System-on-Chip Design of the Cortical-Diencephalic Centre of the Lower Urinary Tract

Francisco Maciá Pérez
(University of Alicante, Alicante, Spain
pmacia@ua.es)

Leandro Zambrano Méndez
(Jose Antonio Echeverria Higher Polytechnic Institute, Havana, Cuba
celzambranom@gmail.com)

José Vicente Berná Martínez
(University of Alicante, Alicante, Spain
jvberna@ua.es)

Roberto Sepúlveda Lima
(Jose Antonio Echeverria Higher Polytechnic Institute, Havana, Cuba
sepul@ceis.cujae.edu.cu)

Abstract

This article presents the design of a field programmable gate array (FPGA)-based prototype of a system on chip (SoC) capable of behaving as one of the nerve centres comprising the neuroregulatory system in humans: the cortical-diencephalic nerve centre. The neuroregulatory system is a complex nerve system consisting of a heterogeneous group of nerve centres. These centres are distributed throughout the length of the spinal cord, are autonomous, communicate via interneurons, and govern and regulate the behaviour of multiple organs and systems in the human body. As a result of years of study of the functioning and composition of the neuroregulatory system of the lower urinary tract (LUT), the centres that regulate this system have been isolated. The objective of this study is to understand the individual functioning of each centre in order to create a general model of the neuroregulatory system that is capable of operating at the level of the actual nerve centre. This model represents an advancement of the current black box models that do not allow for isolated or independent treatment of system dysfunction. In this study, we re-visit our research into the viability of the hardware design of one of these centres—the cortical-diencephalic centre. We describe this hardware because functioning of the centre is completely configurable and programmable, which validates the design for other centres that comprise the neuroregulatory system. In this document, we succinctly present the formal model of the centre, propose a hardware design and an FPGA-based prototype, construct a testing and simulation environment to evaluate it and, lastly, analyse and contrast the results using data obtained from real patients, verifying that the functional behaviour fits that observed in humans.

Keywords

Cortical-Diencephalic Centre, Neuroregulatory System, FPGA, System on Chip

1. Introduction

Researchers continually seek to resolve complex health problems with innovative methods. One strategy consists of combining technological and biological systems to resolve, monitor, correct, or modulate organ or bodily subsystems that, in one way or another, do not function as they should.

Accordingly, the creation of embedded hardware devices that can be implanted into the body to correct its dysfunctions [1, 2] is already achieving results [3, 4, 5].

The neuroregulatory system is one of the most sensitive and important elements of the human body. The system is extremely complex and, consequently, its proper functioning is difficult to study without causing it harm. Consequently, several studies that have addressed this subsystem and its malfunctions, such as (i) in [4], where the author describes the design and implementation of reconfigurable hardware with an architecture capable of emulating neural networks in real time for correcting potential disorders of the nervous system, and (ii) [3], which proposes the development of an emulator of the visual system to reproduce retinal and visual cortex neuron activity. In [5], the authors describe a proposal to implement hardware to improve speech in individuals with hearing impairments. Studies such as these explore the use of artificial systems for resolving problems that involve the neuroregulatory system. All of these reports demonstrate the difficulty in working with and understanding the nervous system and reveal that the root of multiple deficiencies in the body often lies in the nervous system, not in the malfunctioning organ.

After numerous years of study, we have now available a validated and confirmed theoretical model of the neuroregulatory system for the lower urinary tract (LUT) in both normal and abnormal conditions [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. It was confirmed that, in all cases, and despite the fact that abnormal conditions were originally not taken into account in the model, the system behaves in a manner similar to actual biological systems [7]. The LUT was chosen due to social interest and its influence in the treatment of incontinence. Moreover, this system is sufficiently complex for rigorous validation of the model but simple enough to be able to be able to be modelled properly. The model was conceived and expressed using agent theory as a Multi-Agent System (MAS), being composed of agents capable of emulating the behaviour of different nerve centres that comprise it, and following a perceive-deliberate-execute paradigm (PDE agents) [8]. In the proposed model, each agent models a nerve centre, and communication between the agents models neuronal connections. The resulting model facilitates modular development, as it is already composed of independent, self-contained elements. This characteristic allows for the incorporation of system components without needing to considerably alter the remaining entities.

Until now, the primary achieved goals can be summarised as lower urinary tract monitoring and simulation, allowing physicians to identify the dysfunctions in their patients and allowing patients to train themselves through feedback to recuperate or substitute elements of their lost functionality. The next steps are to develop hardware designs based on a proposed architecture that implements the functionality of the neuroregulatory system. The hardware design can be converted into the bases of an embedded system on chip (SoC), which implements the functionality of neuroregulatory centres. In the present study, we focus on the hardware design of a specific centre, the cortical-diencephalic (CD) nerve centre. This centre forms part of the LUT, consisting of a known model for which sufficient information is available for appropriate validation using previously obtained patient data from different clinical techniques such as electromyography and pressure tests. [6, 7, 8]. At the same time, the proposed design was conceived so that, in the near future, it could behave as any other nerve centre via simple modulation of its functional parameters.

With these objectives in mind, the remainder of this study will be structured as follows: section 2 provides a technical overview of the studies with major relevance to our current project; section 3 shows the proposed solution alongside the theoretical basis that precedes it, along with details of the prototype; section 4 presents the results of the tests and validations by comparing the proposed prototype with clinical data; and lastly, section 5 summarises the primary conclusions drawn from the study and provides suggestions for future work.

2. Technical Overview

One of the most important challenges in medicine is improving the quality of life of patients struggling with some form of pathology caused by malfunctions in organs or other parts of the body that cannot be efficiently treated with traditional medicine. The synergy between medicine and technology [2] has been vital in resolving, monitoring, or correcting organ dysfunctions or damaged body systems. One potential solution consists of the creation of hardware that can be implanted in humans and compensate for these malfunctions. One objective for hardware used in the development of bioinspired systems is the creation of robust devices that can reliably replace organ functions [19, 20, 21, 22].

The first step in constructing hardware systems is a complete understanding of the system to be developed. Consequently, numerous proposals emulate modelled biological systems [4], observe its behaviour [24] and, subsequently, aim to synthetically reproduce the behaviour of the biological system [23, 37]. Once the system is understood, proposals of varying degrees of complexity can be developed, such as in [25], where the authors propose the design of a cortical neuroprosthesis capable of producing stimulating currents. In [26], the authors utilise neural networks implemented in reconfigurable hardware (FPGA – Field Programmable Gate Array) for stage segmentation, and in [27], the visual system is modelled to reproduce neuronal activity. The work in [38] implements a hardware system tolerant of errors from the study of prokaryotic bacteria morphology and behaviour, and a cellular machine is created in [45] by modelling the behaviour of the physarum polycephalum mould.

In creating a prototype, FPGA is useful because of its versatility and capacity to synthesise hardware. A wide variety of studies have made use of this technology for different projects, such as (i) a description of tools for the automatic design of visual system models [28]; (ii) a proposal of a hardware platform to study and experiment on of different processing schemes for visual information from artificial retinas [29]; (iii) the design of bioinspired circuits for real-time vision [30, 45]; or (iv) investigating the resolution of auditory dysfunctions [31, 32].

The use of FPGA to implement theoretical or mathematical models has been well accepted by the scientific community. Different studies have implemented FPGA for a number of different techniques, such as the following: (i) a highly efficient architecture to eliminate the negative effects of glasses [33]; (ii) the reconfigurable hardware implementation of an algorithm for facial detection [34]; (iii) modelling portions of the cerebral cortex to solve partitioning problems in processing sensory information [39]; (iv) the proposal of a processing unit by way of a neuronal model with optimised architecture [40]; and (v) a number of studies that implement algorithms to describe the functioning of neuronal networks [35, 36].

These studies illustrate that FPGA is highly flexible in creating prototypes, assisting with designing tasks; enabling the exploration of alternatives to the original design; and providing support by way of parallelisation, processing speed, and the potential for high-density components, among other properties.

The model of the neuroregulatory system that is the focus of this study has been validated and confirmed by studying the LUT. The study of urinary dysfunctions is a complex problem that requires understanding not only the functional organs but also those involved in urinary function regulation[42] as well as the neuronal connections involved in urination. Investigating the neuroregulatory side of this issue, [43] describes cases of incontinence caused by problems with nerve centres and explains how afferent signals act on these centres. These studies focus on the analysis and understanding of the LUT neuroregulatory system and its link with urinary incontinence, clearly demonstrating that the detection of possible dysfunctions, as well as correcting them, are an open problem. Consequently, studies such as [44] have described the development of an embedded system using the self-organisation of artificial neural networks to aid in the diagnosis of urinal dysfunction. From these studies, it can be concluded that projects focused on the LUT neuroregulatory system, few though there are, provide a better understanding of the functioning of this complex system. At the same time, problems arise from improper functioning of the neuroregulatory system that regulates the LUT. It is at this level that our study begins, i.e., in the appropriate neuroregulatory system and resolving a dysfunction that can be localised to a particular neuronal system.

3. History

Our group has been working on modelling and simulating the neuroregulatory system for over a decade, with the goal of contributing to its diagnosis [9], control, and possible correction of dysfunctions [6, 10, 11]. These studies provide the formal framework on which we construct our current design. In this section, we describe the most prominent aspects of this design.

According to the model shown in [6], a neuroregulatory biological system (NBS) is formed from a mechanical system (MS), a neuroregulatory system (NRS), neuronal connections (NC), and a system domain (SD). Formally, it can be defined as follows:

$$\mathrm{NBS} = \langle\, \mathrm{MS}, \mathrm{NRS}, \mathrm{NC}, \mathrm{SD} \,\rangle \tag{1}$$

From this point forward, we discuss only analyses of the neuroregulatory systems (NRS), its neuronal connections (NC), and the system domain (SD), as these are the elements involved in our current study.

Let A and B be ensembles that represent two types of system components. A neuronal connection between an element *x* from A to an element *y* in B (which is not necessarily different from A) is represented as (x, y) / $x \in \mathrm{A}$, $y \in \mathrm{B}$. Element *x* represents the start of the connection, and element *y* the termination.

The neuroregulatory system (NRS) represents the ensemble of nerve centres as entities capable of acting autonomously on the mechanical system based on information on its state and its relation with other involved nerve centres:

$$\mathrm{NRS} = \{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\} \tag{2}$$

The ensemble of neural connections is composed of afferent neuronal connections (ANC), efferent neuronal connections (ENC), and internal neuronal connections (INC). Each neuronal connection is formally defined as follows:

$$\mathrm{NC} = \mathrm{ANC} \cup \mathrm{ENC} \cup \mathrm{INC} \tag{3}$$

Let $x \in$ NRS and $y \in$ MS, then the relationships for ANC, ENC, and INC, which represent different types of neuronal connections, are defined as follows:

$$\mathrm{ANC} \subseteq \mathrm{MS} \times \mathrm{NRS} \tag{4}$$

$$\mathrm{ENC} \subseteq \mathrm{NRS} \times \mathrm{MS} \tag{5}$$

$$\mathrm{INC} \subseteq \mathrm{NRS} \times \mathrm{NRS} \tag{6}$$

The notations $(ms, \alpha t) \in$ ANC, $(\alpha s, ms) \in$ ENC, and $(\alpha s, \alpha t) \in$ INC will be simplified as ${}^{ms}A_{\alpha t}$, ${}^{\alpha s}E_{ms}$ and ${}^{\alpha s}I_{\alpha t}$, respectively. The following notations are used: αs: starting nerve centre; αt: terminal nerve centre; and *ms*: component of the mechanical system.

