A HW/SW Framework Emulating Wearable Devices 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). Chronic venous insufficiency and leg ulcers affect approximately 1-2 people per 1000 of the general population, with approximately 10-20 people per 1,000 ever affected [12]. Ulcer healing rates can be poor with up to 50% of venous ulcers open and unhealed for 9 months. Ulcer recurrence rates are worrying with up to one third of treated patients on their fourth or more episodes. In the UK leg ulcer treatment accounts for 1.3% of the total healthcare budget, while in US treatment costs approximates $3 billion per year [1]. In addition, approximately 1-4% of those with diabetes will develop a foot ulcer annually, and approximately 15% of those with diabetes will develop at least one foot ulcer during their lifetime [5]. The prevalence of diabetic foot ulcers has been estimated to be 3-8% in the diabetes population [6]. The annual incidence of foot ulcers in the US population has been estimated at 1.9% in type 1 and 2 diabetic patients. Various European studies suggest the incidence to be 2.1% and 3.6% [7], respectively.

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 [13], 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. On the other hand, pure software simulation suffers from long simulation times, while at the same time it requires severe deprecation of the original code, since many components. e.g. bluetooth devices, user IF etc, are not efficiently/realistically modelled.

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) [14] device, extended with proper electronic equipment that implements communication, sensor data exchange, user I/F functions and regulation of (NPWT). After all the functional software requirements are met, an exploration on the architectural parameters is performed to extract the most efficient configuration, which can be further provided as input to the hardware designers. The overall framework and its analysis is applied to the use case of remote wound monitoring and management, however it has been developed in a modular manner, that makes its retargeting to emulate other medical wearable devices straightforward. We first present the HW components of the proposed framework and then we discuss the SW features/functionality of the targeted wound management application. We then show through experimental evaluation, how the proposed framework can be used to explore the architectural configurations that lead to resource reduction without sacrificing performance efficiency.

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

MOBIHEALTH 2014, November 03-05, Athens, Greece

DOI 10.4108/icst.mobihealth.2014.257371

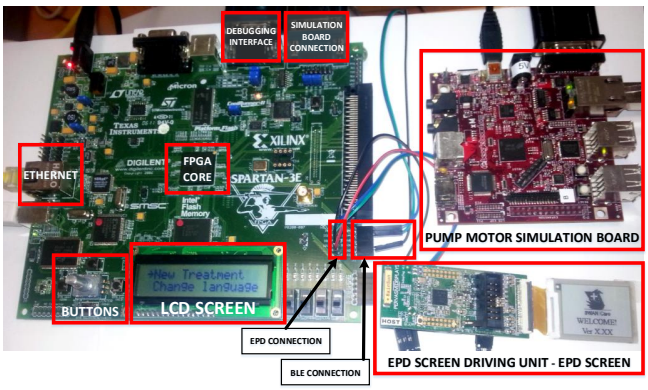

Fig. 1. The HW platform.

II. The HW emulation platform

The HW emulation platform should satisfy a number of requirements to achieve the desired functionality. We summarize the most important hardware requirements of the Swan-iCare wearable NPWT device and their intended use:

- The microprocessor should match the dynamics of the wound evolution and provide a good trade-off between processing power and power consumption.
- The user interface (I/F) should operate on low power mode, provide runtime information of the wound status and enable authorized users to customize the device according to a unique patient profile. The information presented may be critical for patient's health status. Since the device targets to elderly users, usually unfamiliar with electronic devices due to aging and/or mental disabilities, the I/F design should involve clear and easy to understand ways to convey information.
- An I/F for collecting sensor data should be allocated. The sensors may be a part of the hardware design or external devices communicating with the wearable one, through the designated interface.
- A communication interface for Wide Area Networks connectivity is necessary for medical data upload to a clinical back-end server that records the patient's biological parameters evolution.
- An actuator module implementing the therapy should be allocated. In the SNPWD example, the actuator under control is the pump enforcing the negative pressure therapy on the wound. It is a critical component of the device since it is directly linked to the evolution of the patients health.

