ADVANCED MIGRAINE PREDICTION HARDWARE SYSTEM

Kevin Henares

Dept. of Computer Architecture and Automation
Complutense University of Madrid
Madrid, Spain
khenares@ucm.es

Josué Pagán

Electronic Engineering Department
Universidad Politécnica de Madrid, Spain
CCS-Center for Computational Simulation
Madrid, Spain
j.pagan@upm.es

José L. Ayala

Dept. of Computer Architecture and Automation
Complutense University of Madrid
CCS-Center for Computational Simulation
Madrid, Spain
jayala@ucm.es

José L. Risco-Martín

Dept. of Computer Architecture and Automation
Complutense University of Madrid
CCS-Center for Computational Simulation
Madrid, Spain
jlrisco@ucm.es

ABSTRACT

Migraine is one of the most disabling neurological diseases. Its prevalence reaches 15% of the population in developed countries and lead to high economic costs for private and national health systems. There is no cure for the migraine yet, and currently patients combine a basal medication—to reduce the number of crises—with rescue medication to mitigate the pain once this has already started. Objective and accurate migraine prediction turns necessary in order to increase the effectiveness of the current treatments. To tackle this problem authors demonstrated in previous works the feasibility of the migraine modeling and presented a DEVS-based model for migraine prediction in a real time framework. This work presents the first step to bring migraine predictive models and simulations to real devices for the first time in the literature. To do this, the DEVS-based model has been ported to a robust hardware description language (VHDL) to emulate the system under real conditions and hardware constraints to be implemented into an FPGA. This paper demonstrates that an initial DEVS specification can facilitate the design, allowing a straightforward implementation and validation, propitiating the next and final stage: the physical implementation of a predictive migraine monitoring device.

Keywords: migraine, predictive models, failure detector, robust system, FPGA, VHDL

1 INTRODUCTION

The Internet of Things (IoT) is a paradigm that brings innumerable opportunities to control and improve directly or indirectly many different aspects of our lives, as healthcare and personal well-being (Zheng et al. 2014). Small wireless low-power consuming monitoring devices have become rapidly a part of our daily life. There is an increasing interest in the industry for applications related to healthcare and personal well-being. In this area, it is worth to mention activity recognition applications as one of the most attractive applications for industry (Barshan and Yüksek 2014), as these are simple and require few resources (sensors and processing). However, other healthcare fields—such as neurological diseases—require further development to be useful for the industry and the market.

Event detection in neurological diseases is in its early stages, even though, some companies and research groups have proved that it is possible to use unobtrusive wireless monitoring devices for different purposes, such as epileptic seizures as MNJ-Neuroserveis[1] does with headphones or Heldberg *et al.* did in (Heldberg et al. 2015) using smartwatches for monitoring epilepsy episodes, as an example. Physical implementation, validation and the pressure of the fulfillment of the time-to-market constraint is a time consuming and economically expensive process. To tackle this problem, the authors recently developed an advanced Modeling and Simulation (M&S) framework to help to validate the behavior of a migraine attack predictive system (Pagán et al. 2016, Pagán et al. 2017). As a movement forward in this direction, this paper presents the first step to bring migraine predictive models and simulations to real devices.

The migraine is a neurological disease that causes strong headaches. It is considered one of the most disabling neurological diseases and affects around 10% of population worldwide (Lipton and Scher 2001) and 15% in Europe (Stovner and Andree 2010). It affects more women than men, and in the short term, patients face changes in their wellness and their daily activities. On a long-term basis, the migraine is a social problem that deteriorates all areas of patients' life: work/school and personal relationships.

A migraine is more than pain; a cascade of neurological processes precede a long lasting pain—hours or days—that might be followed by a hangover. Additionally, migraine patients fall into overmedication. Currently patients do not have any objective and accurate mechanism to detect the onset of pain. Some migraine sufferers experience premonitory symptoms (objective and unspecific symptoms such as nauseas or yawns that may occur from three days to hours before the pain starts (Giffin et al. 2003)). Other patients also suffer from auras, which are objective and specific disturbances—such as losing vision—-and occur within 30 minutes before the onset of pain.

Migraine sufferers are more prone to suffer from other diseases such as fatigue, anxiety or cardiovascular problems, which leads to high costs for private and national health systems. In Europe, it is estimated that migraine leads to direct and indirect costs of €1,222 per patient per year (Linde et al. 2012). A study case presented in (Pagán et al. 2017)—for 2% of European migraine sufferers—demonstrated that society would benefit from savings of more than € 1272 million based on a migraine prediction system with 76% accuracy.

It is difficult to estimate the onset of pain to make the intake of drugs effective. In addition, the time response of the pharmacokinetics of the drugs (the mechanisms of absorption and distribution of substances in an organism) does not match the long times of the vague predictive symptoms, or the short times of the urgent auras. So, most migraine sufferers wait till the onset of pain to take the rescue medication. The delayed intake reduces the effectiveness of the treatment. Thus, prediction of the onset of a migraine attack will help patients to stop the pain.

Prediction modeling of migraines has been demonstrated by the authors in previous works (Pagán et al. 2015, Pagán et al. 2016). Also, it has been simulated a real time monitoring device for migraine prediction that uses hemodynamic variables to predict the pain (Pagán et al. 2017). Hemodynamic variables are those controlled by the Autonomous Nervous System (ANS), and these change when a migraine occurs.