Taking into account that nerve centres act on the system as autonomous agents, the system domain would consider the LUT as a system of actions and reactions with the following structure:

$$\mathrm{SD} = < \Sigma, \Gamma, \mathrm{P}, React > \tag{7}$$

where:

> Σ represents the ensemble of possible system states. It is formally represented in the following way: $\Sigma = \{\sigma_i / \sigma_i = (s_i, v)$ where $s_i \in$ NC and $v \in \mathbb{R}\}$.
>
> Γ represents the ensemble of possible policy decisions on the system. As these centres do not possess total control over the system and must balance their objectives with those of other centres, the result of each action is represented as a change decision. It is defined as $\Gamma = \{\gamma_i / \gamma_i = (s_i, v)$ where $s_i \in$ NC and $v \in \mathbb{R}\}$.
>
> P represents the ensemble of possible actions that the different nerve centres could perform on the system to modify its state. It is represented as follows: $\mathrm{P} = \{\rho_i / \rho_i = (label, pre, post)$ where *label* is an expression of the form $f(x_1, x_2, \ldots, x_n)$ and *pre* and *post* are groups of formulas of the form $g(a_1, a_2, \ldots, a_n)\}$. For example: ρ_i = (*Setsignalvalue*(), *True*(), *Value*()).
>
> *React*: a function that models the behaviour of the mechanical system under the different $\gamma_i \in \Gamma$.

Having taken into account a well-validated theoretical model of the neuroregulatory system through study of the LUT, in the following paragraphs, we illustrate how to set the model definitions using this system.

The neuroregulatory system is linked to the anatomy and physiology of the neuronal control of the urinary tract, transmission centres, and areas responsible for facilitating and inhibiting the process of urination [4, 5, 6, 7]. This system also describes the behaviour of neuronal connections and control centres of the involved nerves [6, 7, 12, 13, 14]. Nerve centres are groups of neurons with the same function, receiving signals generated by the mechanical system and from other centres, processing said signals and retransmitting them to other centres or to the mechanical system [6, 7, 8, 15]. Interaction with the mechanical system is achieved through an ensemble of afferent and efferent neuronal signals. The neuronal pathways carry the information from the mechanical system towards the control centres, which then process the information and transmit it to the mechanical system, resulting in activity in different parts of the system. Figure 1 shows a basic diagram of the model of the LUT neuroregulatory system [6, 16]. It consists of nine neuronal centres; nine afferent signals (A) coming from the mechanical system; two internal signals (I) generated voluntarily from the facilitatory areas responsible for retention and urination; other internal system signals (I) that transmit impulses from one centre to another; and eight efferent signals (E) that act on the muscles of the mechanical system. Each signal is designated with A, E, or I to identify its type, along with a superscript (on the left) and a subscript (on the right) that indicate the start and terminal, respectively, of the signal. For example, the signal designated ${}^{PM}I_{DGC}$ is an internal signal that starts at the PM (Pontine Micturition) centre and terminates at the DGC (Dorsal Grey Commissure).

Although external to the LUT, the voluntary signals ${}^{M}I_{CD}$ and ${}^{R}I_{CD}$ are designated internal (I) given that they arrive from nerve centres in other neuroregulatory subsystems. These two signals are generated from voluntary areas at the conscious level and depend on internal factors, such as the desire to urinate or not, as well as external factors, such as finding an appropriate location to urinate or encountering flowing water. The ${}^{R}I_{CD}$ signal would activate the moment the individual senses that their bladder is full but wants to continue to retain their urine, and the ${}^{M}I_{CD}$ signal would be active over a short period of time and trigger the act of urination.

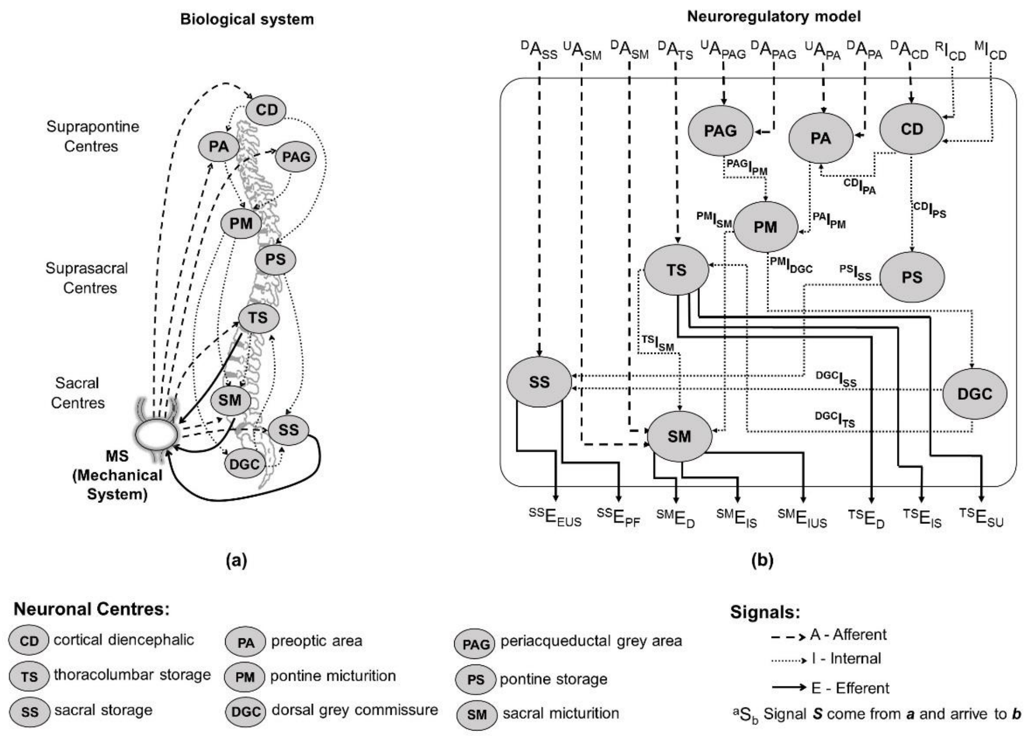

Figure 1. Simple diagram of the model of the neuroregulatory system.

The nerve centres that comprise the neuroregulatory system have been modelled as groups of neurons that process signals and transmit responses. Because this behaviour is well suited to a PDE agent (*Perception*, *Deliberation*, *Execution*) [17], the nerve centres have been modelled on an architecture based on a PDE agent, giving it the ability to retain memory and thus a richer and more potent decision-making process [18]. Given these properties, a nerve centre $\alpha \in \mathrm{NS}$ is defined as follows: $\alpha = (\Phi_\alpha, S_\alpha, \mathit{Percept}_\alpha, \mathit{Mem}_\alpha, \mathit{Decision}_\alpha, \mathit{Exec}_\alpha)$ [6], where Φ_α is the group of perceptions; S_α is the group of internal states; $\mathit{Percept}_\alpha$ provides information to the nerve centre about the system state; Mem_α gives the centre the capacity to be aware of its own state; $\mathit{Decision}_\alpha$ choses the next action to perform; and Exec_α represents the intention of the nerve centre to act on the entire system.

In a nerve centre, perception represents the ability to distinguish and classify system states not only with respect to the most significant characteristics but also to those actions under its purview.

Perception can formally be considered as a function linking one group of values termed "perceptions" with a group of system states [6]:

$$Percept_{\alpha} : \Sigma \rightarrow \Phi_{\alpha} \tag{8}$$

The group of possible perceptions associated with a specific nerve centre is defined in the following way: $\Phi_{\alpha} = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$, where each φ_i is a structure composed of a list of paired objects, an element and its value, corresponding to the system state previously defined [6].

Each nerve centre has an internal state with memory, which allows for more complex behaviours. The group of internal states of a specific centre is defined in the following way: $S_{\alpha} = \{s_1, s_2, \ldots, s_m\}$ [6].

The function of decision making is charged with relating an action with perception in a particular internal state of the nerve centre and is defined as [6]:

$$Decision_{\alpha:}\ \Phi_{\alpha} \times S_{\alpha} \rightarrow P \tag{9}$$

Memorisation links an internal state of the centre with its current perception of the environment and its past behaviour [6]:

$$Mem_{\alpha:}\ \Phi_{\alpha} \times S_{\alpha} \rightarrow S_{\alpha} \tag{10}$$

Once the centre has determined which action to take, it should execute this action using the execution function defined as follows [6]:

$$Exec_{\alpha:}\ P \times \Phi_{\alpha} \rightarrow \Gamma \tag{11}$$

where P is the group of actions that can be performed on the system, Φ_{α} is the group of perceptions that centre α can have about the system, and Γ is the group of inputs from different nerve centres.

Figure 1 (a) also shows a simple representation of the mechanical system (MS) of the LUT that the model seeks to regulate. This system is modelled via a function, *MS*(), which obtains functionality as a group of PDE agents interacting in a Multi-Agent System (MAS).

$$\mathrm{MS}: \Sigma \times \Gamma \rightarrow \Sigma \tag{12}$$

MS() is responsible for generating afferent signals ($\sigma(t+1)$) based on a group of efferent signals (γ(t)). This function coincides with the dynamics of the mechanical system and is carried out due to actions that transform the system from one state to another [4]. This dynamic represents the system reaction to different inputs.

$$\begin{cases} \sigma(t+1) = MS\left(\sigma(t), \prod_{i=1}^{n} Exec_i\left(Decision_i(\phi_i(t), s_i(t)), \phi_i(t)\right)\right) \\ s_1(t+1) = Mem_1\left(\phi_1(t), s_1(t)\right) \\ \quad \vdots \\ s_n(t+1) = Mem_n\left(\phi_n(t), s_n(t)\right) \\ \qquad with\ \phi_i(t) = Percept_i(\sigma(t)) \end{cases} \tag{13}$$

The general system dynamics are thus defined with a group of *card*(NRS)+1 equations, where the first determines the global system state at a time *t* and as a function of the behaviour of each centre, and the remaining equations correspond to the internal state of the different nerve centres at the same time *t*.

4. Proposal

With these baselines established, our proposed model can be based on a system on chip (SoC) that behaves like the model of the neuroregulatory system of the LUT previously described, which we term **Neuronal SoC**. In the previous work [48] a first prototype of the LUT model with strong architectural restrictions was proposed, although with good hardware performance. However, the need to create a SoC that is valid for the LUT and for any other neuroregulatory subsystem makes us evolve to another hardware approach, more flexible and dynamic, completely different and with greater capabilities. Figure 2 demonstrates the general scheme of the design of the SoC, whose primary features will be analysed below.

The chip is responsible for receiving a group of input signals corresponding to afferent nerve signals (A), which result in the running of one of the processes and for generating a group of output signals that correspond to efferent nerve signals (E). This processing is achieved through nine **CN** blocks incorporated into the chip, corresponding to the nine nerve centres that form part of the LUT neuroregulatory system. Two additional blocks (***INPUT*** and ***OUTPUT***) are responsible for manipulating the input and output signals. The chip also includes a compartmentalised ***SHARED MEMORY***, which stores the afferent, internal, and efferent signals responsible for the functioning of the neuroregulatory system.