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 [14]. 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. It forms a quite good match considering the requirements of a wearable medical device since it is more powerful than typical microcontrollers and has similar architectural features to ARM processors [16] that are frequently used in the design of portable devices. 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

TABLE I
IN-WOUND SWAN-ICARE SENSORS

MMPs	25 200 ug/mL	High level of MMPs is associated with poor healing [8].
Bacteria sensor	Y/N	If antibiotic resistant bacteria are detected in wound exudate,at the time an infection is diagnosed.
Wound pH	5.5 - 9	A pH $< 6,5$ or $> 7,3$ might be suggestive of infection if combined with other of infection. Colonization also be should considered above 7.3 [9].
Wound Temp.	20 - 35 ^oC	An increase in the wound temperature can be a sign of inflammation and infection [10].
TNFa	0 - 1500 pg/mL	TNFa is a pro inflammatory cytokine. Its increased level would reflect an inflammatory state of the wound, not compatible with proper healing [11].
CRP	Normal: 1-10 ug/mL Infection: 20 - 500 ug/mL	CRP is recognized as a general marker of infection

[19] 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. In Swan-iCare, the sensors are located on the wound of the patient, while the actual SNPWD device could be located far from the wound to better distribute its weight. Wired connection would not only be aesthetically displeasing but could also prove to be a point of failure if they were damaged. In addition, use of wireless communication technology provides the ability to create different configurations of used sensors according to the individual case of each user. Table I reports the in-wound integrated sensors that are developed within Swan-iCare to monitor the wound status.

Regarding to the user I/F, an ePaper Display (EPD) was connected through a UART interface. Specifically, we integrated the hardware and firmware components of Adaptag development kit [17] from Persavative Displays. 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.

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 [18] external to the FPGA. From a co-design point of view, this offers great flexibility since the external development board can be used to implement any model of any device under control with relative low effort and it is up to the software designer to satisfy the constraints of the control algorithm in the software managing the wearable medical device.

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. Similarly to HW platform, there are a number

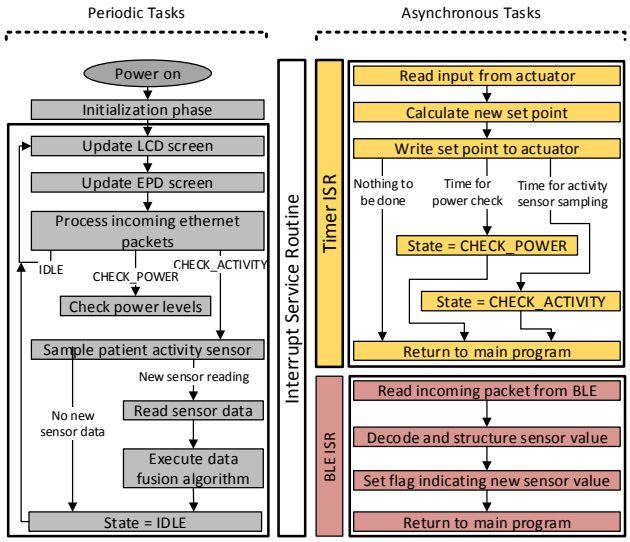

Fig. 2. Task interaction of the embedded SW application.

of requirements the embedded SW should meet for being safe and functional in the context of a medical device, i.e.:

- The SW design inherits all the non-functional requirements derived from the operation in an embedded environment, i.e. low processing power and memory availability, time constraints, reliability requirement to protect the patient under all circumstances, even in the event of a malfunction.
- To facilitate certification, the embedded SW is a bare-metal application, i.e. the SW is developed with no OS support to avoid the reliability issues regarding to resolution of complex interactions between the application and the management software. In addition, the embedded SW application is developed using only fixed sized variables.
- Dynamic memory allocation is avoided to eliminate the possibility that not enough memory is available when requested, due to careless memory management.

To further describe the SW application, the term task will be employed, to group a number of functions related to a specific sub-system of the device. For example, the pump control task encloses all the necessary actions to assess the current state of the negative pressure asserted on the wound and the calculation of its new operation point. Grouping certain functions into tasks enhances modularity of the software architecture. Modularity is essential to ensure that further development of the software, especially the incorporation of new tasks, will be achieved in a smooth way and will interfere with the other tasks in the least disturbing way. Additionally, task grouping enables the implementation of a finite state machine of the interactions and priorities of tasks. In general, the proposed methodology targets the early stages of the development cycle of both software and hardware and thus modularity both in hardware and software is the key to explore the design space and lay the foundation for the refinement of the initial co-design.The supported main tasks are:

- Start-up System Check & Calibration tasks
- User Interface tasks
- Communication tasks
- Reading Sensor Data tasks
- Sensor Data Fusion
- Pump Flow Control tasks

Figure 2 depicts an abstract view of the aforementioned tasks 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.

