Design and Implementation of a Fall Detection System on a Zynq board

Sahar Abdelhedi[1], Mouna Baklouti[2], Riad Bourguiba[1], Jaouhar Mouine[3].

[1] Université de Tunis El Manar, Ecole Nationale d'Ingénieurs de Tunis, LR11ES20 Laboratoire d'Analyse, Conception et Commande des Systèmes, 1002, Tunis, Tunisia.
[2] TELNET Innovation Labs, TELNET Group, Parc Technologique El Ghazala, Tunisia.
[3] Department of Electrical Engineering, Prince Sattam Bin Abdulaziz University Al-Kharj, Kingdom of Saudi Arabia.

Abstract— Population aging has become a worldwide problem. Fall accidents are considered as one of the major health risks, especially for elderly people. Fall detection devices are the key to distinguish a fall from daily activities, automatically alert when a fall occurred and significantly decrease the time of rescue when the monitored patient falls down. The proposed prototype is composed of a tri-axial accelerometer communicating to a Zybo board through Inter-Integrated Circuit (I^2C) interface. A threshold-based algorithm has been implemented based on peaks acceleration detection and inactivity posture recognition after falling and executed as a standalone application on a Zynq Z-7010 Field Programmable Gate Array (FPGA). This first implementation using Vivado and Xilinx SDK showed prominent results in terms of power consumption and time of execution.

Keywords—e-health; embedded systems; fall detection; FPGA; tri-axial accelerometer.

I. INTRODUCTION

Elderly people are the most growing segment of the population and this trend is increasing year after year. Fall accidents are one of the main public health problems among aging society and have a critical role in the deterioration of elderly health and the subsequent need of care. Falling causes several damages to health, sometimes even morbidity. With the increased number of the elderly suffering from fall events each year, it became urgent to find efficient ways to quickly detect falls and help the elderly population after falling.

Remote monitoring devices for chronic patients are one of the best answer to this problem and can ensure the health safety of the elderly. The development of MEMs sensors and wireless networks has made the long-tem monitoring system for elderly possible. In fact, integrating wearable sensors into a fall detection system is suitable for acquiring information about health and activity status of the patients wearing them. A new trend in fall detection is designing a small, smart and wearable device with adapted fall detection algorithms. Multiples technologies like micro-controllers, FPGAs and smart phones have been used to detect falls and have shown prominent results [1].

Main reasons for the development of fall detection devices is to allow non-sufficient people to live safely and independently in their familiar environment. It is supported by new technologies and approaches like video-based approach, environmental techniques using pressure sensors and wearable devices

Beside this, older people living by their own, need to be monitored during the whole day including normal activities like standing, sitting, walking or during accidents like falling. If there is no assistance directly after a fall, it can result in serious injuries. Real-time and accurate fall detection systems can improve quality of life for elderly population and supply help for timely assistance [2].

To accurately detect falls and send real-time alarm messages, it is important to develop an efficient fall detection system that automatically distinguish falling among daily activities and to alert relatives in time to avoid more important consequences after the impact of falling [3]. FPGAs can be considered as one of the best promising technology as it offers a flexible platform that enables the designer to achieve the best performance from hardware (HW) acceleration and can be promptly re-configured with an appropriate soft-core processor and a set of peripheral interfaces.

This work aims to present a fall detection prototype with only one tri-accelerometer that gives in the same time acceleration data of the 3 axes and angular motion information in the horizontal plane connected to a Zybo board (Zynq Z-7010 FPGA) through I^2C communication protocol and when a fall occurs, data are sent wirelessly to caregivers.

This paper is organized as follow. Section II gives an overview of the different works in literature dealing with fall detection algorithms. In section III, we will give an overview of the system architecture. Section IV introduces threshold fall detection methods and presents our fall detection algorithm. Section V shows our implementation results and the evaluation of our approach. Finally, discussion and conclusion are presented in the last section.

II. RELATED WORK

Despite extensive preventive efforts, falling remains a very important healthcare issue. Designing efficient fall detection systems becomes mandatory.