To process the input signals, each **CN** block (Figure 3) is supported by **DECISION** blocks (Figure 4), which process groups of instructions following a micro-program stored in a specially designed program memory contained within each nerve centre. This micro-program consists of a group of sequentially executed instructions that complete a specific task. In this case, a program can be created that behaves like any of the centres that make up the neuroregulatory system. Because of the need to supply the program memory with the executing micro-program and to store initial values into the database, there is a Mode signal (*M*) responsible for indicating if the chip is in the configuration or working mode. Three **DECISION** blocks are used in our model for centre deliberation, as each centre needs to be able to perform similar groups of operations. These **DECISION** blocks contain all of the possible operations that can be completed using the input values. The Decision block is the one that requires the most hardware and produces the highest consumption, to optimize the SoC, we decided to use a Decision block for every 3 CN blocks. After the execution flow study, we have observed that the operations performed by the hardware do not require 9 parallel decision centres working at the same time. With this decision, up to 20% savings in components are produced by synthesizing the chip, maintaining the restrictions of computational parallelism. Using these three blocks guarantees maximum parallelisation among the nerve centres for parallel functioning of the neuroregulatory system. Additionally, delivery of the signal values between different blocks would be through three communication buses: a control bus, an address bus, and a data bus. Lastly, this design also incorporates a clock signal (*Clk*) to synchronise system activity.

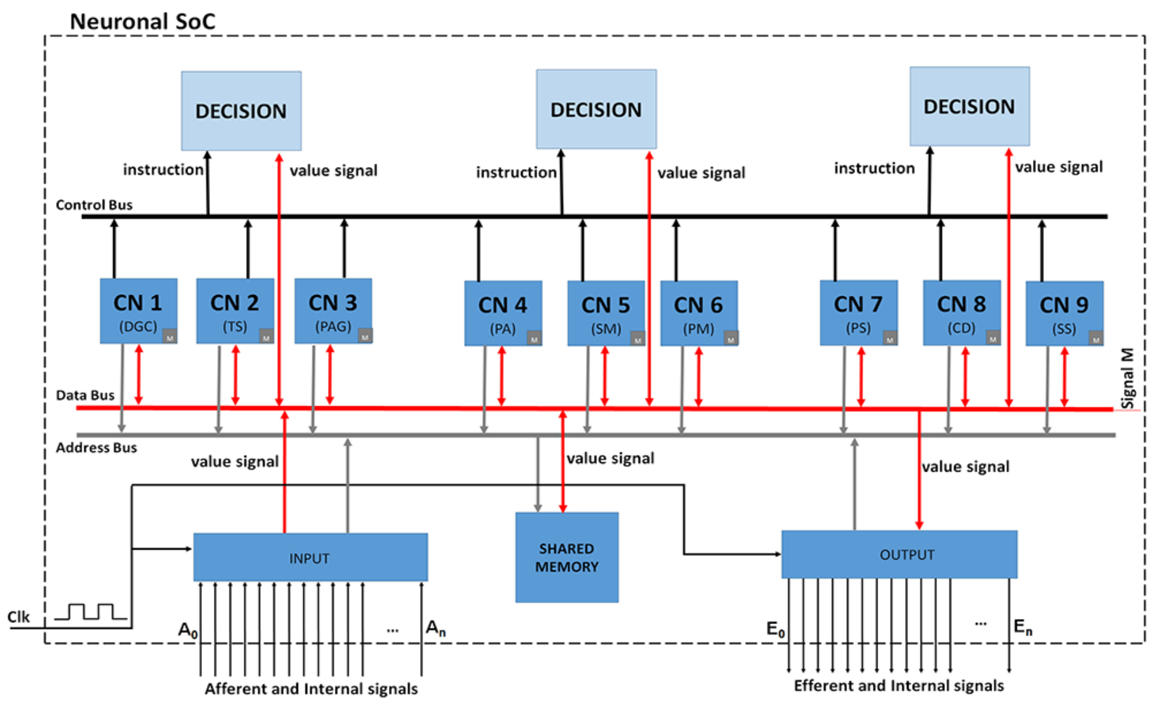

Figure 2. General schematic of the hardware design of Neuronal SoC.

With the general schematic of the system laid out, it is now necessary to go in depth into the internal designs of the nerve centres. Figure 3 demonstrates the schematic of the internal design of one of the nerve centres (**CN** block) and its functional elements.

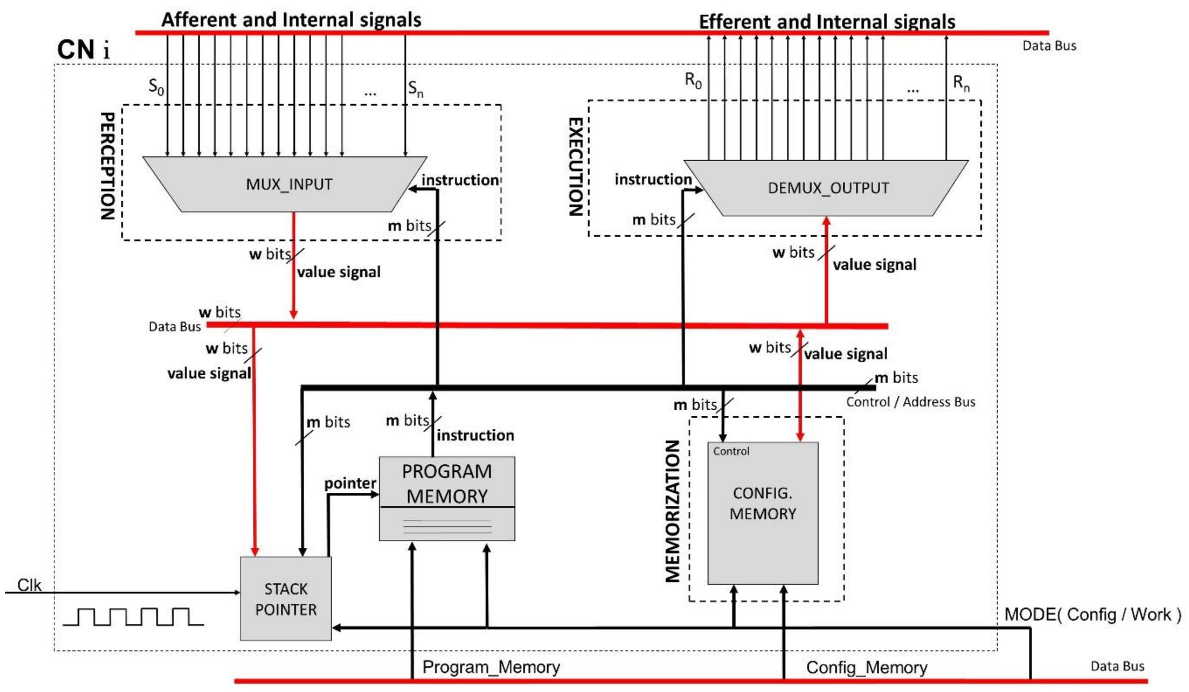

Figure 3. Schematic of the internal design of a nerve centre.

The internal design prototype contains a **PERCEPTION** block with the elements necessary for selecting the signals that form part of the process by executing the micro-program; a **MEMORISATION** block that stores the internal state of the centre and links it to the perception signals, as well as stores other important values needed for the necessary operations; and an **EXECUTION** block that selects the output signals affected by the decision taken.

One of the most important blocks in **Neuronal SoC** is the **DECISION** block that, unlike the LUT MAS model described in section 2, lies outside each nerve centre; thus, (as seen in Figure 2) there can be three generic decision blocks in the system. These blocks would be shared by other centres, contributing to their function. In our proposed model, three **DECISION** blocks are defined to achieve maximum parallelisation in the functioning of the nine nerve centres. Each **DECISION** block would be responsible for performing the decision function (see equation 9) that forms part of the functioning of the nerve centres defined in the theoretical model. For the hardware prototype to perform the decision function, a decision block is required to relate an action with a perception for a particular internal state (see equation 9). Using the signals obtained by the perception block and the internal state stored in the memory of the decision block, a group of operations is executed to take the decision and send the result to the **EXECUTION** blocks. To execute this group of operations, the **DECISION** block was designed as an Arithmetic Logical Unit (**ALU**), which contains all of the possible operations that can be executed using the input values. The internal design of this ALU is shown in Figure 4.

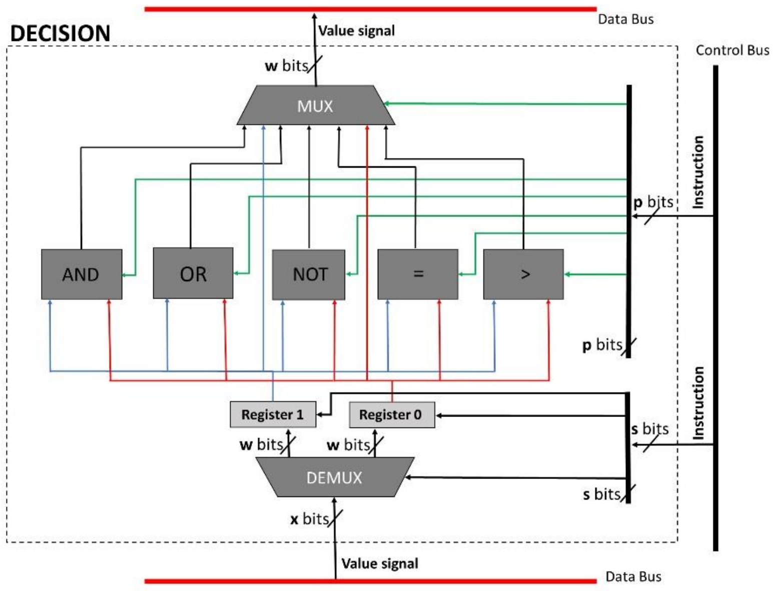

Figure 4. Design of the DECISION block.

The decision block requires the perception signals, threshold values, and the internal state of the centre to perform its operations. The block obtains a signal of ***x*** bits that is sent to a demultiplexer that is responsible for sending the signal to one of the two input registers. Control signal ***s*** also enters the demultiplexer and indicates which of the two registers should store the value of said signal. This control signal forms part of the instructional register that arrives at the decision block from the program memory.

As the design of the decision block was based on the ALU, this block requires a group of modules responsible for performing AND, OR, NOT, greater than (>), and equals (=) operations. The "greater than" and "equal" operations are the only kinds of numerical comparisons that need to be directly implemented, as any others can be easily achieved with logical modifications; for example, "less than" is the equivalent of a NOT performed on the "greater than" operation. To select the required action at the exact moment needed, it was necessary to use the first *p* bits of the control signal arriving from the program memory; the first three bits would indicate the operation required, two more would indicate which decision block should perform the operation, and the remaining bits would indicate if the operation is using the values stored in both or only one of the ALU registers. These *p* bits, along with the *s* control bits going into the demultiplexer, comprise the *m* bits of the instructions stored in the program memory. Once the corresponding action has been completed, it is necessary to indicate what the output result of the operation will be. The block thus also contains a multiplexer responsible for selecting the appropriate output, using the same *p* bits as control as used by the ALU for controlling its operations. We next describe each of the functional blocks that comprise the proposed model.