In these works the predictive modeling of the migraine symptoms was shown feasible through the analysis of the changes in four hemodynamic variables controlled by the ANS: skin temperature (TEMP), electrodermal activity (EDA), oxygen saturation (SpO2) and heart rate (HR). Moreover, it was demonstrated the possibility

[1]MJN: http://mjn.cat/?lang=en (accesed February 2018=

to model subjective pain as a normalized quantification based on a combination of Gaussian curves. These two works addressed the problem through the use of classic modeling methods like state-space systems in (Pagán et al. 2015) and heuristics like Grammatical Evolutionary (GE) algorithms in (Pagán et al. 2016). In this work we use the same basis to bring the virtual simulation into a real hardware implementation.

In (Pagán, Moya, Risco-Martín, and Ayala 2017), the aforementioned predictive models were encapsulated into a M&S architecture based on Discrete Event System Specification (DEVS). Such framework was conceived to demonstrate that in theory, a predictive hardware device would work in a real environment. It included complex behaviors like the recovering of the signal after failures in sensors. The next natural step would be the implementation of such architecture into a real hardware device, to definitely check the emulation of the system under both hardware and real time constraints. To this end, this paper shows the validation under real conditions and hardware constraints of the implementation of the whole architecture into a Field Programmable Gate Array (FPGA). This paper shows the results behavioral emulation of the hardware, and compares them with the virtual-domain DEVS simulator. We also demonstrate that a complex DEVS model can be transformed into a hardware specification on a straightforward basis.

This paper is organized as follows: specifications on DEVS formalism, hardware synthesis and the models used are included in Section 2. The hardware implementation of the migraine prediction system is shown in Section 3. The behavioral emulation of the system is described in Section 4. Finally, the conclusions are drawn in 5.

2 BACKGROUND

This section briefly describes the required background definitions and references to the related literature of i) the basis on which this article is based: DEVS M&S, and ii) the fundamentals for the hardware synthesis implementation.

2.1 Discrete Event System Specification (DEVS)

DEVS is a modeling and simulation formalism that allows us to simulate models under a unified modeling and simulation theory in real time, soft-real time and virtual time (Zeigler et al. 2000). This formal specification makes DEVS a good choice to validate the model before the real implementation of a physical device is performed.

The physical implementation of the migraine predictive system of this paper is inspired on the DEVS complex model presented in (Pagán et al. 2017). Thus, our work is a straightforward transformation of a formal yet intuitive M&S architecture into a real hardware design. All the atomic and coupled DEVS models and interconnections introduced in (Pagán et al. 2017) have been translated to a hardware description language to synthesize the final system.

2.2 Hardware synthesis

FPGAs are integrated circuits whose internal structure can be configured using Hardware Description Languages (HDL). HDL are specialized programming languages that allow to define the structure of electronic circuits and how they have to behave.

FPGAs are composed of arrays of logic blocks that can be arranged through reconfigurable interconnections. Each one of these blocks has multiple logic cells with basic circuits, as adders, flip-flops or multiplexors. These blocks are connected to execute certain tasks, that will be aggregated into components. These components can be parametrized and replicated, so they can work in parallel without any interference between them. This flexibility allows the implementation of high complex custom designs and facilitates the subsequent introduction of additional modules.

The process of FPGA systems design and validation goes through different phases: (i) Specification of the subsystems and its interconnections, (ii) Behavioral simulation (emulation) using test benches, (iii) Synthesis, process where the defined system is translated to a hardware structure, (iv) Mapping of the inputs and outputs of the system with the FPGA physical pins, and (v) Generation of the bitstream that contains all the information about how the FPGA blocks have to be wired. This bitstream will be loaded into the FPGA.

We have chosen VHDL (VHSIC Hardware Description Language) as the description language since it allows, with the suitable tools, to describe and verify the system before the synthesis. A VHDL design implies the creation of different entities (that specify the inputs and the outputs of a subsystem) and its architectures (that specify the behavior of the entities). After that, these entities can be instantiated (being able to pass parameters to customize them) and interconnected. In our case, we will use the DEVS implementation as a template to easily code it.

3 FROM DEVS TO VHDL

After validating the system in the DEVS environment, a VHDL implementation has been developed. Although the hierarchical architecture of DEVS guides and facilitates this implementation, several changes has been made with respect to the simulation and the mathematical models. In particular, our system contemplates two implementations of mathematical models regarding i) the migraine predictive modeling and ii) a history-based signal repair.

3.1 Migraine predictive modeling

The migraine predictive system presented by the authors in (Pagán et al. 2016) and (Pagán et al. 2017) included the possibility of using different types of predictive mathematical models such as Grammatical Evolutionary algorithms and state-space models respectively. In this work we have considered the later, a Subspace State Space System Identification (N4SID) model to generate the prediction of onset of pain of a migraine crisis.

N4SID is an state-space based algorithm (Van Overschee and De Moor 1994). It describes immeasurable states and specifies differential equations that relate future outputs with current and past inputs. It is formally described in Eq. 1 and Eq. 2.

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{1}$$

$$y_k = Cx_k + Du_k + v_k \tag{2}$$

u_k are our $U = 4$ hemodynamic inputs—body temperature, sweating, heart rate, oxygen saturation—at time k. y_k is the output at time k. In this project it will be the predicted pain level. A, B, C, and D are the state-space matrices. v_k and w_k represent white immeasurable noises. More details in (Pagán et al. 2015).