Different research projects have been undertaken to answer this need and to design low-power, small-size and accurate monitoring systems for elderly fall detection. One of the most promising technology to design HW architectures and work on new solutions to maximize efficiency and increase calculation capabilities is FPGAs [4]. In fact, FPGA offers the advantage

978-1-5090-4320-0/16/$31.00 ©2016 IEEE

to send the raw data acquired from the accelerometer to the transmission module quickly and efficiently.

Ahola et Al [5] developed a new sensor platform based on FPGA and targets various applications like fall detection. The main goal of the advanced sensor processor architecture, introduced in this paper, is to minimize power consumption by performing signal processing on a compact and energy processor unit. The hardware prototype was designed to be semi-modular and includes FPGA, battery power management and few sensors (3-axis accelerometer, gyroscope, magnetometer). As a result, like other sensor platforms, this new one is also power hungry with 50 mA of current consumption in sleep mode and more than 100 mA in active mode but it's still a very attractive platform by the flexibility offered by FPGA that can be adapted to a wide range of research areas.

Other searches have been done in this thematic like [6] where the FPGA runs in parallel two different algorithms to detect falls: the 1st one estimates the patient's orientation and the 2nd one estimates his acceleration. The main results of this work showed that compared to a dual-core processor, the FPGA implementation offers a great speedup over the software (SW) implementation with the advantage of reducing power consumption.

Grassi et Al [7] presented an integrated system with 3 devices: MEMS wearable wireless accelerometer and a 3D Time-of-Flight camera to detect falls in a home context. Fall detection algorithms are implemented on a low power FPGA, which runs in parallel six different routines: two approaches for each axis. The first one measures the stress in terms of fall energy, while the second checks the acceleration shape and the third one estimates the orientation of the person after a fall.

A bio-inspired stereo vision fall detection system is introduced in [8]. The system design includes two optical detector chips, an FPGA, a digital signal processor (DSP) and a wireless communication module. The optical chips capture video frames. The FPGA creates the input data for the DSP by calculating 3D representations of the environment. The DSP is loaded with a neural network that is used for classification purposes. Falls are divided into 4 phases: pre-fall, critical, post-fall, and recovery phase. The trial results are 90% of fall detection rate for all networks, and 97%–98% for the best network.

This work [9] aims to present a fall detection system based on the Shimmer platform and Zynq board. In this prototype, the accelerometer data are sent to the processing unit using Bluetooth wireless technology. The Discrete Wavelet Transform (DWT) is used for preprocessing the raw data collected from the Shimmer platform then dimensionality reduction is realized using Principal Component Analysis (PCA) and finally classification is carried out using a Decision Tree (DT) classifier. This system has been designed and tested using MATLAB for SW verification and then the HW implementation has been performed on Zynq FPGA (SoC). Using the Zynq platform permitted a high flexibility in designing the system and a better performance depending on how much the design is complex.

Bianchi et Al [10] introduced the integration of a wireless sensor platform to a home control and monitoring network to recognize abnormal gait and falls. Low power strategies based on a threshold fall detection algorithm are implemented at both software and hardware levels (both for microcontroller and FPGA). The sensor platform prototype includes a 3D accelerometer, a Zigbee transceiver and a PIC18LF4620 microcontroller. It has been implemented as a subsystem of an ASIC design and showed prominent results in terms lower power consumption, reliability and portability.

In this work compared to the existing ones, we will mainly focus on the implementation of a threshold-based fall detection algorithm on Zynq-7000 FPGA board. The data used for analysis are acquired in real-time from one tri-axial accelerometer ADXL345 through I^2C interface. We managed with this prototype to speed up the fall detection while conserving algorithm accuracy and minimize power consumption.

