Hardware Realization of Biological Mechanisms Using VHDL and FPGAs

*N. Botros, M. Akaaboune, J. Alghazo, and M. Alhreish

Southern Illinois University at Carbondale
Department of Electrical Engineering
Carbondale, IL 62901
*e-mail: botrosn@siu.edu

ABSTRACT

In this paper we present a hardware realization of a simplified blood-pressure regulation mechanism. The mechanism has been proposed and then modeled by using Hardware Description Language (VHDL). After verifying the simulation results, Computer Aided Design (CAD) tools are implemented to download the simulation into a Xilinx FPGAs chip. The chip has been tested by comparing the behavior of the signals on its pins representing the blood pressure and other parameters of the mechanism with that of the proposed mechanism. Our results show that the chip mimics the mechanism.

Keywords: **VHDL, FPGAs, System on a chip, Biological mechanisms, Modeling and simulation.**

1. INTRODUCTION

To understand the function and/or malfunction of biological mechanisms and to develop a drug to treat the malfunction, researchers need to perform a thorough investigation of the mechanism. Due to ethical constraints, medical experiments on humans or animals are not allowed or extremely limited. The ultimate goal of this paper is to realize (build on an Integrated electronic chips) biological mechanisms such as the excretion system (kidney), Molecular Transcription (genetics), or cellular pathways. After realization, the hardware (the chip) will mimic the function of the realized mechanism. The chip will offer a unique opportunity for researchers to explore the function of the mechanism and the effect of external agents such as drugs, hormones, or chemical transmitters..

The realization is done through implementation of Computer Aided Design (CAD) tools including Hardware Descriptive Language (VHDL) to download the design of the hardware into Xilinx, or compatible Field Programmable Arrays (FPGAs) chips. VHDL is an extremely powerful tool that is used to model and realize complex systems [2]-[5]. VHDL has gone through a tremendous development and upgrading, it offers a unique tool to model complex systems through knowledge of the system behavior, [3]. By specifying the behavior of the output of the system with the change of its inputs, VHDL tools can model the system and convert it to Integrated Circuit (ICs) blocks. The model can be displayed and tested on the screen of a computer, and if it satisfies the requirement of the user, it can be downloaded into a Field Programmable Gate Arrays (FPGAs) chip. The FPGAs is a chip manufactured with a modern solid state technology that allows a huge number of Integrated Circuits to be packed in a small area. The nowadays FPGAs can contain in a single chip more than 300 blocks that can be programmed to be combinational circuits (adders, subtractors, decoders,..) or sequential circuits (flip-flops, counters, memories,..)

The chip after realization mimics the selected biological mechanism. Values of the input signals or parameters of the mechanism such as blood pressure, Na^+ concentration, or affinity of binding are applied to user-selected pins on the chip. These values can be in digital form (discrete) or analog (continuous) form; an Analog-to-Digital converter (A/D) should be used if the input is in analog form. The output of the mechanism can be viewed at the user-selected output pins of the chip; the view can be in digital form or analog form by using a Digital-to-Analog (D/A) converter.

At present time there is an emerging technology,[1] close to the one proposed here. In this technology, the mechanisms are only software-simulated on the screen of the computer. In comparison with the software-simulation, the hardware realization proposed here has the following advantages: a) the chip, once is realized, is a stand-alone electronic circuit that does not need a host computer for operation, b)the chip is compact (1"x1"x.1" without socket and 1"x1"x.7" with socket) , c) the chip has an average cost of less than $30, d) The chip can be easily interfaced to: external signals or stimuli, external devices, or another chip through the input/output pins, e)the downloaded design can be easily modified by just changing the VHDL-program and re-downloading, f) the chip, because it is dedicated hardware without a host computer, can operate in real-time applications where a higher speed of operation is needed.

2. THE MECHANISM

To investigate the feasibility of our modeling technique, we consider a simplified version of the blood-pressure regulation mechanism done by the kidney. The mechanism

is described as follows: if the blood pressure (BP) decreases, the activity of the sympathetic neural (SN) system increases, the renal blood flow (RF) decreases, the sodium excretion (Na^+) decreases which leads to an increase (stabilization) in the blood pressure.

3. VHDL MODELING

We implement VHDL-behavior modeling to describe our mechanism. A flow chart of this description is shown in Figure 1. To express the inverse relationship between the activity of the Sympathetic Neural (SN) system and the blood pressure, we implement a simple formula SN = K1-BP where K1 is a constant. We do the same to express the inverse relationship between the Renal Flow (RF) and SN. For the relationship between the sodium excretion (Na^+)and RF, we use the formula Na^+ = 2 RF. The threshold value (the normal physiological value) of the blood pressure is assumed to be 17 units, for Na^+ is 14, and for RF is 14. A portion of the VHDL source code for the mechanism is shown below.