3.2 History-based signal repair

If a sensor breaks, prediction worsens. Being able to temporary repair a signal keeps the predictive system active till the error is fixed. Precedent DEVS simulator contemplated a *Sensor Status Detector* with a subsystem for signal repairing based on Gaussian Process Machine Learning (GPML). Because of the limi-

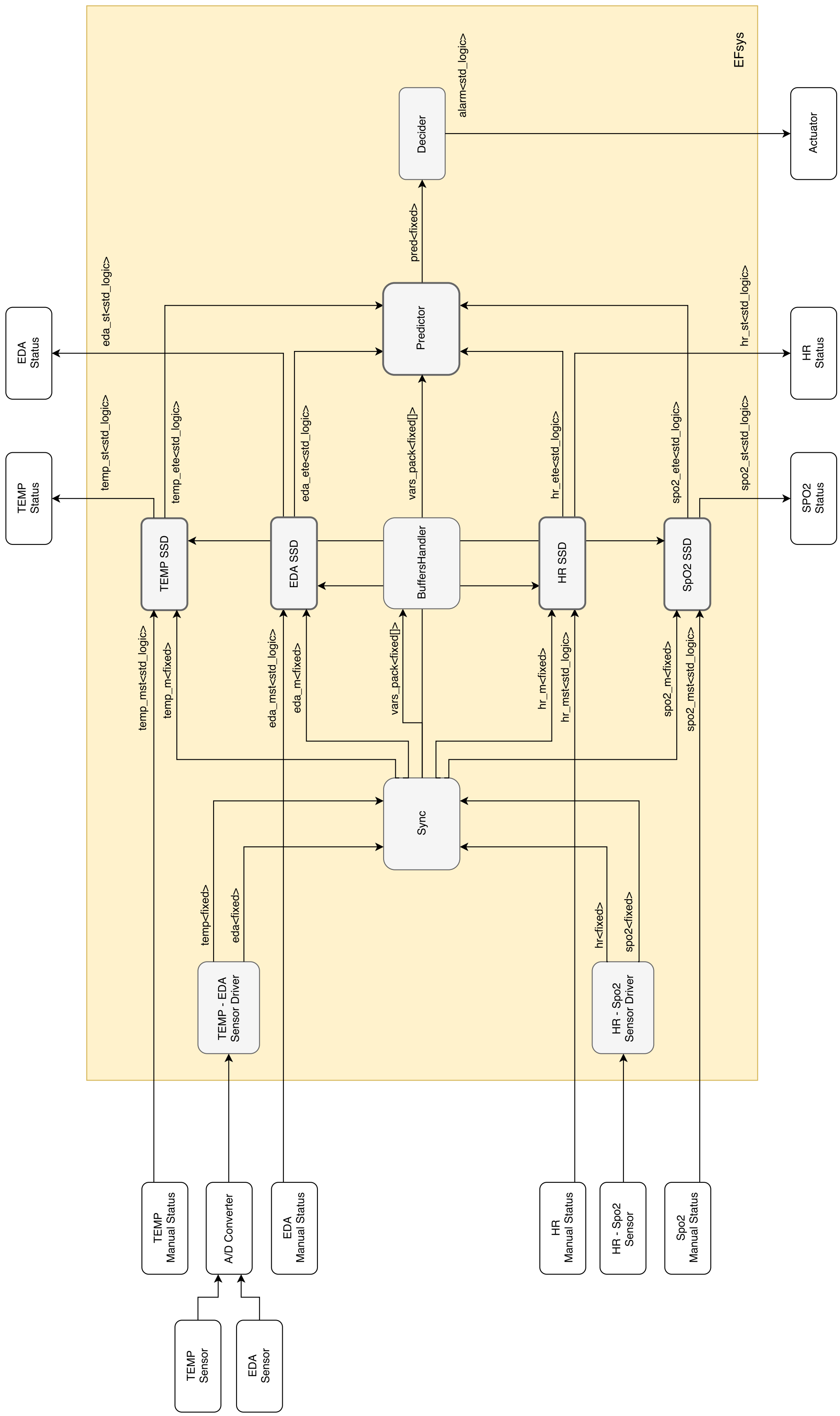


Figure 1: Root component of the migraine pruning system implemented in an FPGA with VHDL.

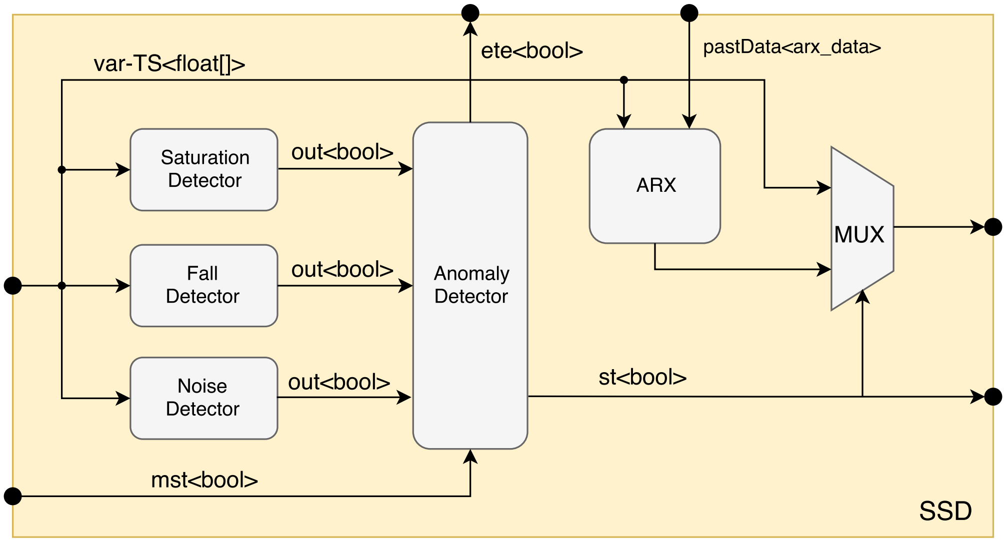


Figure 2: `SSD` module. It detects errors in the input signal and recovers it (with the `ARX` module) while it is not restored.

tations of VHDL, in this paper we have explored simpler ways to perform this task, and an Auto Regressive model with eXogenous inputs (ARX) model is used to repair the signal (in case of sensor failure).