III. Fall Detection Prototype Architecture

A. System Overview

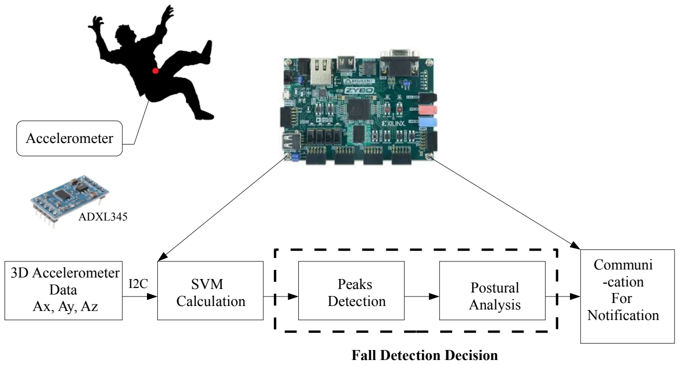


Fig. 1: HW Configuration of the fall detection system

The proposed fall detection prototype, illustrated in Fig.1 is composed of a 3-axis accelerometer ADXL345 connected to a Zynq Z-7010 FPGA [11] through I^2C communication protocol. We choose I^2C communication protocol as it can support multi-master system, communicates at 100 KHz and even if it is more complex than SPI, its SW implementation can be done easily.

The 3D-accelerometer data are acquired from the ADXL345 through I^2C communication protocol. I^2C signals are routed through the extended multiplexed I/O interface (EMIO), passed through to the Programming Logic (PL) of the Zybo board and then the acquired data are widely sent to the Processing System (PS) of the Zybo board where data are analyzed (SVM calculation, thresholds comparison...) according to the fall detection algorithm. In case of a fall, the board LEDs are blinking to alert that a fall occurred and a warning message is sent to the relatives. In resume, this system aims to sense the body posture change, analyze the 3D-accelerometer data and decides whether a patient is falling or not.

B. 3D-Accelerometer (ADXL345)

A single 3D accelerometer is used to extract different motion parameters, to avoid power dissipation on a wider set of MEMs devices [12]. It can be used for both acceleration and angle estimation, detect magnetic strength in 3 directions and provide angular motion information in the horizontal plane.

Only one accelerometer is needed for our prototype which is a strong guarantee for the size and energy consumption [12].

A digital MEMS accelerometer ADXL345, from Analog Devices, is used to implement the fall detection algorithm. It features a serial interface which is either SPI or I^2C. It has a high resolution (13 bit) measurement at up to +/- 16g of sensitivity and measures the static acceleration of gravity in tilt-sensing application, as well as dynamic acceleration resulting from motion to shock [13]. Its high resolution (3,9mg/LSB) enables also measurement of inclination change less than 1.0°. An integrated memory management system with a 32-level first in, first out (FIFO) buffer can also be used to store data to minimize host processor activity and lower overall system power consumption. The low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation which is a great advantage to detect falling of elderly people. [14]

C. *Zynq Z-7010 FPGA*

It is important to design not only a reliable fall detection system but also a system with a real-time response to help saving lives, money and time.

Combined with a processor, FPGA technology can offer a reconfigurable and flexible architecture in a single chip which bypasses the problem of power consumption [4], especially with the development of new low-power FPGA devices this last years like Xilinx 7 series FPGAs with power estimation and optimization features. FPGA offers also HW acceleration to achieve better performance in terms of both processing and response time. In fact, as it aims healthcare applications, the notion of time is extremely important, any improvement in terms of processing time can be vital.

The Zynq system on chip (SoC) integrates a dual-core ARM Cortex A9 based processing system (PS) and programmable logic (PL). The Zynq platform offers the flexibility and scalability of an FPGA as well as the performance and ease of use associated with ASIC which will make it suitable for a large number of applications [15]. The communication between the PS and PL is performed using the advanced extensible interface (AXI) protocol. The use of the Zynq platform enables a high flexibility in designing the system allowing the execution of some parts of the system on the PS in a SW manner leaving the rest for HW acceleration on the PL. This board is also affordable in terms of cost and offers great performance in terms of scalability, efficiency, execution time and ease of use.

Fig.2 shows the proposed system architecture for fall detection. The system is mainly composed of an embedded processor and hardware accelerators. The processor is used to manage and control the different operations and also execute SW tasks such as calculation operations. The HW cores are dedicated to fall detection alert.