The module designed for bringing the functions (see equation 8) to the hardware has *n* inputs connected to different signals. From here, signals involved in the process of the functioning of the nerve centre are selected. One micro-program is responsible for selecting these input signals; therefore, the part of the chip that carries out the functionality of perception possesses the required elements. In this way, using the appropriate micro-program, the signals that form part of the execution can be selected. A block was constructed to achieve this functionality, receiving *n* different signals and only passing those necessary for proper function. This block requires a control signal of *m* bits from the program memory that would be responsible for indicating which of the input signals would be used by the centre.

Because of the requirement of the *n* input signals, a multiplexer was designed with *n*, *x*-bit inputs ($S_0 - S_n$) and an output of *w* bits, ensuring that the size of the data to be managed is defined according to the needs of each design. This multiplexer is responsible for passing only the input value indicated by the *m*-bit control signal, which uses a mask to tell the multiplexer which of the input signals to pass. Needing to rely on *n* inputs makes the design more generic.

The model of the neuroregulatory system suggests that each centre is capable of performing deliberation using two fundamental processes, one being memorisation (see equation 10). The hardware prototype thus contains a memorisation block responsible for storing the internal state of the nerve centre at each moment in order to obtain a more complex development in each centre. This ability allows the system to check on the previous state and to assist in correcting potential incorrect functioning. This memorisation block has also been used in our proposed model to store threshold values that will be used in the decision block, along with possible values of the output signals of the centre in question.

The memorisation block needs to be configured by storing threshold values and potential relevant output signals. This block includes a one-bit *MODE* signal that indicates two possible states for data memory: the *Config* state, in which the memory stores the necessary values for the execution of the centre coming from the *Config Data_Memory* signal, and the *Work* state, which indicates that the memory is working and being used by the other blocks from the design. The memorisation block also receives a *Control* signal of *m* bits from the program memory; the first three bits indicate whether to read or write memory at any one instant and the address where the memory should be read or written. Because the values of the input and output signals are of *x* bits, the memory block is correspondingly composed of *x*-bit words. It is important to note that the memory block can write a value in one address not only based on what is contained in the three bits of the *Control* signal but also when the address appears as a destination block within the bits of the *Control* signal.

Another block found within the internal design of the nerve centre performs executions (see equation 11). The hardware prototype for this function contains an execution block that, similar to our model chip, has *n* outputs connected to different signals, from which are selected those resulting from the execution processes. The micro-program is responsible for selecting these output signals; therefore, the part of the chip that carries out executions possesses the necessary elements that can select the output signals using this micro-program and thus influence the different system centres. A block was created for this task that receives an *x*-bit signal that can then be transmitted to *n* different output signals. This block is also influenced by an *m*-bit control signal that arrives from the program memory and is responsible for selecting which of the output signals would be assigned with the salient signal value.

Figure 3 shows a generic hardware prototype that could function as any of the centres that comprise the neuroregulatory system. This figure demonstrates that the design could handle *n* input signals and *n* output signals, emulating the execution of nerve centres with the same properties. The generic prototype also contains a program memory that can write a micro-program to execute the complete behaviour of any centre in a step-by-step manner. It is important to note that the design of each of the abovementioned blocks is contingent on their functioning independently of the program stored in the program memory.

This program memory is managed via a pointer that moves position with every clock cycle from the *Clk* signal. Additionally, at any moment before execution of the centre, the program memory should be configured with the micro-program that will be executed; therefore, a one-bit *MODE* signal is supplied that indicates two possible states: the *Config* state when the executing program is being written (using the *Config program_memory* signal), and the *Work* state when it is working and thus preventing the previously written program from being altered.

A signal from the decision block can be transmitted to command the program memory pointer to move to a specific location using *go to* jump statements.

To carry out the operations called in the functioning of the centres and bring them to the hardware, it was decided to separate each of them into back-to-back operations. This choice requires that the action that must be carried out, the origin block, and the destination block be indicated for each instruction stored in the program memory. *m*-bit words were used that contain a structure as shown in Figure 5. The first *a* bits indicate the operation to be carried out on the data from a starting component, the output of which is subsequently sent to a terminal component. The next *b* bits indicate on which starting component the previously indicated action will be carried out; the next *c* bits signal the address or register within the starting component where the value needed to carry out the required operation is located. Another *b* bits indicate the terminal component where the data will be sent, and the final *c* bits indicate the address or register within the terminal component where the data should be copied.

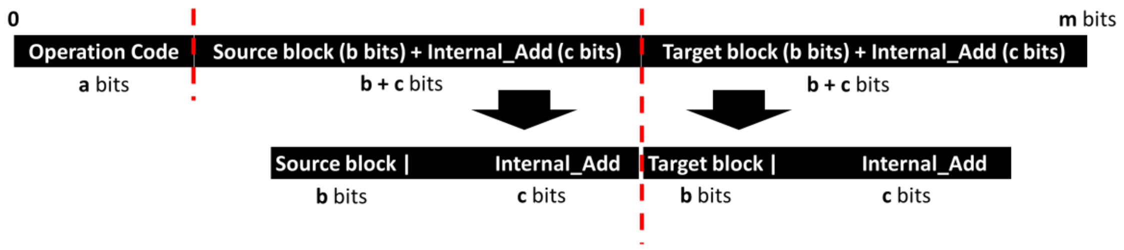

Figure 5. Structure of the instructions.

The proposed model above defines a group of operations that can be carried out on any one centre as well as the binary coding that indicates, at any moment, the operation that should be carried out. Table 1 shows these operations with their corresponding operational codes. The model also takes into account that each module present in the design could be both a starting and terminal component in managing the data. Therefore, each of the components was also codified to be aware, at every instant, from where and to where the signal values were being sent. Table 2 shows the coding scheme for each starting component, and Table 3 shows the scheme for the terminal components.

Table 1. Operations.

Operation	**Code**
NULL (Ø)	000
LOAD	001
AND	010
OR	011
NOT	100
=	101
>	110

Table 2. Staring components

Components (Start)	**Code**
INPUT	00
MEMORY	01
ALU	10

Table 3. Terminal components

Components (Terminus)	**Code**
OUTPUT	00
STACK P.	01
MEMORY	10
ALU	11

The first step in executing a centre is the configuration of the data memory and the program memory. The data memory will contain *w*-bit words; up to 2^{C} positions can be addressed to it, where *c* is the number of bits selected to indicate the addresses within the data memory. The memory will also contain the configuration values needed for execution of the centre, for which there will be reserved the first 2^{C-1} memory addresses. The remaining memory addresses will be reserved for work performed with the operation data.

The program memory will contain all of the instructions necessary to perform the execution of the centre. The memory pointer will move position by position, running each of the instructions through the control bus so that they can be received by the remaining system blocks.

5. Tests and Validation

5.1 Demonstrating the functioning of the CD centre on a prototype

The cortical-diencephalic (CD) centre is the highest-level nerve centre within the neuroregulatory system and the focus of our design when carrying out tests and validation. As part of the validation of our design, we created a testing procedure that involves the development of an example demonstrating the functioning of the CD centre. As described previously, each centre is composed of a structure of four functional blocks: perception, memorisation, decision, and execution. A greater understanding of the functioning of the CD centre first requires a presentation of the truth table (Table 4), which shows the signals and operations called in the execution of this centre.

Table 4. Truth table for the CD centre. ∀ Val ∈ R

Perception			**Memorisation**			**Decision**					**Execution**	
$^{D}\mathbf{A}_{CD}$	$^{M}\mathbf{I}_{CD}$	$^{R}\mathbf{I}_{CD}$	**previous state**			**current state**			$^{CD}\mathbf{I}_{PA}$	$^{CD}\mathbf{I}_{PS}$	$^{CD}\mathbf{I}_{PA}$	$^{CD}\mathbf{I}_{PS}$
			$^{D}\mathbf{A}_{CD-1}$	$^{M}\mathbf{I}_{CD-1}$	$^{R}\mathbf{I}_{CD-1}$	$^{D}\mathbf{A}_{CD}$	$^{M}\mathbf{I}_{CD}$	$^{R}\mathbf{I}_{CD}$				
$< {}^{D}H_{CD1}$	0	0	$< {}^{D}H_{CD1}$	0	0	Val	0	0	0	0	0	0
$< {}^{D}H_{CD2}$ $\geq {}^{D}H_{CD1}$	0	1	$< {}^{D}H_{CD2}$ $\geq {}^{D}H_{CD1}$	0	1	Val	0	1	0	1	0	1
$< {}^{D}H_{CD2}$ $\geq {}^{D}H_{CD1}$	1	0	$< {}^{D}H_{CD2}$ $\geq {}^{D}H_{CD1}$	1	0	Val	1	0	1	0	1	0
$\geq {}^{D}H_{CD2}$	x	x	$\geq {}^{D}H_{CD2}$	x	x	Val	Val	Val	1	0	1	0

The functioning of the CD centre consists of receiving a series of signals (${}^{D}A_{CD}$, ${}^{M}I_{CD}$, ${}^{R}I_{CD}$), followed by a performance of the comparisons laid out in the truth table. Each condition is verified sequentially and, when one condition is true, the corresponding output values are returned. The first condition verified is ($^{D}A_{CD} < {}^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) ^ (¬ $^{R}I_{CD}$) ^ (Previous State). If true, the returned output signal values are ($^{CD}I_{PA}$, 0) and ($^{CD}I_{PS}$, 0). If the condition is not true, the next one is verified: ($^{D}H_{CD1} \leq {}^{D}A_{CD} < {}^{D}H_{CD2}$) ^ (¬ $^{M}I_{CD}$) ^ ($^{R}I_{CD}$) ^ (Previous State). If true, the output values returned are ($^{CD}I_{PA}$, 0) and ($^{CD}I_{PS}$, 1). If this condition is also not true, the next condition would be compared: ($^{D}H_{CD1} \leq {}^{D}A_{CD} < {}^{D}H_{CD2}$) ^ ($^{M}I_{CD}$) ^ (¬$^{R}I_{CD}$) ∨ ($^{D}A_{CD} \geq {}^{D}H_{CD2}$) ^ (Previous State). If this condition if true, the output signals are ($^{CD}I_{PA}$, 1) and ($^{CD}I_{PS}$, 0). It is important to note that each condition should be verified so that an output signal is always produced.

These comparisons require threshold values and the previous stage of the centre to be carried out. Therefore, these values must be stored in the data memory along with the values of the output signals for each of the conditions, resulting in the memory configuration shown in Table 5. The threshold values $^{D}H_{CD1}$ and $^{D}H_{CD2}$ would be stored in the first two positions in the memory, all possible values of the output signals $^{CD}I_{PA}$ and $^{CD}I_{PS}$ in addresses 02 through 07, and lastly the value of the previous state of the centre in address 20. The addresses 3D, 3E, and 3F would be reserved to be used as accumulators where the results of the operations carried out would be stored, and addresses 3A, 3B, and 3C would store the addresses in the program memory where the pointer should jump to begin execution of the program depending on which condition from the truth table (Table 4) was found to be true.

Table 5. Configuration of the data memory for the CD centre.