Autoregressive (AR) models assume that current values of a variable depend on a polynomial combination of its own past values. In addition, ARX models consider the influence of past data of exogenous (external) variables as well. As our hemodynamic variables relate each other through the ANS, we contemplate these polynomial models as Eq. 3 shows:

$$\begin{gathered} y[k] + a_1 * y[k-1] + \ldots + a_{n_a} * y[k-n_a] = \\ b_1 * u[k-n] + \ldots + b_{n_b} * u[k-n_b-n+1] + e[k] \end{gathered} \tag{3}$$

$y[k]$ is the output—one hemodynamic variable—at time k. n_a and n_b are the number of poles and, zeros plus one, of the polynomial. n is the number of input samples that occur before the input affects the output. $y[t-1], \ldots, y[t-n_a]$ are the previous outputs on which the current output depends. $u[k-n], \ldots, u[k-n-n_b+1]$ are the previous exogenous inputs on which the current output depends—the remaining non-damaged hemodynamic variables. $e[k]$ is a white-noise disturbance value.

In this work, both the N4SID and the ARX algorithms have been computed using the System Identification Toolbox of the MATLAB software[2].

Next, we describe the hardware model, which is equivalent to the DEVS model published in (Pagán et al. 2017). With the exception of the floating point operations and some slightly differences detailed below, each DEVS transition and output function has an equivalent representation in VHDL.

3.3 Hardware model

A general view of this system is shown in Figure 1 (DEVS coupled models are shown with a thicker border). As inputs, it uses the four hemodynamic variables (body temperature, sweating, heart rate and oxygen saturation) and four buttons to report the restoration of each one of them. As outputs, it has four indicators that indicate the status of the sensors and another one to warn of the predicted onset of the pain.

[2]MATLAB 2015. version 8.5.0.197613 (R2015a). Natick, Massachusetts, The MathWorks Inc.

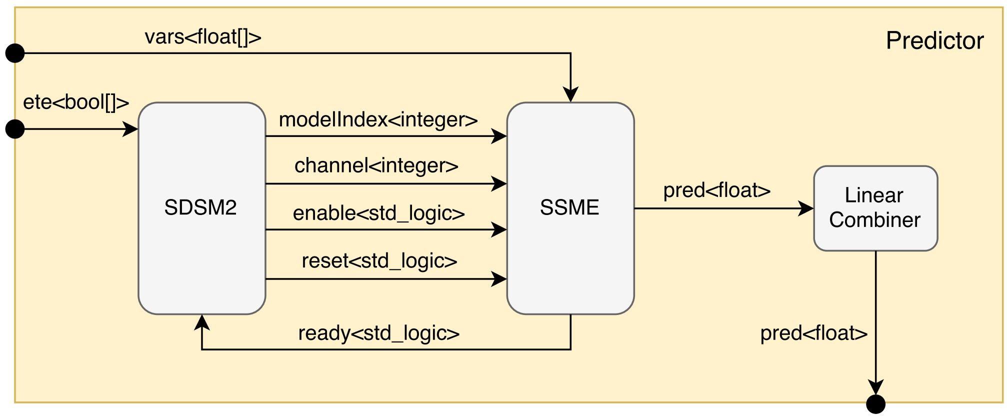


Figure 3: `Predictor` module. It generates predictions of new pain episodes. The `SDMS2` module controls the predictions generation and the `LinearCombiner` group them together to produce a single output.

The system has the following main components (a more detailed description of them can be read in (Pagán et al. 2017)):

- `Drivers`: they read the input of the sensors (interpreting the appropriate protocols), calculate the corresponding physical magnitude, convert the obtained measurements to the appropriate data type and send them to the synchronizer (`Sync`). Since this paper does not cover the tests with real sensors, its implementation will be added as future work.
- `Sync`: it packs the input values of the different sensors into a unified data structure. Each minute it receives a pulse signal and it averages the values received for each variable . It receives 180 samples of temperature and sweating and 60 of heart rate and oxygen saturation per minute. That information is sent to the Sensor Status Detectors (`SSDs`).
- `SSDs` (Figure 2): they check if different types of errors are present in the input signal (saturation, fall or noise). If one of them is detected, the status signal raises, so that the patient can restore the sensor. While the sensor is not restored, the system tries to repair the signal temporarily. An `ARX` module is used for that purpose. It generates estimations of the input using previous samples of both the controlled variable and the exogenous ones. When it has passed too much time since the error detection, an *Elapsed Time Exceeded* (ETE) signal is raised. That signal points out that the variable implied can not be used reliably to generate predictions, so the `Predictor` module will discard it. The `SSDs` are connected to the output of the `Sync` , instead of going after the `Drivers` . That is because error detection and signal repair capabilities have to operate over data separated by minutes. In the DEVS simulation the input data was already stored by minutes. Conversely, in the VHDL implementation the synchronizer is in charge of doing this task.
- `BufffersHandler`: it stores and handles the previous inputs of the system. It is convenient to centralize the management of the previous data, used by the `ARX` modules (in the SSDs).
- `Predictor` (Figure 3): it generates a prediction of occurrence of a new pain episode. For that, it contains 5 sets of models (Pagán et al. 2015). Each one of them is related with a different group of three or four input variables. Each set has 3 predictive models whose results are combined to refine the prediction. These models are trained for different prediction horizons. In this way, when a sensor fails or recovers, the models used to predict change. The system only can predict when all the variables are fine or when there is an error in one of them. With less than three variables the prediction is not considered representative and is not supported by the system. The management of the state-space models is done by a unique model in the Predictor (`SSME`). In this way, the `SDMS2`