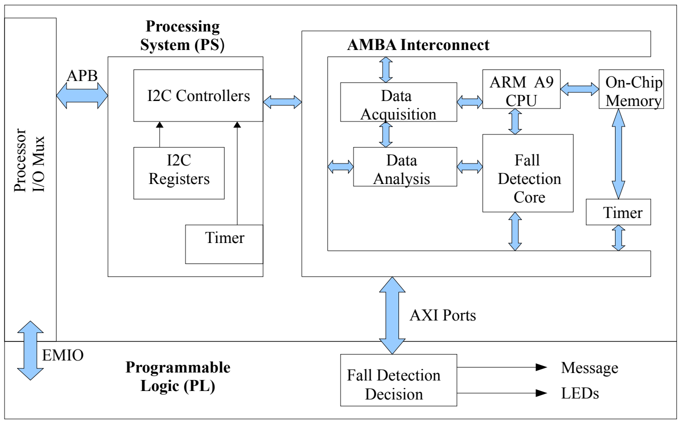


Fig. 2: Proposed Architecture on the Zynq board

In the proposed architecture, the communication between the ADXL345 and the Zybo board is via I^2C communication protocol. The I^2C module is a bus controller that can works either as a master or a slave in a multi-master design. It supports an extremely wide clock frequency range from DC up to 400 Kb/s [11]. In master mode, a transfer can only be initiated by the processor writing the slave address into the I^2C address register. The output acquired through the I^2C module are sent to the peripheral controllers connected to the processor as slaves via Advanced Microcontrollers Bus Architecture (AMBA) interconnect and contains readable and writable registers that are addressable in the processors memory space. As the fall detection algorithm is implemented as a standalone application on the ARM Cortex A9 CPU of the application processor unit (APU), the modules of data analysis, SVM calculation, thresholds comparison , postural analysis are integrated into the PS part of the Zybo board. The fall detection decision is sent as a data flow through AXI interface to the PL where LEDs blink and a message is sent to relatives in case of a fall.

IV. Fall Detection Algorithm

A. *Threshold-based fall detection algorithms:*

The human body fall is generally a spontaneous loss of balance and happens in a short period. During this time, the 3-vector of the human body gravity and the acceleration are changing greatly, the variation of body posture follows. [16].

Threshold-based methods are the most commonly used to detect elderly people falling in their home context [17]. In this technique, if the resultant acceleration is above fixed thresholds and the transition time between 2 daily postures is shorter than specified at the same time, a fall happens. In this kind of algorithms, it is necessary to start by defining several parameters depending on the accelerometer outputs, used for the definition of the movement. The impact detection is calculated by using vertical accelerometer output or angular rate measurements. Finally, a fall alarm occurs when all the test conditions are true.

This method is less complex and therefore requires lowest computational cost and power which is a main advantage as it became easy to implement it for real-time applications [18]. Thresholds are usually calibrated and provided based on simulated fall signals collected from different mass, age, gender, diseases, clinical history... But the fixed threshold-based fall detection algorithm is not flexible for different individuals which is a main factor increasing the number of false positives.

For our algorithm, after a training period while testing different thresholds fixed in previous works and based on a database of acceleration data of simulated falls, we were able to adjust and fix the 2 thresholds of our fall detection algorithm.

B. Fall Detection Algorithm Design

When a person starts falling, the acceleration decreases from 1g (corresponding to Earth gravity) to around 0,5g [19]. We chose then 0,75g as the lower threshold in our algorithm. And we also noticed that upon the impact with the ground, a short strong increase in the acceleration [20] is measured and this sudden change in acceleration corresponds to the upper threshold equal to 2g. Following the detection of the large impact, the patient's orientation is then monitored to determine if a fall followed by a long-lie has occurred. A person was assumed to be in a lying position if the body tilt θ is lower than 45°(front fall) and greater than 135°(back fall).