Dir.(Hex)	**Value**	**Dir.(Hex)**	**Value**
00	DHCD1	20	1
01	DHCD2	21…	empty
02	0	..39	empty
03	0	3A	83
04	0	3B	88
05	1	3C	93
06	1	3D	accumulator
07	0	3E	accumulator
08…	empty	3F	accumulator
…1F	empty		

Following is the pseudocode that describes the functioning of the first condition to be compared: ($^{D}A_{CD} < {}^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) ^ (¬ $^{R}I_{CD}$) ^ (Previous State). This is presented to illustrate the general work flow used by the proposed hardware to interpret each of the conditions presented by the functioning of the CD centre:

```
Compare_condition ($^{D}A_{CD} < {}^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) ^ (¬ $^{R}I_{CD}$) ^ (Previous State)
Inputs: $^{D}A_{CD}$, $^{M}I_{CD}$, $^{R}I_{CD}$
Outputs: $^{CD}I_{PA}$, $^{CD}I_{PS}$
Nº    Pseudocode                                   Comment
1     Start
2     Perception_block← $^{D}A_{CD}$, $^{M}I_{CD}$, $^{R}I_{CD}$;   Assigning the inputs to the first three addresses of the
                                                   perception block.
3     Execution_block← $^{CD}I_{PA}$, $^{CD}I_{PS}$;            Assigning the outputs to the first addresses in the
                                                   execution block.
4     Separating the comparisons into bitwise
      operations (AND, OR, NOT, >, =);
5     Compare ($^{D}A_{CD} < {}^{D}H_{CD1}$)
  5.1 ALU_reg1← Perception_block[0];               Load signal $^{D}A_{CD}$ in register 1 of the ALU
  5.2 ALU_reg0← Memory[0x0];                       Load threshold $^{D}H_{CD1}$ in register 0 of the ALU
  5.3 Memory[0x3D]← Result of  ALU_reg0 >
      ALU_reg1;
6     Negate_signals ($^{M}I_{CD}$, $^{R}I_{CD}$)
  6.1 ALU_reg0← Perception_block[1];               Load signal $^{M}I_{CD}$ in register 0 of the ALU
  6.2 ALU_reg1← Perception_block[2];               Load signal $^{R}I_{CD}$ in register 1 of the ALU
  6.3 Memory[0x3E] ← Result of not ALU_reg0;
  6.4 Memory[0x3F] ← Result of not ALU_reg1;
```

```
7     Carry out logical AND between the result of Comparing ($^{D}A_{CD} < ^{D}H_{CD1}$) and the result of negating signal $^{M}I_{CD}$;
  7.1   ALU_reg0← Memory[0x3D];        Load the value stored in memory address 3D in register 0 of the ALU.
  7.2   ALU_reg1← Memory[0x3E];        Load the value stored in memory address 3D in register 1 of the ALU.
  7.3   Memory[0x3D]←Result of ALU_reg0 AND ALU_reg1;
8     Carry out logical AND between the result of operation ($^{D}A_{CD} < ^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) and the result of negating signal $^{R}I_{CD}$;
  8.1   ALU_reg0← Memory[0x3D];
  8.2   ALU_reg1← Memory[0x3F];
  8.3   Memory[0x3D]← Result of ALU_reg0 AND ALU_reg1;
9     Carry out logical AND between the result of operation ($^{D}A_{CD} < ^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) ^ (¬ $^{R}I_{CD}$) and the Previous State;
  9.1   ALU_reg0← Memory[0x3D];
  9.2   ALU_reg1← Memory[0x20];        The previous state of the centre is stored in memory address 20.
  9.3   Memory[0x3E]← Result of ALU_reg0 AND ALU_reg1;
10    Verify if the condition is true;
  10.1  ALU_reg0← Memory[0x3A];        Load the memory address within the program memory where the pointer should jump into register 0 of the ALU
  10.2  ALU_reg1← Memory[0x3E];        Lload into register 1 of the ALU the result of the ($^{D}A_{CD} < ^{D}H_{CD1}$) ^ (¬ $^{M}I_{CD}$) ^ (¬ $^{R}I_{CD}$) ^ (Previous State)
  10.3  IF result ALU_reg0 AND ALU_reg1 == 0;
  10.4  THEN Stack_P← 0;               Assign a 0 to component Stack_P, indicating that the program memory pointer should continue sequentially to the next comparison.
  10.5  ELSE Stack_P← address 83;      The program memory points should move to address 83 to continue executing the micro-program
  10.6  After jumping to program memory address 83 the current state is stored and the outputs are returned;
11    ALU_reg0← Memory[0x3D];
12    Memory[0x20]← ALU_reg0;          Store the new state of the centre in memory address 0x20
13    Execution_block [0]← Memory[0x02];   Value of the output signal $^{CD}I_{PA}$ from memory to the Execution Block.
14    Execution_block[1]← Memory[0x03];    Value of the output signal $^{CD}I_{PS}$ from to the Execution Block.
15    Stack_P← NULL operation (ϕ);     Tells the program memory to start micro-program execution from address 0x0
16    End
```

Table 6 shows the instructions stored in the program memory in binary format used to carry out execution of the first condition (*see Table* 4) for this centre.

Table 6. Instructions for executing the first condition of the CD centre.

Operation code	Source block	Internal _Add	Target block	Internal _Add	Description
001	00	000000	11	000001	LOAD INPUT REG 0 ALU REG 1
001	01	000000	11	000000	LOAD MEMORY ADD 0 ALU REG 0
110	10	000011	10	111101	> ALU REG 1 and 0 MEMORY ADD 3D
001	00	000001	11	000000	LOAD INPUT REG 1 ALU REG 0
001	00	000010	11	000001	LOAD INPUT REG 2 ALU REG 1
100	10	000000	10	111110	NOT ALU REG 0 MEMORY ADD 3E
100	10	000001	10	111111	NOT ALU REG 1 MEMORY ADD 3F
001	01	111101	11	000000	LOAD MEMORY ADD 3D ALU REG 0
001	01	111110	11	000001	LOAD MEMORY ADD 3E ALU REG 1
010	10	000011	10	111101	AND ALU REG 0 and 1 MEMORY ADD 3D
001	01	111101	11	000000	LOAD MEMORY ADD 3D ALU REG 0
001	01	111111	11	000001	LOAD MEMORY ADD 3F ALU REG 1

010	10	000011	10	111101	AND ALU REG 0 and 1 MEMORY ADD 3D
001	01	111101	11	000000	LOAD MEMORY ADD 3D ALU REG 0
001	01	100000	11	000001	LOAD MEMORY ADD 20 ALU REG 1
010	10	000011	10	111110	AND ALU REG 0 and 1 MEMORY ADD 3E
001	01	111010	11	000000	LOAD MEMORY ADD 3A ALU REG 0
001	01	111110	11	000001	LOAD MEMORY ADD 3E ALU REG 1
010	10	000011	01	000000	AND ALU REG 0 and 1 STACK P REG 0
001	01	111101	11	000000	LOAD MEMORY ADD 3D ALU REG 0
001	10	000000	10	100000	LOAD ALU REG 0 MEMORY ADD 20
001	01	000010	00	000000	LOAD MEMORY ADD 2 OUTPUT REG 0
001	01	000011	00	000001	LOAD MEMORY ADD 3 OUTPUT REG 1
000	10	000000	01	000000	Ø ALU REG 0 STACK P REG 0

The descriptions above illustrate how the execution of the CD centre checks whether the first of the conditions in Table 4 is true; the execution of the rest of the conditions occurs similarly. Appendix 1 contains the rest of the instructions as a complete configuration of the program memory used to execute the CD centre.

5.2 Creating the hardware prototype

Testing and validating the proposed hardware was carried out using FPGA in order to avoid needing to construct a physical circuit. Using this technology avoids possible errors that underlie the physical materialisation of the hardware, such as bad connections, a broken component, or sporadic errors from incorrect voltages, resulting in a reconfigurable, equivalent circuit. In our model, we used the FPGA model ZYBO Zynq-XC7Z010 [49], which covers all of our requirements.

5.2.1 Signal schematic

Figure 6 shows the signals exchanged among the different modules of the hardware prototype, their origin and terminus, their nomenclature and the number of bits that comprise them. This prototype implements the functions described above for the centre model in the hardware and includes a large number of variables that will be translated into signals of different sizes.

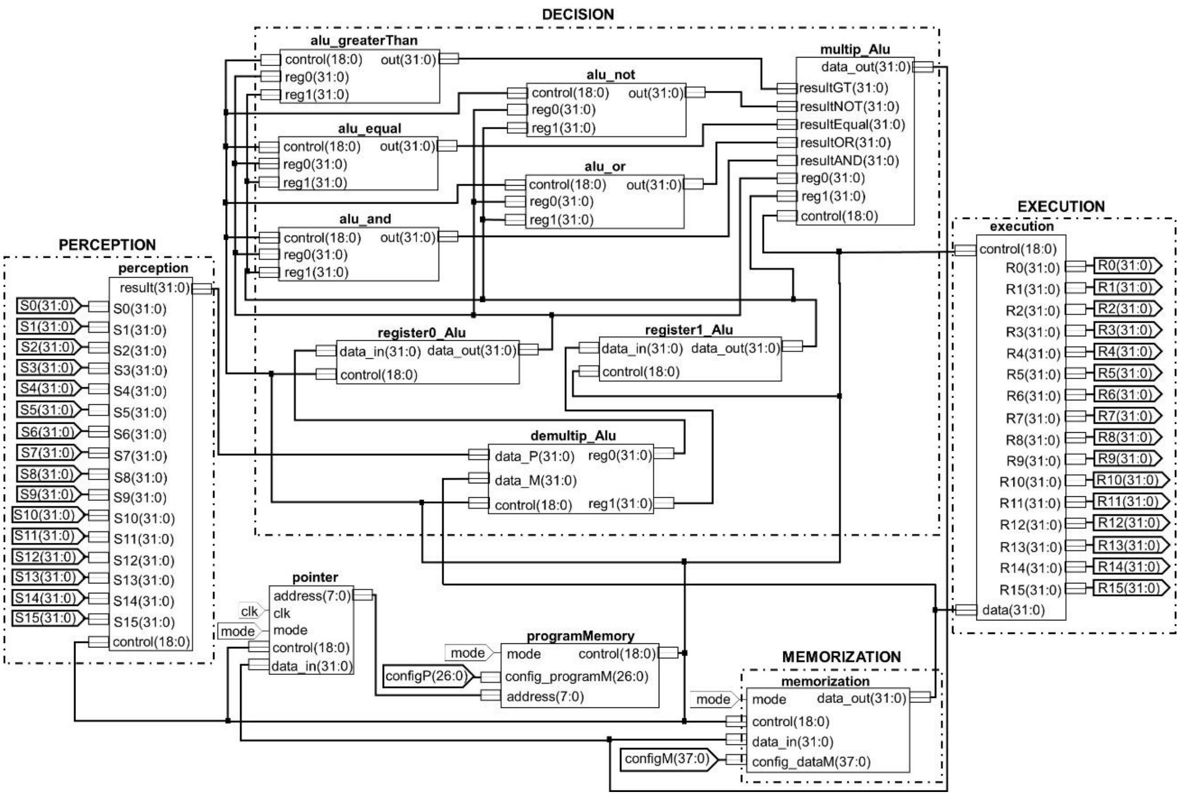

Figure 6. Hardware diagram of the CD centre.

After creating the hardware design shown in Figure 6, we obtained an equivalent circuit implemented in FPGA, which uses the different resources summarised in Table 7.