module is responsible for selecting the correct models and requesting the generation of the three predictions, which will be carried out sequentially.

- `Decider`: activates the alarm when the output generated by the `Predictor` module exceeds a certain threshold (trained previously with several hours of data).

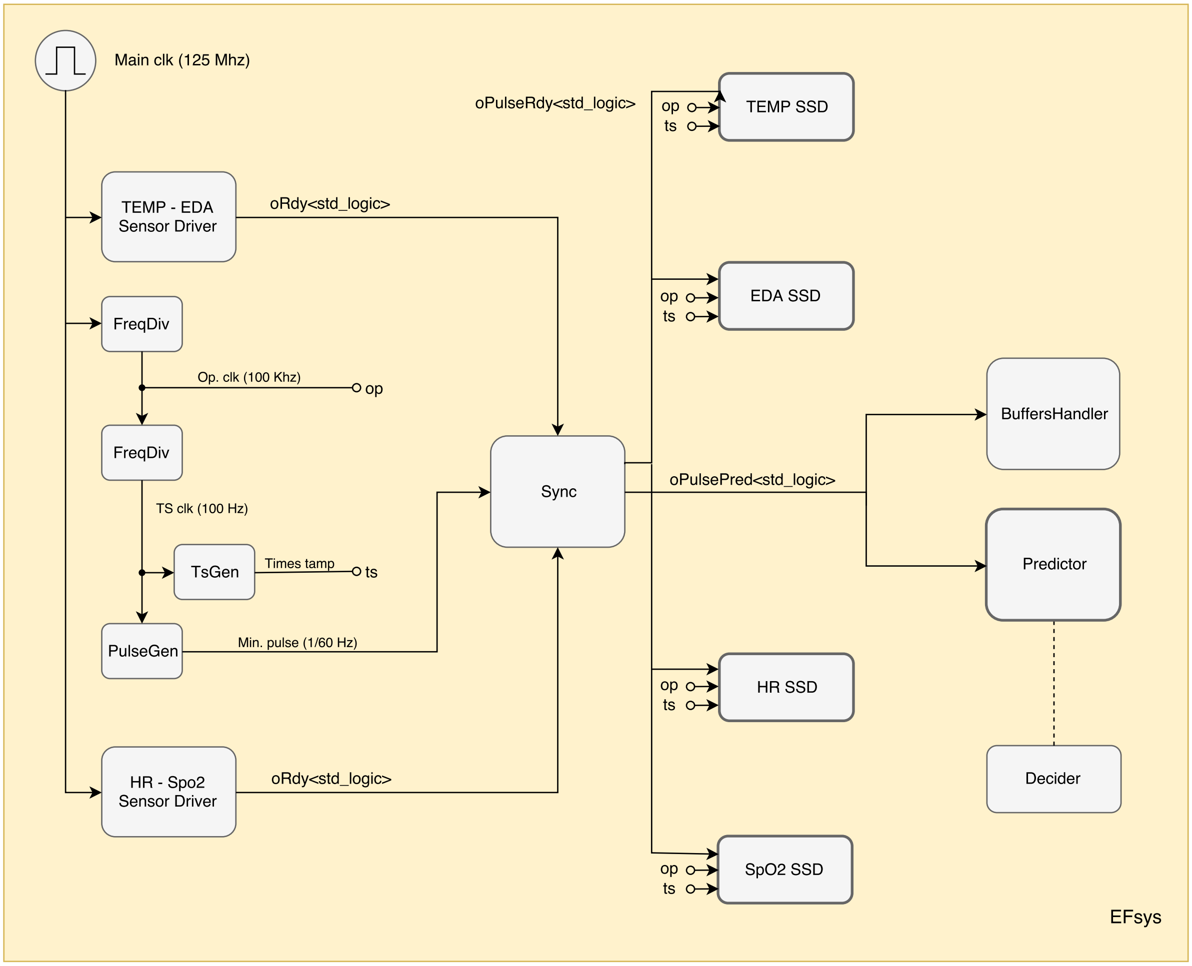


Figure 4: Clock and pulse signals used to synchronize the prediction system.