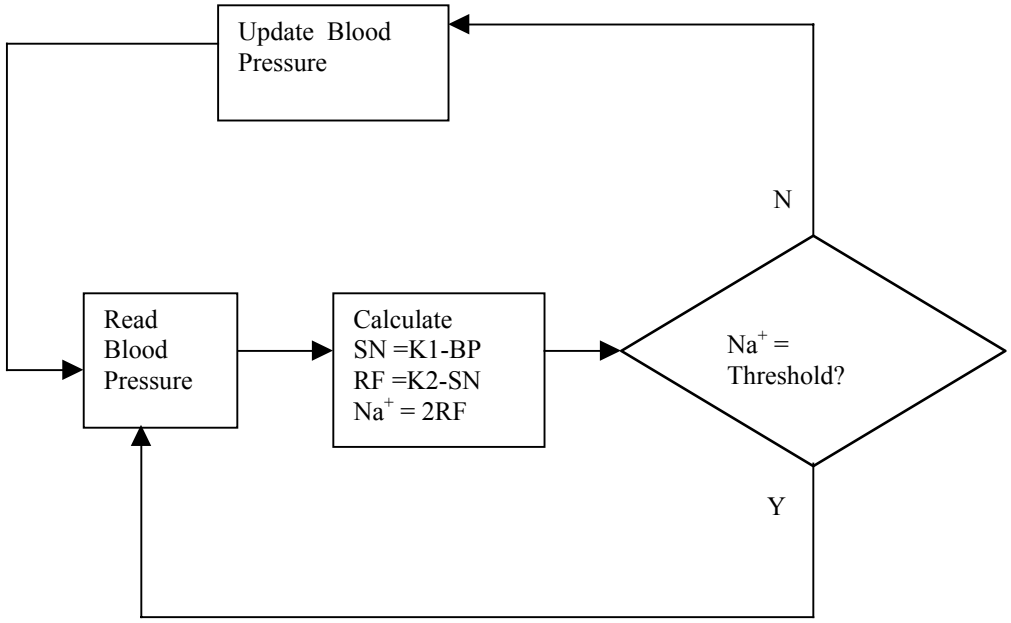

Figure 1. A Flow chart of modeling of the mechanism

The output of the VHDL simulation is shown in Figure 2. As shown in this Figure if the BP decreases, the Na^+ decreases and stabilizes the blood pressure.

4. HARDWARE REALIZATION

After verifying the simulation, we downloaded the simulation on a Xilinx FPGAs XC4005 chip using Xilinx CAD tools. The signals representing the BP, SN, RF, and Na^+ were assigned to selected pins of the chip. These signals behaved similarly to the simulation results shown in Figure 2.

The VHDL Program

```
library ieee;
use ieee.std logic 1164.all;
--In this entity, there are all the parameters that
--the program needs, especially the initial value
--of the blood pressure which is BP1, and for the
--adjusted values of the blood pressure is bp
entity BLOOD is
port ( clk: in std_logic; bp1: in std_logic_vector(5 downto
0); bp: buffer std_logic_vector(5 downto 0); sn: buffer
std_logic_vector(5 downto 0); rf: buffer
std_logic_vector(50 downto 0); na: buffer
std_logic_vector(5 downto 0) ) ;
end BLOOD;
architecture BH of BLOOD is
-- A procedure convert from binary to integer--
--procedure bin2i (bin: in std_logic_vector; int:out
--integer) is
--Another procedure to convert from integer to binary
--procedure int2b (int : in integer; bin: out
--std_logic_vector) is
-- Relation between BP and SN which is inversely
--proportional
function pro ( y: std logic vector) return std_logic_vector
is
constant k: integer:=17; constant XO: integer:=14; variable
rst, i : integer; variable j: std_logic_vector (5 downto 0) ;
begin
bin2i (y, i) ; rst:= (k+x0)-i; int2b (rst, j); return j; end pro;
-- A function that represent the relation between SN and
RF --which is inversely proportional.
function prop (y: std_logic_vector) return std_logic_
vector is
-- Relation between RF and NA " proportional"
function prom (y: std_logic_vector) return
std_logic_vector is
constant k : integer :=15; constant XO: integer:=14;
variable rst, i :integer; variable j: std_logic_vector (5
downto 0 );
begin
bin2i (y, i); rst:= 2*(x0+k)-2*i; int2b(rst,j);
return j; end prom;
-- Function to increment BP
function incr (y: std_logic_vector) return std_logic_vector
is
variable rft, w: integer; variable x: std_logic_vector( 5
downto 0);
begin
bin2i (y,w); rft:= w + 1; int2b(rft, x); return x;
end incr;
-- Another function that decrement BP
function decr(y:std_logic_vector) return std_logic_vector
is
-- The main program.
begin
process (clk)
constant bp0: std_logic_vector:="010001";
constant sn0: std_logic_vector:="001110";
constant rf0: std_logic_vector:="001110";
constant na0: std_logic_vector:="011100";
variable i,j: boolean; variable bpt:std_logic_vector (5
downto 0);
begin
bpt:=bp1;
if clk='1' then
if bpt=bp0 then bp<=bp0; sn<=sn0; rf<=rf0;na<=na0;
end if;
elsif bpt<bp0 then bp<=incr(bpt); sn<=pro(bp);
rf<=prop(sn); na<=prom(sn); bpt:=bp;
if bp=bp0 then i:=true; esle i:=false; end if;
if na<na0 then    if i=false then bp<=incr(bp);   end if;
if bp=bp0 then sn<=sn0; rf<=rf0; na<=na0;bp<=bp0;
end if;end if;
elsif bpt>bp0 then bp<=decr(bpt); sn<=pro(bp);
rf<=prop(sn); na<=prom(sn); bpt:=bp;
if bp=bp0 then j:=true; else j:=false; end if;
                    if na>na0 then   if j=false then
bp<=decr(bp); end if; if bp=bp0 then sn<=sn0; rf<=rf0;
na<=na0;bp<=bp0; end if;
                    end if; end if;
end process;
end BH;
```