Beside this, the parameters tested in the proposed algorithm are mainly time interval length and large acceleration threshold. It was necessary to consider that changes in angles and accelerations can be part of everyday activities such as picking something from the floor, stretching or sitting down and therefore must not be selected as a fall or it will cause a larger number of false alarms. That's why we selected the 2 thresholds based on the basic trade-off between detecting the maximum number of falls and avoiding false positives. In fact, the major problem of fall detection algorithms is false positives and if a system has an important false positive rate, it will disturb users with frequent false alarms which is very uncomfortable and decreases the acceptance and reliability of the system.

In order to improve reliability of the system, we take into consideration the patient's context by analyzing what happened before, during and after a fall. Four types of falls were evaluated: back, front, left and right fall. And to avoid false positives, we used adaptive thresholds depending on the previous position of the patient so that the fall detection system is more generic and allow a tailored response for each specific patient.

For our fall detection algorithm depicted in Fig.3, we chose to use as the first parameter the sum-vector (SV) of the 3-axes acceleration to detect the different phases of the fall. The use of the sum vector of 3-axis accelerometer outputs as the main parameter provides robustness to the system [17].

The second parameter used is the body tilt θ. It corresponds to the tiled angle between the accelerometer y-axis and the vertical direction. SV and θ are determined by equation 1 and 2 respectively.

$$\mathrm{SV}= \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{1}$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{(A_y^2+A_z^2)}}{A_x(i)}\right)\times \frac{180}{\pi} \tag{2}$$

where A_x , A_y and A_z corresponds to the x-axial, y-axial and z-axial accelerations respectively.

Threshold-based Algorithm for Fall Detection

1. Initialization
Initial Position of the patient: $\theta_{pr,0}$=0° and $\theta_{pr,1}$=90°
Falls_Detected = 0
Start ADXL345 and Registers Instantiation
I²C Zybo controllers configuration
I²C Initialization
2. Collect Data : a_x, a_y, a_z
    for i=0 **to** n do
Acquire data from ADXL345: Read_Accelerometer
                                                  Zybo_iic_write
                                                  Zybo_iic_read
3. Calculate SV
Read data from accelerometer: ReadGAM
                                                  ReadAccel
4. Threshold Comparison
    if SV<th1 : Possible Fall
    delay
5. Compute SVM
    if SVM>th2 and SV<th1
6. Calculate Body tilt angle
    if ($\theta_{pr,0} > \theta_{th,1}$ or $\theta_{pr,1} < \theta_{th,2}$)
        then Fall_Detected ++
                Blink LEDs
    end for

Fig. 3: Pseudo-code of the fall detection algorithm

The first threshold detects the FREE FALL phase and the second one corresponds to the IMPACT. We added to this algorithm a posture recognition phase in addition to thresholds comparison.

Most fall detection systems using accelerometers around waist could easily assume that before and after a fall, there is a significant change in orientation. This slight change in orientation can be defined as a difference of 45° between the gravity position measured by the accelerometer. So it will be interesting to detect the INACTIVITY phase after the fall which corresponds to a lying posture by checking if the body tilt θ exceeds 45° in both sides [22].

In this algorithm, as there are various calculation operations and comparisons, we managed to write a C function for each step to avoid computational complexity. It was mandatory to use pre-defined functions of Xilinx to acquire data from the ADXL345 through I²C communication protocol and to write the functions to read and write from ADXL345 registers (iic_write and iic_read) to send the data to the ARM processor for analysis, calculation and thresholds comparison. Then we write in the main program 2 functions ReadGAM and ReadAccel for data calibration and to multiply the raw data by 0,0039 to convert it in g for thresholds comparison. Finally, after SV and θ calculation, if all the conditions (FREE_FALL, IMPACT detected and patient's position = lying) are true, the Zybo board LEDs blink and the counter is incremented.