System synchronization is controlled by a set of clock and pulse signals. These can be seen in Figure 4 and are the following:

- Main clock: reference clock used in the `Drivers` for control the communications and in the generation of all the remaining clock and pulses.
- Operations clock: used for the synchronous components of the system to control its operation (except `Drivers`). The clock frequency is 100 KHz.
- Timestamp clock: used to generate pulses each minute. That pulses are used to cause the `Sync` to generate new packets.
- `Drivers` pulses: used to notify the `Sync` of a new reading.
- `Sync` pulses: after the preparation of a new packet a pulse is generated to communicate that fact to the `SSD`. Later, it waits a given number of cycles and generates another pulse that informs the

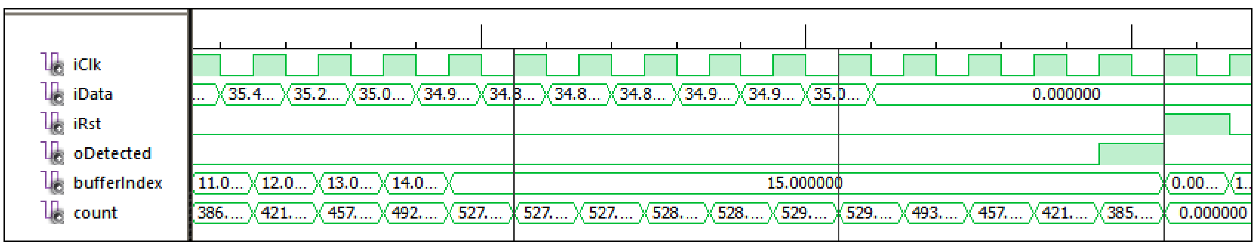
Predictor and BuffersHandler modules of the presence of the new data. That delay corresponds to the processing time spent in the ARX modules to regenerate the signal (if necessary).

To deal with decimal numbers in VHDL the FLOAT32 data type was firstly used, as in the original DEVS model. However, when the system was synthesized it needed many resources to handle the operations. For this reason, a fixed precision data type was used instead. They are used auxiliary fixed data types during all the operation that assure that all the operations can be supported without the appearance of overflows.

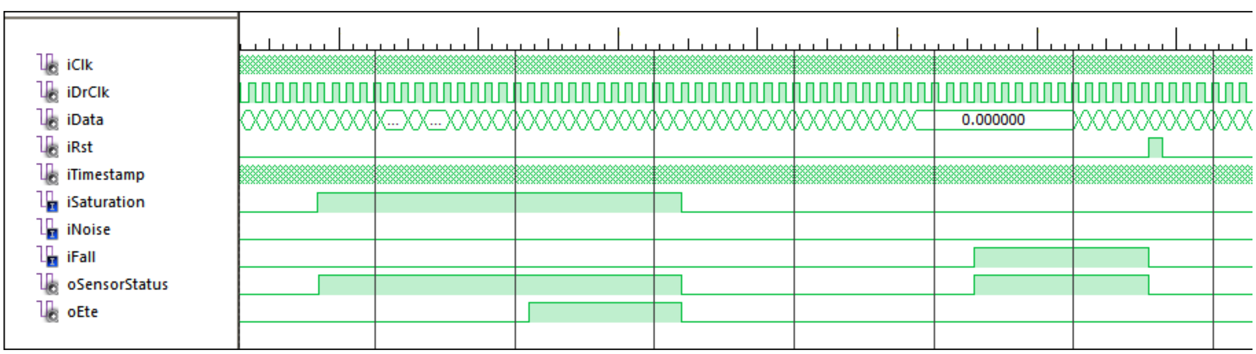
4 SIMULATION OF THE HARDWARE MODEL

To implement the migraine prediction system, the design software Xilinx ISE 14.7 has been used. A Zynq-7000 FPGA has been established as target FPGA (XC7Z010 device, CLG400 package). The components of the system had been emulated using the ISE simulator (ISim).

To validate the system, data acquired from real patients are used. They were monitored in ambulatory conditions with a wireless body sensor network (WBSN), as described in (Pagán et al. 2015). They are saved so that a value for each of the variables is available per minute.



(a) FallDetector, that detects falls in the signal

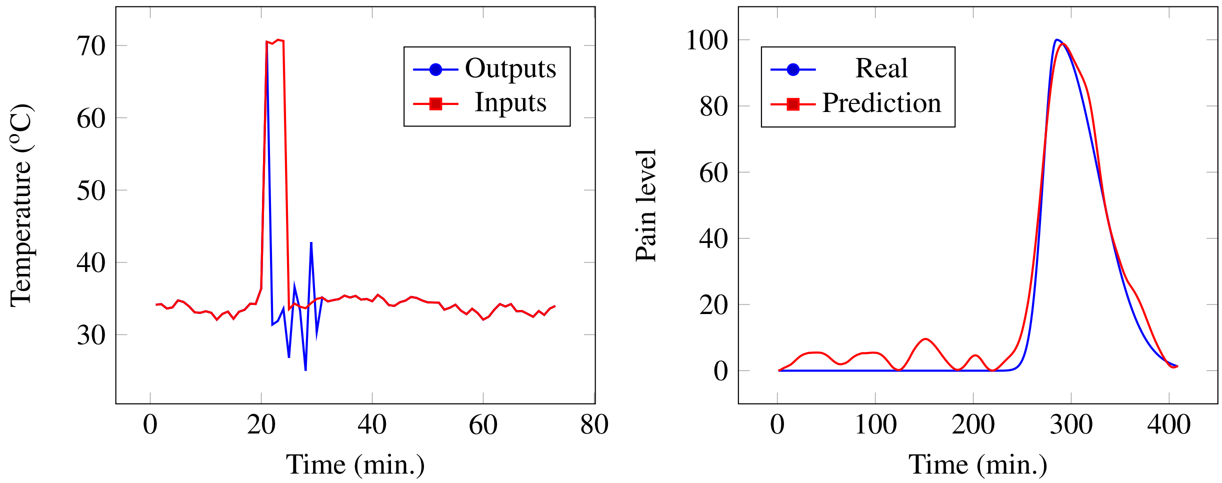


(b) AnomalyDetector, that manages the errors provided by the detectors

Figure 5: Error handling in the *SSD*.