5. REFERENCES

[1] SimBioSys Physiology Labs, www.critcon.com, 1999.

[2] T. Riesgo, Y. Torroja, and E. de la Torre, "Design Methodologies Based on Hardware Description Languages," IEEE Transactions on Industrial Electronics, vol. 46, no. 1, pp. 3-12, 1999.

[3]Moore, J. and Botros, N., "Design and Implementation of an Expandable Hopfield Neural Network Using VHDL Structural Modeling," International Journal of Robotics and Automation, vol.. 12, Issue 1, pp. 33-37, 1997.

[4] Y. Alquesi, and N. Botros, "A Floating-Point FFT Processor Using Field Programmable Gate Arrays," International Journal of Computers and Their Applications, vol. 3, no. 2, pp. 92-99, 1996.

[5] N. Botros, and M. Abdul-Aziz, M., "Hardware Implementation of an Artificial Neural Network Using Field Programmable Gate Arrays (FPGAs)," IEEE Transactions on Industrial Electronics, vol. 41, no. 6, pp. 665-667, 1994.

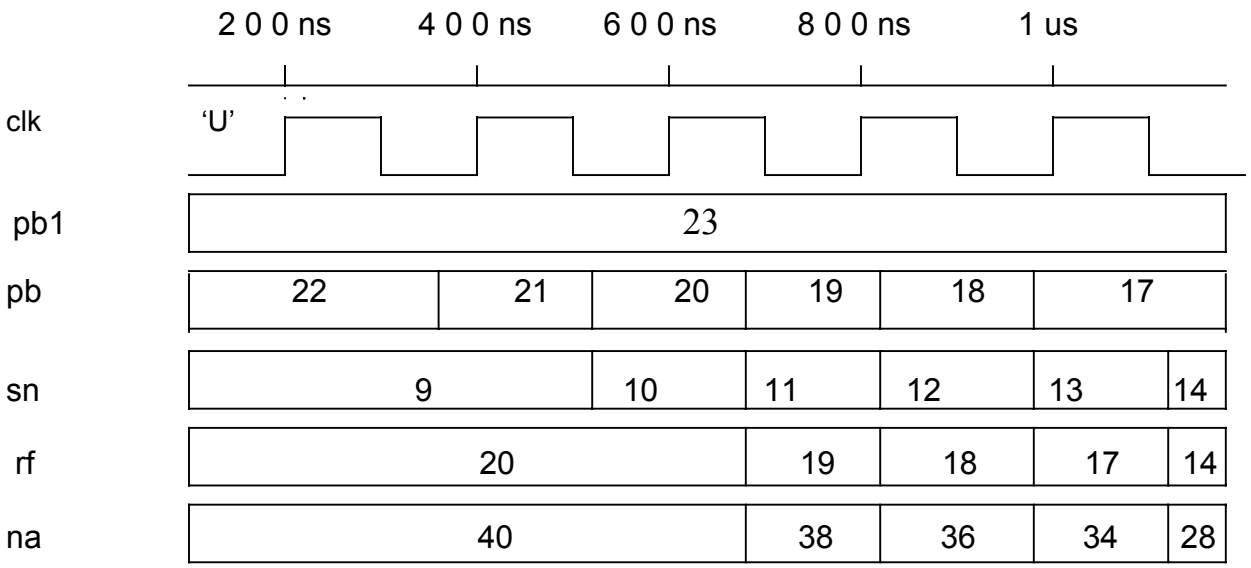

Figure 2. Simulation Results (redrawn from the VHDL screen)