Table 7. Summary of device utilisation

Logic utilisation	Used	Available	Utilisation
Number of slice registers	7293	35200	20%
Number of slice LUTs	5050	17600	28%
Number of fully used slice LUT-FF pairs	3497	8846	39%

The hardware incorporated 7293 of the 35,200 available slice registers available in the FPGA, along with 5050 of 17,600 Look-Up Tables. It is fortunate that only a small fraction of the available resources are used given that the Look-Up Tables are fundamental for creating logical functions and critical in estimating the energy consumed by the proposed design [46, 47]. Additionally, the prototype uses 3497 of 8846 LUT-FF pairs, or 39% of the available resources. Even taking these demands into account, our hardware prototype still requires few resources.

5.3 Isolated hardware testing

To ensure that the proposed prototype functions similarly to the biological system desired, a testing and validation process was designed that utilised a group of the signals obtained from clinical experiments carried out during the modelling of the neuroregulatory system and which were previously used to validate the function of the previously developed LUT model [6, 10].

During the testing procedure, we will carry out implementation of the CD centre in the reconfigurable hardware and will submit said implementation to isolated simulation of its behaviour using the above group of signals. The expected result should coincide with the behaviour of the CD centre obtained during the modelling phase.

5.3.1 Defining the group of test signals

The first step in carrying out the tests for the hardware design of the CD centre was to extract a representative sample of the group of signals obtained during the modelling stages. This subgroup consists of values for each of the signals involved in the execution of the CD centre: ${}^{D}A_{CD}$, ${}^{M}I_{CD}$ and ${}^{R}I_{CD}$. The values were extracted from tests and validation of the theoretical model and were obtained from clinical tests during the modelling stage [6] and made available in the dataset [50]. The frequency of both the data and the FPGA is 1 Hz. These continuous signals (as a result of their origin from a biological system) were discretised by multiplying each value by 10^3 for use in the digital hardware prototype:

$$Value_{FixedPoint} = wholePart\ (Value_{FloatingPoint} \times 10^3) \qquad (14)$$

For simplicity, the following graphs show the time course of the values of the signals, which also reflect the urodynamics of the system. Figure 7(a) shows the changes in signal ${}^{D}A_{CD}$, Figure 7(b) the changes in signal ${}^{M}I_{CD}$, and Figure 7(c) the values of signal ${}^{R}I_{CD}$.

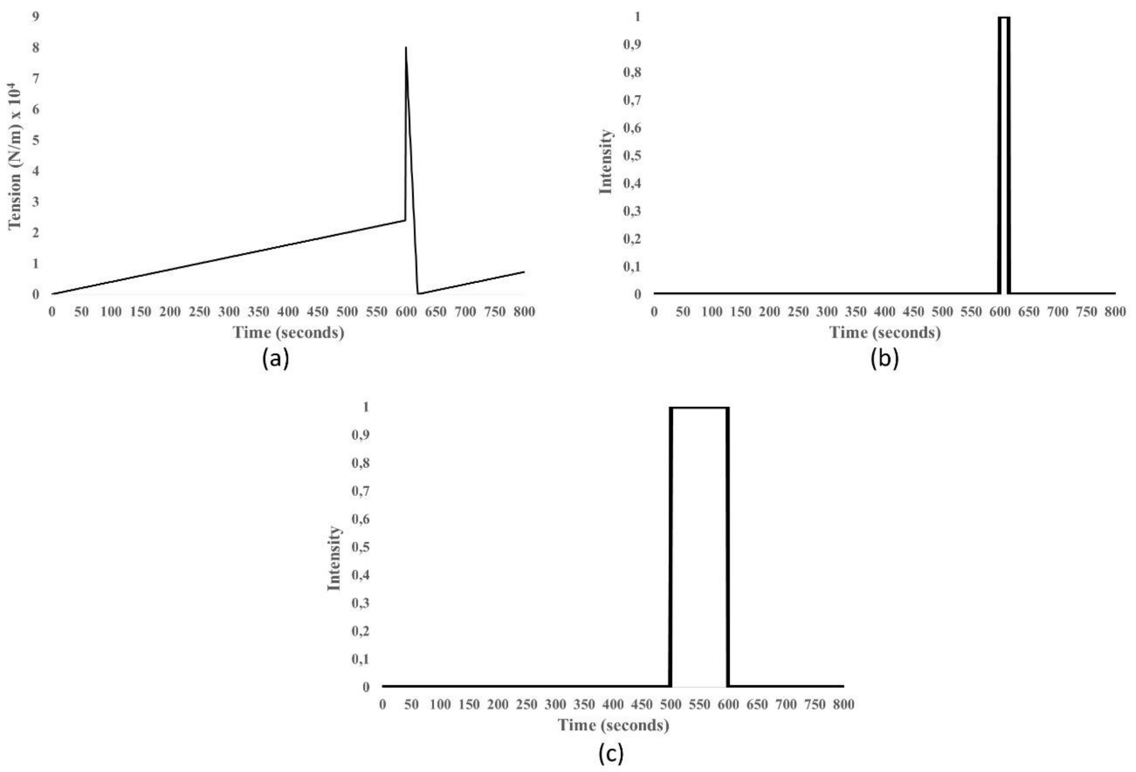

Figure 7. a) Afferent signal ${}^{D}A_{CD}$, **b)** Internal signal ${}^{M}I_{CD}$, **c)** Internal signal ${}^{R}I_{CD}$.

Signal ${}^{D}A_{CD}$ (Figure 7(a)) shows how the tension in the detrusor muscle increases gradually until it experiences a sharp increase, followed by an immediate decrease to the initial value. Signal ${}^{M}I_{CD}$ (Figure 7(b)) is inactive until it experiences a brief increase in intensity, followed by a swift return to the initial value. Lastly, signal ${}^{R}I_{CD}$ (Figure 7(c)) is also inactive for some time before increasing in intensity for a relatively short duration, then returning to its initial value. This dynamic describes how the vesical pressure in the detrusor increases as an individual begins sending the desire to urinate, until a point is reached where the person must voluntarily engage in retention. Once the maximum tension value is hit, the person experiences such intense tension that a signal is produced to trigger manual urination.

In addition to the group of input signals, we also defined the values associated with the thresholds $^{D}H_{CD1}$ and $^{D}H_{CD2}$. Although these values are specific to each individual, they can be fixed to produce normal LUT behaviour for a person without any dysfunctions; here, we used the values 2.00 and 18.2, respectively.

We lastly also selected the group of values that should be obtained as outputs in the signals $^{CD}I_{PA}$ and $^{CD}I_{PS}$, the time course of which are shown in Figures 8(a) and 8(b). These values were also discretised using the method described above for the input signals.

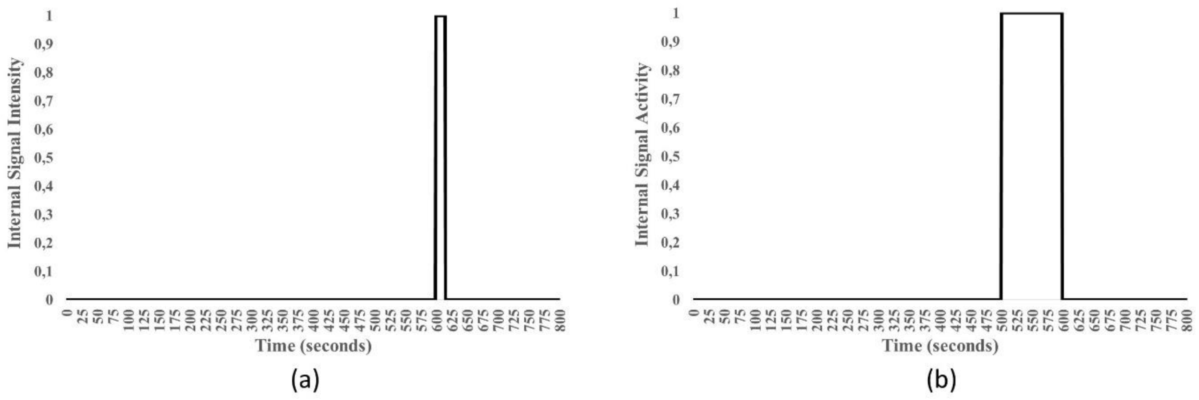

Figure 8. a) Internal signal $^{CD}I_{PA}$, **b)** Internal signal $^{CD}I_{PS}$.

5.3.2 Isolated hardware simulation

Simulating the hardware consisted of submitting the FPGA circuit to the sequence of input signals described previously and recovering the signals the system generates as a result, namely, $^{CD}I_{PA}$ and $^{CD}I_{PS}$.

Before initiating the simulation, both the program memory and data memory must be configured. The *mode* signal, which is responsible for indicating the system state, must be set to configuration mode, allowing the storage of needed configurations into both memories. The *configP* signal tells the system to store the program (Appendix 1) necessary for executing the CD centre, while the *configM* signal allows the data memory to store the values required for correct functioning of the centre. Among these values are the thresholds and all possible values of the output signal, resulting in the schematic for the data memory shown in Table 5. Once the configuring process is completed, the *mode* signal is set to 1, indicating that it is now in the working state and that the CD centre should begin functioning.

Figure 9 shows the execution performed by the hardware; the entire execution period is not shown—only the time span containing the peak detrusor tension and the generation of signals that result in voluntary tension and subsequent urination. The figure shows all of the input and output signals and their behaviour, allowing them to be studied in order to understand their changes over time.

The figure identifies six important epochs in the evolution of the signals. In epoch 1, the output signal $^{CD}I_{PA}$, which the CD centre outputs to the preoptic area (PA), is held at 0 and corresponds to the individual storing their urine. This lasts until epoch 2, where the signal takes on the value 1; it is activated when the individual enters the urination phase. At epoch 3, the output signal once again deactivates; urination has ceased, and urine storage resumes.

Over epoch 4, the internal output signal ${}^{CD}I_{PS}$, which the CD centre sends to the pontine storage (PS) centre, remains at 0 while the individual maintains storage of urine. At epoch 5, the signal is activated as the individual begins to feel the urge to urinate but finds themselves unable to do so in their current situation. During epoch 6, the signal deactivates, corresponding to the moment the individual finds a good place to urinate; the act of urination causes the signal to return to 0.

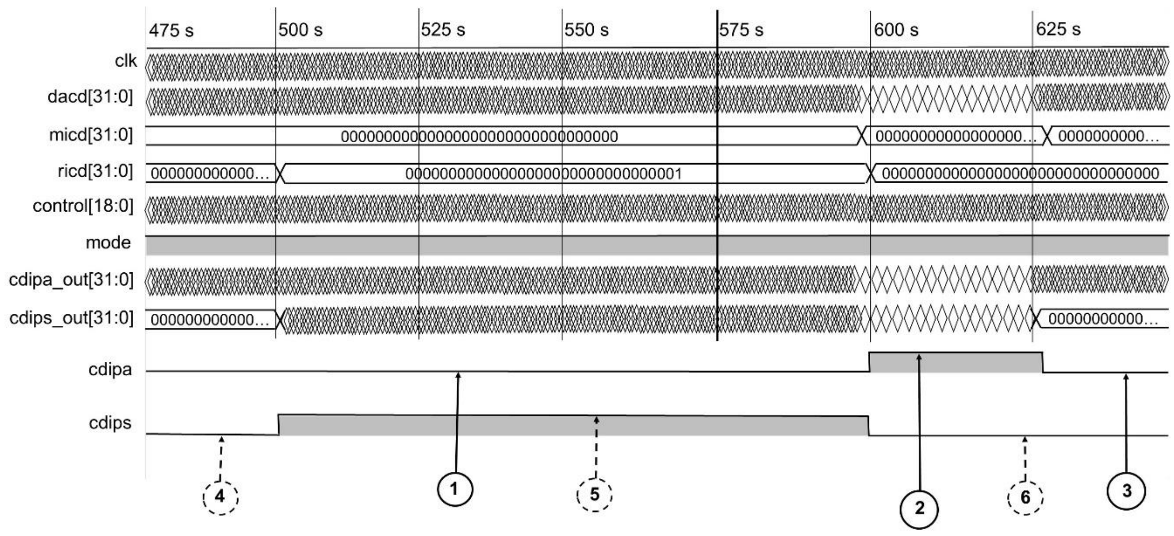

Figure 9. Values of the output signals ${}^{CD}I_{PA}$ and ${}^{CD}I_{PS}$.