V. Implementation & Results

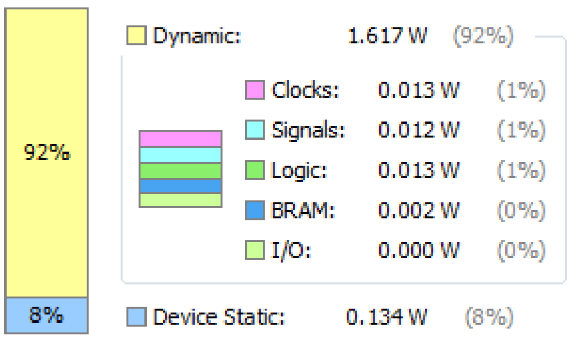
A. Implementation Results

For the experimentation, we first designed a fall detection system with the Zynq Z-7010 FPGA, a timer, an AXI Performance Monitor (APM) and we configured the I^2C interface to communicate with the 3D accelerometer ADXL345. The configuration of the I^2C interface on the Zybo Pmod requires to follow different steps so that data can be acquired from the accelerometer through I^2C for reading and writing. This block design has been then synthesized, simulated and implemented using Vivado. The resource utilization on the Zybo board is summarized in TABLE I and the power consumption results are depicted in TABLE II.

TABLE I. Ressource utilization on the Zynq Z-7010

Resource	Utilization	Available	Utilization %
FF	3815	35200	10.84
LUT	4314	17600	24.51
Memory LUT	352	6000	5.87
I/O	4	100	4.00
BRAM	1	60	1.67
BUFG	1	32	3.12

TABLE II. Power consumption on the Zynq Z-7010



B. Hardware Execution on the Zynq Z-7010

In our algorithm, data are collected in real-time from a 3D accelerometer connected via wires to the Zybo board viva Pmod connector. Data are read from ADXL345 through I^2C communication protocol. In fact, the ADXL345_POWER_CTL and ADXL345_MEASURE registers are on so that measuring and reading accelerometer data can start. It is mandatory to write through I^2C to all the necessary registers of the ADXL345 so that accelerometer outputs can be acquired and analyzed by the fall detection algorithm. The appropriate ADXL345 registers are configured to fix the resolution and data format. Then the writing and reading modes are activated through I^2C to start sending data from the 3D-accelerometer to the Zybo board for data analysis and fall detection decision.

For every module of the architecture, a C function have been developed for reading and writing through the I^2C from the accelerometer to the Zybo, calculation operations and threshold comparison. The developed C files have been executed and simulated on the ARM A9 processor combined to the FPGA as a standalone application using Xilinx SDK. Every time a fall occurs, a message is sent alerting a fall has been detected and Zybo board LEDs are blinking. This first implementation on the Zybo board validated our fall detection algorithm and showed prominent results in terms of execution time (TABLE III) and accuracy of falls detected. All the modules of the proposed prototype have been tested separately, the communication have been configured both for the HW and SW sides and finally the proposed prototype has been validated by simulation and in real time on the Zybo board.

TABLE III. Execution Time Summary

Execution Time	Latency
8,62 ns	3 cycles

As we added an APM in the Vivado Project, we were able to evaluate the critical partitioning trade-offs between the ARM Cortex A9 and the programmable fabric for a variety of different traffic scenarios. In fact, with this APM perspective, we were able to do the architecture exploration and to evaluate the PS performance. The performance analysis showed the following results (TABLE IV).

TABLE IV. ARM- A9 CPU Utilization

CPU Utilization (%)	99,14%
CPU Instructions/cycle	0,53
L1 Data Cache Access	783,57 M

To evaluate the proposed fall detection algorithm, we calculate sensitivity and specificity of our fall detection algorithm. Sensitivity (SE) is the capacity of detecting real falls (Eq. 3): it's defined as the ratio between the number of falls properly detected (true positives) and the falls that really happened (true positives plus false negatives).

$$SE = \frac{TP}{TP + FN} \quad (3)$$

Specificity (SP) is the capacity to filter false alarms and corresponds to the ratio between fall properly discarded (true negatives) and the total number of discarded actions (true and false negatives) (Eq. 4).

