, the cache memory size should be carefully chosen in order to speed-up the execution, since the data access patterns in memory can be such that the average execution is increased even compared to the design with no cache memory, e.g. 256 cache size configuration for the SVM w/o FPU.

V. CONCLUSION

In this paper we presented a framework for HW and SW emulation of wearable devices for remote wound monitoring and management. The framework utilizes the hardware design flexibility provided by FPGAs devices, which are further extended with a set of off-the-shelf hardware components to create a modular HW platform emulating state-of-art medical devices. An embedded SW application customized for wound monitoring and management has been developed and ported on the proposed HW platform, and an exploration campaign on the architectural design choices to tune design parameters suitable for the application specific HW and SW requirements.

REFERENCES

[1] [Online] SWAN-iCare, www.swan-icare.eu
[2] [Online] Xilinx Inc., http://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html
[3] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. 2012. Learning from Data. AMLBook.
[4] [Online] Pervasive Displays Inc., http://www.pervasivedisplays.com/kits/adapTag
[5] [Online] Beagle Board, http://beagleboard.org/beagleboard-xm
[6] [Online] LWIP, http://www.xilinx.com/ise/embedded/edk91i_docs/lwip_v2_00_a.pdf