List of sensors	Frequency of measurement	Normal range or range of values	Send alarm when below or above these values		Alarm or warning to: Patient, Nurse, GP/specialist	DFU	VLU
			Min	Max			
Negative Pressure on wound							
Negative pressure device	Continuously	25 to 125 mmHg	25	125	Alarm to Patient + Nurse	x	x
Inflammation							
Wound temperature	Daily	20-25°C	<20	>25	Warning to Nurse and specialist	x	x
MMPs 9	Weekly	1-20 μg/mL	No alarm	No alarm	No alarm	x	x
TNFα	Weekly	0-1500 pg/mL	No alarm	No alarm	No alarm	x	x
CRP	Weekly	TBD	No alarm		No alarm	x	x
Infection							
pH	Daily	5.5 - 9	<6.5	>7.3?	Warning to Nurse and specialist	x	x

Fig. 3. Encoded medical conditions for alarm and/or warning generation at the SNPD data fusion engine.

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. The architecture of the data fusion engine consists of four levels:

- **Level 1:** This level is responsible for identifying hardware malfunction problems related to sensors. Due to the criticality of a medical system, this level produces alarms to protect the patient from the possible harm of a malfunctioning medical apparatus. An example of a malfunction would be a pH sensor to indicate value greater than 14.
- **Level 2:** In this level, a warning is produced whenever the input of the sensor is extreme according to the expected range indicated by the medical experts. In other words, a pH value of 13 is correct in terms of normal sensor operation but it is not likely to be acquired in a wound tissue.
- **Level 3:** In this case, measured values are within expected range but also within the range of what the medical experts believe to indicate a deteriorating course for the status of the wound.
- **Level 4:** This final level incorporates modern data management and classification algorithms which enable the system to infer the status of the wound using complex correlations of sensor data. These correlations are probably impossible to be discovered by mere inspection of data and as a consequence the use of these analytical tools is imperative.

Regarding to Level 4 of the data fusion engine, medical experts provided a characterization of the ranges of values read from the sensors and their relationship with wound status. Figure 3 depicts the encoded medical conditions that suggest an alarm or warning generation. These directives were used to create an artificial labelled data set which in turn was used to train machine learning algorithms and the subset of the

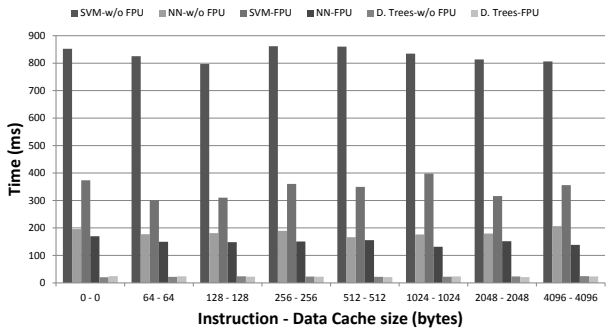

Fig. 4. Architectural configurations impact on performance.

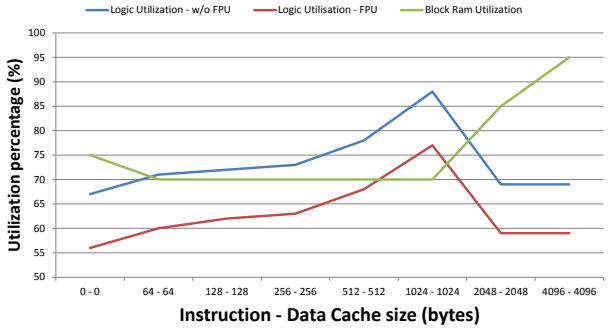

Fig. 5. Resource (logic and memory) utilization for differing architectural configurations.

algorithms that achieved an acceptable classification accuracy threshold, where tested to discover whether their computational needs are met by the hardware resources. Eventually, the classification algorithms used where Neural Networks (NN), Support Vector Machines (SVM) and Decision Trees (D. Trees) [15].