$$SP = \frac{TN}{TN + FP} \quad (4)$$

TABLE V. SENSITIVITY AND SPECIFICITY OF THE PROPOSED FALL DETECTION ALGORITHM

	{ START OF FALL + IMPACT}		{START OF FALL + IMPACT + POSTURE}	
	SE (%)	*SP (%)*	*SE (%)*	*SP (%)*
Proposed Algorithm	89,5	97	96	97,5

C. Discussion

It is clearly seen that this first implementation on Zynq SoC using Vivado and Xilinx SDK showed an important gain in terms of execution time and power consumption. Beside this, the specificity and sensitivity values are very prominent even if it will be more interesting to use adaptive thresholds depending on a specific patient and considering real-life conditions (environment, indoor or outdoor context, different activities of daily life...). This first implementation validated the proposed algorithm but more work has to be done to classify the data acquired and to improve the accuracy by adding a recognition posture phase.

Additional changes have to be done in the future prototype to have experimental values from real falls and not simulated falls. To be more accurate, we have to use real-time outputs from the tri-axial accelerometer from patients and not only simulated falls, add feature extraction and data classification steps in the fall detection algorithm and implement it again on the Zybo board. The HW implementation has to be evaluated using the performance analysis in Xilinx SDK and depending on the results, more efforts will be done to parallelize the calculation operations in order to increase the efficiency of the proposed fall detection prototype. The future results will be detailed in next papers.

VI. CONCLUSION AND FUTURE WORK

We have presented in this paper, a fall detection prototype based on a threshold-based algorithm: the lower one which detects the start of the FREE FALL and the upper which corresponds to the IMPACT detection. The implementation was performed on a Zynq Z-7010 board connected to a 3D accelerometer ADXL345 via I^2C communication protocol.

The proposed system senses the body posture change, acquires data through I^2C interface, analyzes the data according to the different modules implemented and decides whether a patient is falling. The development on the Zybo board using Vivado and Xilinx SDK permitted a high flexibility and re-configurability while designing the system. There is also a gain in terms of execution time, power consumption and number of latency cycles. This implementation on the Zybo board is not yet optimized in terms of resource utilization and architecture communication.

The Zynq board has an FPGA HW on the same IC as the processor which offered the advantage to send the raw data acquired from the accelerometer to the ARM A9-CPU quickly and efficiently. For fall detection, the memory spaces available in FPGAs offers a huge advancement compared to microcontrollers for data samples storing.

The proposed prototype is still under testing to improve its performance and accuracy in terms of avoiding false positives. Testing our system in real-life conditions will make it a more robust and automated system with higher acceptance. We will have to personalize the system according to the user profile and environment. A HW implementation, using Vivado HLS has to be done to evaluate the proposed approach to use a Zybo board for fall detection of elderly people.

ACKNOWLEDGEMENT

This research and innovation work is carried out within a MOBIDOC scholarship funded by the EU under the PASRI project and in collaboration with Telnet Holding company.

REFERENCES

[1] A. Pantelopoulos and N.G. Bourbakis, "A Survey on Wearable Sensor-Based Systems for Health Monitoring and Prognosis", IEEE Transactions on Systems, Man, and Cybernetics —Part C: Applications And Reviews, Vol. 40, N°. 1, January 2010.

[2] Y.Yang and X.Zhao, "Development of a Fall Detection Algorithm based on a Tri-axial Accelerometer", 4th International Conference on Biomedical Engineering and Informatics (BMEI), IEEE 2011, p.1371-1374.

[3] S.Zhang, P.McCullagh, C.Nugent and H.Zheng, "A Theoretic Algorithm for Fall and Motionless Detection", 3rd International Conference on Pervasive Computing Technologies for Healthcare,1-3April 2009,PERVASIVE HEALTH 2009.

[4] S.Abdelhedi, R.Bourguiba, J.Mouine and M.Baklouti, " Fall Detection FPGA-Based Systems: A Survey", AUSMT Journal, Accepted.

[5] T. Ahola, P. Korpinen, J. Rakkola, T. Ramo, J. Salminen and J.Savolainen, "Wearable FPGA Based Wireless Sensor Platform", Proceedings of the 29th Annual International Conference of the IEEE EMBS, Cité Internationale, Lyon, France, August 2007, pp.2288-2291.