Figure 5 reflects how the errors are managed in the SSD modules. Figure 5a shows how the FallDetector module behaves. When it detects a fall, the signal *oDetected* is raised, notifying the new error. Once this signal has been activated, it will remain in high state until the data in the buffer are valid or the reset signal (*iRst*) is raised. When the module resets, the buffer is emptied and has to be refilled before detecting new errors in the signal. This situation will occur when the patient press a restoration button after being informed of the failure of one of the signals. The SaturationDetector and NoiseDetector modules present a similar behaviour. The outputs of those three detectors are attached to the AnomalyDetector module. That module manages two output signals: *oSensorStatus*, that will be activated when at least one error is detected in the signal, and *oETE*, that will be activated when an error lasts a certain time. Figure 5b shows the aforementioned situation.

When the *oSensorStatus* output of the `AnomalyDetector` is raised, the `ARX` enables. Figure 6a depicts the response of this module to a saturation error. Once the error is detected, the `ARX` module corrects the signal and approximates those values that the variable could take if it were fully operational.



(a) `SSD` outputs after an error detection

(b) Comparison between real and predicted pain levels

Figure 6: Outputs of the predictive system.

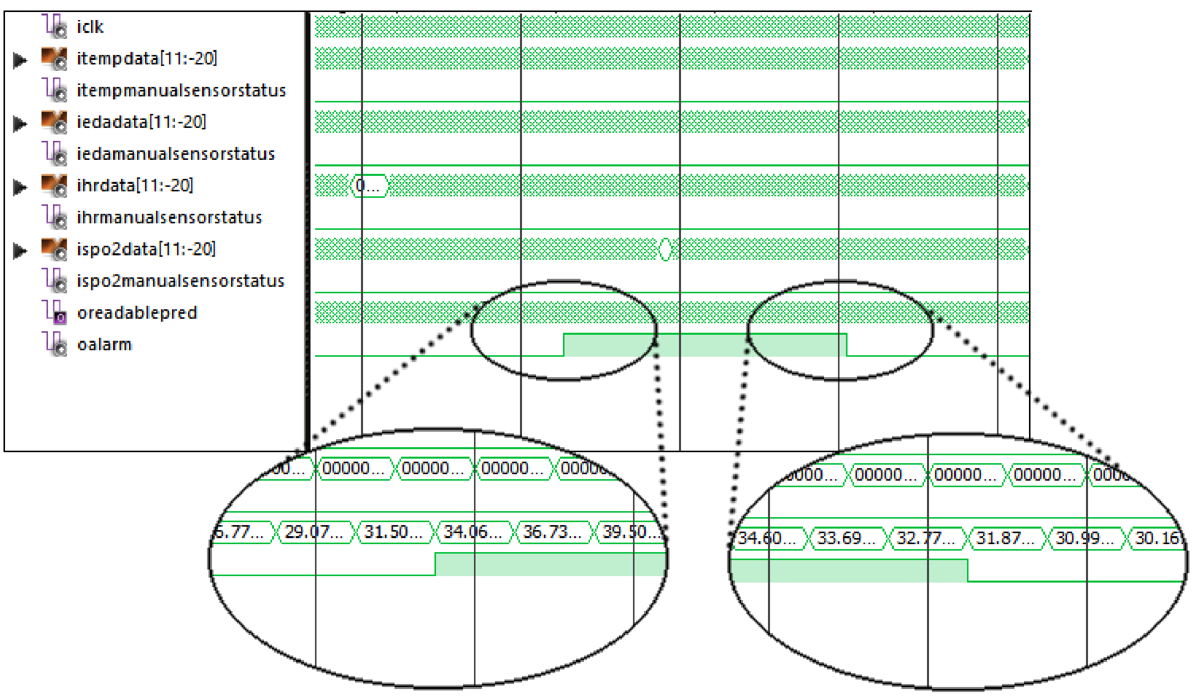


Figure 7: General view of the system simulation.

These signals, after going through the `SSD` modules, go to the `Predictor`. In that module a set of three state-space models are selected based on the *ETE* signals of the four SSDs. After that three predictions are generated, one for each model of the selected model set. Those predictions are unified by making an average. Figure 6b illustrates the predictions generated by the system in response to a known episode. That predictions are compared with the subjective pain curve, generated with patient data. The prediction oscillates over the reference curve and triggers an alarm when the migraine event occurs. For that, it is used a threshold (32 in that example), obtained studying the data previously. When the prediction goes over that threshold it is considered that the pain episode will occur in the next few minutes. The generated prediction curve has a fit of 83.63% respect the original one.

Finally, Figure 7 shows a simulation of the root component of the system. It shows the four hemodynamic variables data corresponding to a real episode, stored previously, and the prediction generated by the system. In this way, it can be seen how the *oAlarm* signal is raised when the prediction provided by the corresponding predictors set exceeds a threshold (32 in this example). Consequently, the alarm signal is set low when the value goes back below that threshold.