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 4 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.

Figure 5 shows an analysis on the FPGA's resource, i.e. logic and memory, utilization that each configuration exhibits. Resource utilization can be directly linked to the cost of implementation, since it is highly correlated to area complexity. As expected, up to 1024 cache size, the logic utilization is increased. For configurations with $> 1024B$ cache size, we observe a decreasing logic utilization with a emerging increase in memory utilization. The observed knee is due to FPGA device specific optimization, since for cache sizes smaller than $1024B$ the Xilinx synthesis tools maps the cache memory tag arrays to scratch registers, for greater than $1024B$ it maps the cache's tag memory arrays to on-chip RAMs (i.e. BRAMs).

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] Bergan JJ, Schmid-Schonbein GW, Smith PD, et al. Chronic venous disease. N Engl J Med. 2006;355(5):488-498.
[2] King H , Aubert RE, Herman WH. Global burden of diabetes, 1995-2025: prevalence, numerical estimates, and projections. Diabetes care. 1998; 21:1414-1431.
[3] Akthar S, Shaper N, Apelqvist J et al. A review of the Eurodiale studies: What lessons for diabetic foot care? Current Diabetes Reports. 2011; vol. 11. Nr. 4: 302-309.
[4] IDF diabetes atlas, http://www.idf.org/atlasmap/atlasmap
[5] Bartus CL, Margolis DJ. Reducing the incidence of foot ulceration and amputation in diabetes. Current Diabetes Reports. 2004; 4:413-418.
[6] Apelqvist J, Larsson J. What is the most effective way to reduce incidence of amputation in diabteic foot. Diabetes/metabolism research and reviews. 2000;16 (suppl. 1):75-83.
[7] Bartus CL, Margolis DJ. Reducing the incidence of foot ulceration and amputation in diabetes. Current Diabetes Reports. 2004; 4:413-418.
[8] Yu Liu, Danquing Min, Thyra Bolton, Vanessa Nube, Stephen M. Twigg, Dennis K. Yue, Susan V. McLennan, Increased Matrix Metalloproteinase-9 Predicts Poor Wound Healing in Diabetic Foot Ulcers, Diabetes Care 32:117119, 2009.
[9] Lars Alexander Schneider, Andreas Korber, Stephan Grabbe, Joachim Dissemond, Influence of pH on wound-healing: a new perspective for wound-therapy?. Arch Dermatol Res (2007) 298:413420.
[10] Manish Bharara, Jeffrey Schoess, Aksone Nouvong and David G. Armstrong, Wound Inflammatory Index: A Proof of Concept Study to Assess Wound Healing Trajectory., Ph.D. Journal of Diabetes Science and Technology Volume 4, Issue 4, July 2010.
[11] Manjit S. Gohel, MB, MRCS, Robin A. J. Windhaber, MSc, MRCS, John F. Tarlton, PhD, Mark R. Whyman, MS, FRCS, and Keith R. Poskitt, MD, FRCS., The relationship between cytokine concentrations and wound healing in chronic venous ulceration, J. Vasc Surg 2008; 48:1272-7
[12] John J. Bergan, M.D., Geert W. Schmid-Schnbein, Ph.D., Philip D. Coleridge Smith, D.M., Andrew N. Nicolaides, M.S., Michel R. Boisseau, M.D., and Bo Eklof, M.D., Ph.D., Chronic Venous Disease, N Engl J Med 2006; 355:488-498 August 3, 2006
[13] [Online] SWAN-iCare, www.swan-icare.eu
[14] [Online] Xilinx Inc., http://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html
[15] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. 2012. Learning from Data. AMLBook.
[16] [Online] ARM Inc, http://www.arm.com
[17] [Online] Pervasive Displays Inc., http://www.pervasivedisplays.com/kits/adapTag
[18] [Online] Beagle Board, http://beagleboard.org/beagleboard-xm
[19] [Online] LWIP, http://www.xilinx.com/ise/embedded/edk91i_docs/lwip_v2_00_a.pdf