[6] M. Neggazi, A. Amira and L. Hamami, "A Wireless Reconfigurable System for Falls Detection", 11th International Conference on Information Sciences, Signal Processing and their Applications: Main Tracks, 2010, pp.77-82.

[7] M. Grassi, A. Lombardi, G. Rescio, M. Ferri, P. Malcovati, A. Leone, G. Diraco, P. Siciliano , M. Malfatti and L. Gonzo, "An Integrated System for People Fall-Detection with Data Fusion Capabilities Based on 3D ToF Camera and Wireless Accelerometer", IEEE SENSORS 2010 Conference, 2010, pp.1016-1019.

[8] A. Belbachir, M. Litzenberger, S. Schraml, M. Hofstatter, D. Bauer, P. Schon, M. Humenberger, C. Sulzbachner, T. Lunden, T. M. Merne, "CARE: A dynamic stereo vision sensor system for fall detection" In Proceedings of the International Symposium on Circuits and Systems, Seoul, Korea, 20–23 May 2012, pp. 731–734.

[9] A. Ait Si Ali, M. Siupik, A. Abbes, F. Bensaali and P. Casaseca-de-la-Higuera, " HLS Based Hardware Acceleration on the Zynq SoC: a case study for fall detection system", IEEE/ ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, Qatar, 10-13 Nov 2014, pp.685-690.

[10] V. Bianchi, F. Grossi, G. Matrella, I. De Munari and P. Ciampolini, " A Wireless Sensor Platform for Assistive Technology Applications", 11th EUROMICRO Conference on Digital System Design Architectures, Method and Tools, Parma, Italy, 3-5 Sept 2008, pp.809-816.

[11] Zynq-7000 All Programmable SoC, Technical Reference Manual, Xilinx, September 2014.

[12] Y. Xiuping, L.Jia-nan and F. Zuhua, "Hardware Design of Fall Detection System Based on ADL345 Sensor", 8th International Conference on Intelligent Computation Technology and Automation, 2015, pp.446-449.

[13] J.Yuan, K.Kiong Tan, T. Heng Lee and G. Choon Huat Koh, "Power-Efficient Interrupt-Driven Algorithms for Fall Detection and Classification of Activities of Daily Living", IEEE Sensors Journal, VOL.15, N°3, March 2015.

[14] ADXL345 Preliminary Technical Data, Analog Devices, 2008.

[15] ZYBO Reference Manual, Revised February 14 , 2014

[16] Zhibo Pang, Qiang Chen, Lirong Zheng, “A Pervasive and Preventive Healthcare Solution for Medication Noncompliance and Daily Monitoring”, 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies, 2009, ISABEL 2009.

[17] Kangas M, Konttila A, Lindgren P, Winblad I, Jamsa T , “Comparison of low-complexity fall detection algorithms for body attached accelerometers”, Gait&Posture 28, 2008, pp.285–291.

[18] A.Sorvala, E.Alasaarela, H.Sorjova and R.Myllyla, "A Two-Threshold Fall Detection Algorithm for Reducing False Alarms", 6th International Symposium on Medical Information and Communication(ISMICT), California,USA,25-29March,2012,pp1-4.

[19] L.Ren, W.Shi Z.Yu and J.Cao, "ALARM: A Novel Fall Detection Algorithm based on Personalized Threshold", 17th International Conference on E-health Networking, Application and Services (HealthCom), 2015, pp.410-415.

[20] H. Gjoreski, M. Luštrek and M. Gams, “ Accelerometer Placement for Posture Recognition and Fall Detection”, 7th International Conference on Intelligent Environments, 25-28 July 2011, pp.47-54.

[21] Yoan Charlon, "Conception de dispositifs électroniques portés pour le suivi de l'état de santé des personnes âgées", PhD thesis, Department of Electrical Engineering, University Toulouse 3 Paul Sabatier, France, Sept.2014, pp.58-70.