5 CONCLUSION

This paper describes a VHDL implementation of a robust migraine predictive system. It has been developed directly from a DEVS complex model. It is able to detect errors in the input signals and regenerate them through ARX models. Migraine crises are predicted using sets of state-space models. That sets are prepared to work with different input variables and are trained taken in account several prediction horizons. Each predictive model has been selected to mitigate the effect of failures in sensors, maintaining the accuracy level even in case of such errors. The predictive system has been implemented using a previous DEVS model as a reference. As a result, the validation at the output of each module of the hardware specification has been a straightforward process.

Next steps include the implementation of more complex predictive algorithms to get more accurate models. We will also test the system in a development board, introducing data from sensors in real time instead of using previously recorded data. We will also analyze the reduction in the precision with the use of fixed point. Finally, the research group will work on automating the process of generating the VHDL code directly from the DEVS specification, for example using the DEVS Modeling Language (Mittal and Risco-Martín 2017).

6 ACKNOWLEDGEMENTS

This research has been funded by the EU (FEDER) under research grant PI15/01976 and the Spanish Ministry of Economy and Competitiveness under research grant TIN 2015-65277-R.

REFERENCES

Barshan, B., and M. C. Yüksek. 2014. “Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units”. *The Computer Journal* vol. 57 (11), pp. 1649–1667.

Giffin, N., L. Ruggiero, R. Lipton, S. Silberstein, J. Tvedskov, J. Olesen, J. Altman, P. J. Goadsby, and A. Macrae. 2003. “Premonitory symptoms in migraine An electronic diary study”. *Neurology* vol. 60 (6), pp. 935–940.

Heldberg, B. E., T. Kautz, H. Leutheuser, R. Hopfengartner, B. S. Kasper, and B. M. Eskofier. 2015. “Using wearable sensors for semiology-independent seizure detection-towards ambulatory monitoring of epilepsy”. In *EMBC, 2015*, pp. 5593–5596. IEEE.

Linde, M., A. Gustavsson, L. Stovner, T. Steiner, J. Barré, Z. Katsarava, J. Lainez, C. Lampl, M. Lantéri-Minet, D. Rastenyte et al. 2012. "The cost of headache disorders in Europe: the Eurolight project". *European journal of neurology* vol. 19 (5), pp. 703–711.

Lipton, R. B., and A. I. Scher. 2001. "Epidemiology and economic impact of migraine". *Current Medical Research and Opinion®* vol. 17 (S1), pp. s4–12.

Mittal, S., and J. L. Risco-Martín. 2017. "DEVSML 3.0: Incorporating Docker and Microservices for Rapid Deployment of DEVS Farm in Cloud Environment". In *Proceedings of the 2017 Spring Simulation Multionference (SpringSim 2017)*.

Pagán, J., J. M. Moya, J. L. Risco-Martín, and J. L. Ayala. 2017. "Advanced Migraine Prediction Simulation System". In *Summer Computer Simulation Conference (SCSC)*.

Pagán, J., D. Orbe, M. Irene, A. Gago, M. Sobrado, J. L. Risco-Martín, J. V. Mora, J. M. Moya, and J. L. Ayala. 2015. "Robust and accurate modeling approaches for migraine per-patient prediction from ambulatory data". *Sensors* vol. 15 (7), pp. 15419–15442.

Pagán, J., J. L. Risco-Martín, J. M. Moya, and J. L. Ayala. 2016. "A Real-Time Framework for a DEVS-based Migraine Prediction Simulator System". In *MAEB 2016*.

Pagán, J., , J. L. Risco-Martín, J. M. Moya, and J. L. Ayala. 2016. "Grammatical Evolutionary Techniques for Prompt Migraine Prediction". In *Genetic and Evolutionary Computation Conference, 2016*. ACM.

Pagán, J., M. Zapater, and J. L. Ayala. 2017. "Power transmission and workload balancing policies in eHealth mobile cloud computing scenarios". *Future Generation Computer Systems*, pp. –.

Stovner, L. J., and C. Andree. 2010. "Prevalence of headache in Europe: a review for the Eurolight project". *The journal of headache and pain* vol. 11 (4), pp. 289–299.

Van Overschee, P., and B. De Moor. 1994. "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems". *Automatica* vol. 30 (1), pp. 75–93.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2 ed. Academic Press.

Zheng, Y.-L., X.-R. Ding, C. C. Y. Poon, B. P. L. Lo, H. Zhang, X.-L. Zhou, G.-Z. Yang, N. Zhao, and Y.-T. Zhang. 2014. "Unobtrusive sensing and wearable devices for health informatics". *Biomedical Engineering, IEEE Transactions on* vol. 61 (5), pp. 1538–1554.

AUTHOR BIOGRAPHIES

KEVIN HENARES is a Ph.D. candidate at the Complutense University of Madrid (UCM). His email address is khenares@ucm.es.

JOSUÉ PAGÁN is a research assistant at the Technical University of Madrid (UPM). His work focuses on the development of robust methodologies for information acquisition in biophysical and critical scenarios. His email address is j.pagan@upm.es.

JOSÉ L. AYALA is currently an Associate Professor in the Department of Computer Architecture and Automation at the Complutense University of Madrid. His research interests focus on thermal and energy aware design and management of processor-based systems, design of embedded processors, thermal estimation, health monitoring and wireless sensor networks. His email address is jayala@ucm.es.

JOSÉ L. RISCO-MARTÍN received his Ph.D. from Complutense University of Madrid, and currently is Associate Professor in the Department of Computer Architecture and Automation at Complutense University of Madrid. His research interests include computer aided design, optimization and discrete event simulation. His email address is jlrisco@ucm.es.