By studying these results, we can conclude that the synthesised hardware reproduces the behaviour of the CD centre; the signals resulting from the execution of the hardware coincide with the values expected and shown in Figures 8(a) and 8(b).

After the hardware synthesis on FPGA, we have estimated the worst case of power and cooling system, using Xilinx Power Estimator [52]. This estimate produces a maximum consumption value of 1.48w and a maximum junction temperature of 40.6ºC. This allows that using a battery of about 3000mA, it is possible to obtain continuous use of more than 200h of the chip. Another work to be done, after we synthesize the complete chip, will be the optimization of the hardware to achieve a very low consumption of the hardware, although this step will not be done until the synthesis of the complete model is achieved, since we do not want to sacrifice system functionality.

6. Conclusions

This project has taken another step forward in the development of a system-on-chip analogue for the neuroregulatory system. Based on the functioning and composition of the cortical-diencephalic centre, a human nerve centre of the neuroregulatory system, we have proposed an original design for a generic hardware architecture capable of emulating the behaviour of both the CD centre and any other neuronal centre of the neuroregulatory system by varying the hardware programming.

The proposed hardware has been validated by implementing a prototype in FPGA and simulating the hardware design of the CD centre using data obtained from real patients. The results of this validation show that the behaviour of the resulting urodynamics curves coincides with the results of tests carried out on the existing theoretical model and is consistent with results obtained from clinical trials.

The model proposed here can be used to correct dysfunctions in one or more nerve centres that form part of the neuroregulatory system. This is one of the most important benefits of this proposal. Until now, treating dysfunctions required the use of black box solutions that suggest replacing almost the entire neuroregulatory system, including functioning centres, instead of focusing on the specific centre or centres that are not malfunctioning.

Synthesising the hardware design using FPGA demonstrates that few resources are necessary for its implementation, rendering possible the ultimate development of the neuroregulatory system based on a SoC with a large number of nerve centres.

In the short term, this hardware design should continue to be validated and analysed with other nerve centres. Such work would allow for the study of the hardware requirements that are necessary to implement each of the nerve centres that comprise the neuroregulatory system. This analysis would in turn serve as the basis for maximum generalisation of the proposed model. Subsequently, a chip could be designed that integrates multiple nerve centres working together to generate a neuroregulated system. In the longer term, we propose the creation of a chip with completely configurable nerve centres that can control both biological and artificial systems that require or can adjust to this type of neuroregulatory control while also achieving the construction of systems that can aid medical professionals in decision making and detecting malfunctions.

References

[1] G. Indiveri and T.K. Horiuchi. Frontiers in neuromorphic engineering. Frontiers in neuroscience. 5 (2011), pp.118.

[2] C. Mead. Neuromorphic electronic systems. Proceedings of the IEEE, 78(10), (1990), pp. 1629-1636.

[3] T. Kawasetsu, R. Ishida, T. Sanada, and H. Okuno. A hardware system for emulating the early vision utilizing a silicon retina and SpiNNaker chips. 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS), (2014), pp. 552–555.

[4] S. W. Moore, P. J. Fox, S. J. T. Marsh, A. T. Markettos, and A. Mujumdar. Bluehive - A field-programable custom computing machine for extreme-scale real-time neural network simulation. 2012 IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), (2012), pp. 133–140.

[5] L. Bendaouia, S.M. Karabernou, L. Kessal, H. Salhi, F. Ykhlef. FPGA-implementation of a bio-inspired medical hearing aid based DWT-OLA. 2014 International Conference on Audio, Language and Image Processing (ICALIP), (2014), pp.806-811.

[6] A. Soriano. Modeling and Simulation of the Neural Regulator of the Lower Urinary Tract, PhD Thesis, Ph.D. Dissertation, Dept. Comp. Tech., University of Alicante, Alicante, Spain, (2001). http://hdl.handle.net/10045/4069

[7] J.M. García Chamizo, F. Maciá Pérez, A. Soriano Payá, D. Ruiz Fernández. Simulation of the Neuronal Regulador of the Coger Urinary Tract Using a Multiagent System. Lecture Notes in Computer Science. Springer-Verlag, 2687 (2003), 591-598.

[8] D. Ruiz Fernández, J.M. García Chamizo, F. Maciá Pérez, A. Soriano Payá, Modeling the Distributed Control of the Lower Urinary Tract Using a Multiagent System, Lecture Notes in Artificial Intelligence, Springer-Verlag 3131 (2004) 104 -114.

[9] D.G. Méndez, Modeling and simulation of neurobehavioral lower urinary tract. System diagnostic support, Doctoral Thesis. Department of Computer Technology, University of Alicante, in September 2008. http://hdl.handle.net/10045/10325

[10] J.M. García, J. Romero, F. Macia, A. Soriano. Modeling and simulation of neuronal regulator of the lower urinary tract. Urodynamics Applied, vol.15 n0.2, (2002), pp 591-598

[11] A. S. Paya, D. R. Fernández, D. Gil, J. M. García Chamizo, and F. M. Pérez. Mathematical modelling of the lower urinary tract. Computer methods and programs in biomedicine, vol. 109, no. 3, (2013), pp. 323–338.

[12] D. Ruiz Fernández, Modeling of self-regulation of biological systems. Characterization and correction of neurogen urinary dysfunctions in human, Ph.D. dissertation, University of Alicante, Alicante, Spain, (2003). http://hdl.handle.net/10045/3964

[13] J.M. García-Chamizo, A. Soriano-Paya, F. Maciá-Perez, D. Ruiz-Fernadez. Modelling of the sacral micturition centre using a deliberative intelligent agent. Proceedings of the IV International Workshop on Biosignal Interpretation (BSI 2002). Como (Italy). (2002), pp. 451-454.

[14] R. Fernández and S. Payá, Robust Modelling of Biological Neuroregulators, Engineering in Medicine and Biology 27th Annual Conference, no. 5, (2005), pp. 2981–2984.

[15] García, J.M., Macía, F., Soriano, A., Flórez, F.: A Multi-Agent System uses ArtificialNeural Networks to Model the Biological Regulation for the Lower Urinary Tract, WSES Conference on Neural Networks and Applications. Interlaken (Switzerland). (2002), 162-167.

[16] D. Gil, M. Johnsson, J. M. García Chamizo, A. S. Paya, and D. R. Fernández, Modelling of urological dysfunctions with neurological etiology by means of their centres involved, Applied Soft Computing, vol. 11, no. 8, (2011), pp. 4448–4457.

[17] J. Ferber. Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison-Wesley, 1999.

[18] D. R. Fern, J. M. Garcia, and F. Macia, Modelling of Dysfunctions in the Neuronal Control of the Lower Urinary Tract, International Work-Conference on the Interplay Between Natural and Artificial Computation IWINAC 2005, (2005), pp. 203–212.

[19] E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri, Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems, Proceedings of the IEEE, vol. 102, no. 9, (2014), pp. 1367–1388.

[20] J. C. Flaherty, K. S. Guillory, M. D. Serruya, and A. H. Caplan, Transcutaneous implant. US Patent US7647097B2, 2003.

[21] M. A. Moffitt, R. Carbunaru, T. K. Whitehurst, and A. E. Mann, Electrode contact configurations for an implantable stimulator. US Patente US7702385B2, 2005.

[22] A. B. Krishnan, K. P. Peeyush, Blood Group Determination Using Vivado System Generator in Zynq SoC. 7th WACBE World Congress on Bioengineering 2015. Springer International Publishing, (2015), pp. 166-169.

[23] F. Corradi, D. Zambrano, M. Raglianti, G. Passetti, C. Laschi, and G. Indiveri, Towards a Neuromorphic Vestibular System, Biomedical Circuits and Systems, IEEE Transactions on, vol. 8, no. 5, (2014), pp. 669–680.

[24] H. Soleimani, A. Ahmadi, and M. Bavandpour, Biologically Inspired Spiking Neurons: Piecewise Linear Models and Digital Implementation. Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 59, no.12, (2012), pp. 2991-3004.

[25] S. Romero, F. J. Pelayo, C. A. Morillas, A. Martínez and E. Fernandez, Reconfigurable retina like preprocessing platform for cortical visual neuroprosthesis. Handbook of Neural Engineering, (2007), pp 267-279.

[26] B. Girau, C. Torres-Huitzil, Massively distributed digital implementation of an integrate-and-fire LEGION network for visual scene segmentation, Neurocomputing, Volume 70, Issues 7-9, (2007), pp 1186-1197.

[27] T. Kawasetsu, R. Ishida, T. Sanada, and H. Okuno, A hardware system for emulating the early vision utilizing a silicon retina and SpiNNaker chips, in 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS), (2014), pp. 552–555.

[28] A. Martínez, F.J. Pelayo, C. Morillas, S. Romero, B. Pino, Automatic Generation of Bio-inspired Retina-Like Processing Hardware. Computational Intelligence and Bioinspired Systems Volume 3512, (2005), pp 527-533.

[29] A. Martinez, S. Romero, E. Ros, A. Prieto and FJ. Pelayo, Reconfigurable hardware implementation of a model of retina. Proceedings of the Second Conference on Field Programmable Logic and Applications (JCRA'2002), (2002), pp. 97-101.

[30] S. Mota, E. Ros, J. Díaz, R. Agis and F. de Toro, Bio-inspired motion-based object segmentation. In International Conference Image Analysis and Recognition, Springer, Berlin, Heidelberg, (2006), pp. 196-205.

[31] A. Mishra, and A.E. Hubbard. A cochlear filter implemented with a field-programmable gate array. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, no.1, (2002), pp. 54-60.

[32] M. P. Leong and C. T. Jin, An FPGA-Based Electronic Cochlea, Journal on Applied Signal Processing, vol. 7, (2003), pp. 629–638.

[33] P. Zicari, Efficient and high performance FPGA-based rectification architecture for stereo vision. Microprocessors and Microsystems, vol. 37, no 8, (2013), p. 1144-1154.

[34] Y. Meng, A Mobile Vision System with Reconfigurable Intelligent Agents. International Joint Conference on Neural Networks IJCNN '06, (2006). pp. 1483-1488.

[35] A. Dinu, M.N. Cirstea, and S.E. Cirstea, Direct Neural-Network Hardware-Implementation Algorithm. IEEE Transactions on Industrial Electronics, vol. 57, no. 5, (2010), pp. 1845-1848.

[36] Y. Ji, F. Ran, C. Ma, and D. J. Lilja. A hardware implementation of a radial basis function neural network using stochastic logic. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition. EDA Consortium, San Jose, CA, USA, (2015), pp 880-883.

[37] H. Soleimani, A. Ahmadi and M. Bavandpour. Biologically Inspired Spiking Neurons: Piecewise Linear Models and Digital Implementation. IEEE Transactions on circuits and systems, regular papers, vol. 59, no. 12, (2012), pp 2991-3004.

[38] M. Samie, G. Dragffy, A. M. Tyrrell, T. Pipe, and P. Bremner. Novel Bio-Inspired Approach for Fault-Tolerant VLSI Systems. IEEE Transactions on very large scale integration (vlsi) systems, vol. 21, no. 10, (2013), pp 1878-1891.

[39] L. Rodriguez, L. Fiack, and B. Miramond. A neural model for hardware plasticity in artificial vision systems. Design and Architectures for Signal and Image Processing (DASIP), 2013 Conference on, (2013), pp. 30-37.

[40] L. Fiack, L. Rodriguez, and B. Miramond. Hardware design of a neural processing unit for bio-inspired computing. IEEE 13th International New Circuits and Systems Conference (NEWCAS), (2015), pp. 1-4.

[41] M.A. Tsompanas, G.Ch. Sirakoulis. Modeling and hardware implementation of an amoeba-like cellular automaton. Bioinspiration & Biomimetics, vol. 7, no. 3, (2012), p. 036013.

[42] M.V. Kinder, E.H.C. Bastiaanssen, R.A. Janknegt, E. Marani, The Neuronal Control of the Lower Urinary Tract: A Model of Architecture and Control Mechanisms, Archives of Physiology and Biochemistry, 107 (1999), pp. 203-222.

[43] J. Morrison, L. Birder, M. Craggs, W.C. de Groat, J. Downie, M. Drake, C.J. Fowler, K. Thor, Neural control, Incontinence 1, (2005), pp. 363‑422.

[44] D. Gil, A. Soriano, and C. A. Montejo, Embedded System For Diagnosing Dysfunctions In The Lower Urinary Tract. Proceedings of the 2007 ACM symposium on Applied computing, (2007), pp. 1695–1699.

[45] J. Díaz, E. Ros, S. Mota and R. Agis, "Real-time embedded system for rear-view mirror overtaking car monitoring," Lecture Notes in Computer Science, Springer-Verlag, 4017, (2006), pp. 385-394.

[46] A. Rodriguez, J. Valverde, C. Castanares, J. Portilla, E. de la Torre, T. Riesgo, Execution Modeling in self-aware FPGA-based architectures for efficient resource management, 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), (2015), pp 1 – 8.

[47] L. Wang; X. Wang; T. Wang; Q. Yang, High-Level Power Estimation Model for SOC with FPGA Prototyping, in Fourth International Conference on Computational Intelligence and Communication Networks (CICN), (2012) pp.491-495.

[48] F. Maciá Pérez, L. Zambrano-Mendez, J.V. Berná-Martínez, R. Sepúlveda Lima. Hardware design of the cortical-diencephalic centre of the lower urinary tract neuroregulator system. Computers in biology and medicine, vol. 77, (2016), pp. 156-172

[49] Zynq Z-7010 Trainer Board at Digilent, product description, (2018), https://store.digilentinc.com/zybo-zynq-7000-arm-fpga-soc-trainer-board/

[50] F. Maciá, J.V. Berna-Martinez, Database for analyzing the operation of the center cortico-Diencephalic – Internal and afferent signals, (2016), http://rua.ua.es/dspace/handle/10045/56407.

[51] F. Maciá Perez, L. Zambrano-Mendez, J.V. Berna-Martinez, R. Sepúlveda Lima. Configuration of the program memory for the CD centre, (2018), http://hdl.handle.net/10045/75487

[52] Xilinx Power Estimator tool estimate power consumption. V. 7 Series and Zynq®-7000 (2018). https://www.xilinx.com/products/technology/power/xpe.html

Appendices

Appendix 1: Configuration of the program memory for the CD centre [51].

Pos.	Operation code	Source block	Internal _Add	Target block	Internal _Add	Description
0	001	00	000000	11	000001	LOAD INPUT REG 0 ALU REG 1
1	001	01	000000	11	000000	LOAD MEMORY DIR 0 ALU REG 0
2	110	10	000011	10	111101	> ALU REG 1 y 0 MEMORY DIR 3D
3	001	00	000001	11	000000	LOAD INPUT REG 1 ALU REG 0
4	001	00	000010	11	000001	LOAD INPUT REG 2 ALU REG 1
5	100	10	000000	10	111110	NOT ALU REG 0 MEMORY DIR 3E
6	100	10	000001	10	111111	NOT ALU REG 1 MEMORY DIR 3F
7	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
8	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
9	010	10	000011	10	111101	AND ALU REG 0 y 1 MEMORY DIR 3D
10	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
11	001	01	111111	11	000001	LOAD MEMORY DIR 3F ALU REG 1
12	010	10	000011	10	111101	AND ALU REG 0 y 1 MEMORY DIR 3D
13	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
14	001	01	100000	11	000001	LOAD MEMORY DIR 20 ALU REG 1
15	010	10	000011	10	111110	AND ALU REG 0 y 1 MEMORY DIR 3E
16	001	01	111010	11	000000	LOAD MEMORY DIR 3A ALU REG 0
17	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
18	010	10	000011	01	000000	AND ALU REG 0 y 1 STACK P REG 0
19	001	00	000000	11	000000	LOAD INPUT REG 0 ALU REG 0
20	001	01	000000	11	000001	LOAD MEMORY DIR 0 ALU REG 1
21	110	10	000011	10	111101	> ALU REG 1 y 0 MEMORY DIR 3D
22	101	10	000011	10	111110	= ALU REG 1 y 0 MEMORY DIR 3E
23	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
24	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
25	011	10	000011	10	111111	OR ALU REG 1 y 0 MEMORY DIR 3F
26	001	00	000000	11	000001	LOAD INPUT REG 0 ALU REG 1
27	001	01	000001	11	000000	LOAD MEMORY DIR 1 ALU REG 0
28	110	10	000011	10	111101	> ALU REG 1 y 0 MEMORY DIR 3D
29	001	00	000001	11	000000	LOAD INPUT REG 1 ALU REG 0
30	001	00	000010	11	000001	LOAD INPUT REG 2 ALU REG 1
31	100	10	000000	10	111110	NOT ALU REG 0 MEMORY DIR 3E
32	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
33	010	10	000011	10	111101	AND ALU REG 0 y 1 MEMORY DIR 3D
34	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
35	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
36	010	10	000011	10	111101	AND ALU REG 0 y 1 MEMORY DIR 3D
37	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
38	001	01	111111	11	000001	LOAD MEMORY DIR 3F ALU REG 1
39	010	10	000011	10	111101	AND ALU REG 0 y 1 MEMORY DIR 3D
40	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
41	001	01	100000	11	000001	LOAD MEMORY DIR 20 ALU REG 1
42	010	10	000011	10	111110	AND ALU REG 0 y 1 MEMORY DIR 3E
43	001	01	111011	11	000000	LOAD MEMORY DIR 3B ALU REG 0
44	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
45	010	10	000011	01	000000	AND ALU REG 0 y 1 STACK P REG 0
46	001	00	000000	11	000000	LOAD INPUT REG 0 ALU REG 0
47	001	01	000000	11	000001	LOAD MEMORY DIR 0 ALU REG 1
48	110	10	000011	10	111101	> ALU REG 1 y 0 MEMORY DIR 3D
49	101	10	000011	10	111110	= ALU REG 1 y 0 MEMORY DIR 3E
50	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
51	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
52	011	10	000011	10	111111	OR ALU REG 1 y 0 MEMORY DIR 3F
53	001	00	000000	11	000001	LOAD INPUT REG 0 ALU REG 1

54	001	01	000001	11	000000	LOAD MEMORY DIR 1 ALU REG 0
55	110	10	000011	10	111101	> ALU REG 1 y 0 MEMORY DIR 3D
56	001	01	111111	11	000000	LOAD MEMORY DIR 3F ALU REG 0
57	001	01	111101	11	000001	LOAD MEMORY DIR 3D ALU REG 1
58	010	10	000011	10	111101	AND ALU REG 1 y 0 MEMORY DIR 3D
59	001	00	000001	11	000000	LOAD INPUT REG 1 ALU REG 0
60	001	01	111101	11	000001	LOAD MEMORY DIR 3D ALU REG 1
61	010	10	000011	10	111101	AND ALU REG 1 y 0 MEMORY DIR 3D
62	001	00	000010	11	000000	LOAD INPUT REG 2 ALU REG 0
63	100	10	000000	10	111110	NOT ALU REG 0 MEMORY DIR 3E
64	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
65	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
66	010	10	000011	10	111101	AND ALU REG 1 y 0 MEMORY DIR 3D
67	001	00	000000	11	000000	LOAD INPUT REG 0 ALU REG 0
68	001	01	000001	11	000001	LOAD MEMORY DIR 1 ALU REG 1
69	110	10	000011	10	111110	> ALU REG 1 y 0 MEMORY DIR 3E
70	101	10	000011	10	111111	= ALU REG 1 y 0 MEMORY DIR 3F
71	001	01	111110	11	000000	LOAD MEMORY DIR 3E ALU REG 0
72	001	01	111111	11	000001	LOAD MEMORY DIR 3F ALU REG 1
73	011	10	000011	10	111110	OR ALU REG 1 y 0 MEMORY DIR 3E
74	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
75	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
76	011	10	000011	10	111101	OR ALU REG 1 y 0 MEMORY DIR 3D
77	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
78	001	01	100000	11	000001	LOAD MEMORY DIR 20 ALU REG 1
79	010	10	000011	10	111110	AND ALU REG 1 y 0 MEMORY DIR 3E
80	001	01	111100	11	000000	LOAD MEMORY DIR 3C ALU REG 0
81	001	01	111110	11	000001	LOAD MEMORY DIR 3E ALU REG 1
82	010	10	000011	01	000000	AND ALU REG 1 y 0 STACK P REG 0
83	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
84	001	10	000000	10	100000	LOAD ALU REG 0 MEMORY DIR 20
85	001	01	000010	00	000000	LOAD MEMORY DIR 2 OUTPUT REG 0
86	001	01	000011	00	000001	LOAD MEMORY DIR 3 OUTPUT REG 1
87	000	10	000000	01	000000	Ø ALU REG 0 STACK P REG 0
88	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
89	001	10	000000	10	100000	LOAD ALU REG 0 MEMORY DIR 20
90	001	01	000100	00	000000	LOAD MEMORY DIR 4 OUTPUT REG 0
91	001	01	000101	00	000001	LOAD MEMORY DIR 5 OUTPUT REG 1
92	000	10	000000	01	000000	Ø ALU REG 0 STACK P REG 0
93	001	01	111101	11	000000	LOAD MEMORY DIR 3D ALU REG 0
94	001	10	000000	10	100000	LOAD ALU REG 0 MEMORY DIR 20
95	001	01	000110	00	000000	LOAD MEMORY DIR 6 OUTPUT REG 0
96	001	01	000111	00	000001	LOAD MEMORY DIR 7 OUTPUT REG 1
97	000	10	000000	01	000000	Ø ALU REG 0 STACK P REG 0