TELKOMNIKA, Vol.10, No.2, June 2012, pp. 281~290
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

Design of FPGA Based Neural Network Controller for Earth Station Power System

Hanaa T. El-Madany[1*], Faten H. Fahmy[1], Ninet M. A. El-Rahman[1], and Hassen T. Dorrah[2]
[1]Electronics Research Institute, National Research Center Building, Cairo, Egypt
[2]Electrical Power & Machines Dept., Cairo University, Egypt
E-mail: hanaa_tolba@yahoo.com*

Abstrak

Otomasi dari pembangkitan kode bahasa deskripsi perangkat keras untuk model jaringan syaraf tiruan (JST) dapat mengurangi waktu implementasinya ke perangkat digital, sehingga menghemat biaya yang signifikan. Untuk menerapkan JST menjadi ke desain perangkat keras, diperlukan penerjemah model ke dalam struktur piranti. Bahasa VHDL digunakan untuk mendeskripsikan JST ke dalam perangkat keras. Kode VHDL juga telah diusulkan untuk implementasi JST ketika mempresentasikan hasil simulasi dengan aritmatika floating point dari stasiun bumi dan sistem tenaga satelit menggunakan perangkat lunak ModelSim ® PE 6.6 simulator. Integrasi antara MATLAB ® dan VHDL digunakan untuk menghemat waktu eksekusi dari komputasi. Hasil penelitian menunjukkan bahwa antara MATLAB dan VHDL terdapat kesesuaian yang baik untuk pembangkitan JST umpan cepat dan fleksibel yang mampu menangani operasi aritmatika floating point; jumlah minimum irisan CLB, dan kecepatan kinerja yang baik. Hasil sintesis FPGA diperoleh dengan skema RTL view dan skema teknologi dari Xilinix, dan jumlah kebutuhan minimum sumber daya perangkat keras diperoleh jika menggunakan Xilinix VERTIX5.

Kata kunci: *FPGA, jaringan syraf tiruan, stasiun bumi, VHDL*

Abstract

Automation of generating hardware description language code of neural networks models can highly decrease time of implementation those networks into a digital devices, thus significant money savings. To implement the neural network into hardware design, it is required to translate generated model into device structure. VHDL language is used to describe those networks into hardware. VHDL code has been proposed to implement ANNs as well as to present simulation results with floating point arithmetic of the earth station and the satellite power systems using ModelSim® PE 6.6 simulator tool. Integration between MATLAB® and VHDL is used to save execution time of computation. The results shows that a good agreement between MATLAB and VHDL and a fast and flexible feed forward NN which is capable of dealing with floating point arithmetic operations; minimum number of CLB slices; and good speed of performance. FPGA synthesis results are obtained with view RTL schematic and technology schematic from Xilinix tool. Minimum number of utilized resources is obtained by using Xilinix VERTIX5.

Keywords: *earth station, FPGA, neural network, VHDL*

1. Introduction

In the last three decades, numerous alternative control techniques, such as neural network control, have been proposed instead of conventional classical technique. Development of artificial neural networks (ANN's) theory has inspired new resources for possible implementation of better and more efficient control. ANN's have capability of learning the dynamical systems that estimated input-output functions. ANN's doesn't need mathematical modeling of the plants. ANN's have to be trained and they need some information (not based on mathematical model but sometimes taken measurement from plant) about the plant. Generally, input-output characterization or desired output of the plant or neuro controller is sufficient [1].

Designing and implementing intelligent systems has become a crucial factor for the innovation and development of better products of space technologies [2, 3]. Field programmable gate array (FPGA) is a digital device that owns reprogrammable properties and robust flexibility. For the neural network based instrument prototype in real time application, conventional specific VLSI neural chip design suffers the limitation in time and cost. With low precision artificial neural network design, FPGAs have higher speed and smaller size for real time application than the

Received December 19, 2011; Revised May 18, 2012; Accepted May 26, 2012

VLSI and DSP chips. So, many researchers have made great efforts on the realization of neural network (NN) using FPGA technique [4, 5].

The emphasis of this paper is concerned with implementing ANN using Hardware Description Language (VHDL) as well as to present simulation results with floating point arithmetic of the earth station power system controller.

2. Earth Station Power System Architecture

PV systems are most effective at remote sites off the electrical grid. In this system, a storage battery is needed. Excess energy produced during times with no or low loads charges the battery, while at times with no or too low solar radiation the loads are met by discharging it. A charge controller supervises the charge/discharge process in order to ensure a long battery lifetime. Figure 1 represents the main components of standalone PV system. By virtue of the variable nature of the energy source sun, one of the most expensive aspects of a PV power system is the necessity to build in system autonomy. Autonomy is required to provide reliable power during "worst case" situations, which are usually periods of adverse weather, seasonally low radiation values or unpredicted increased demand for power. The addition of autonomy could be accomplished by over sizing the PV array and greatly enlarging the battery storage bank - generally the two most costly system components. An additional benefit of this approach is the added system reliability provided by the incorporation of the back-up energy source [6].

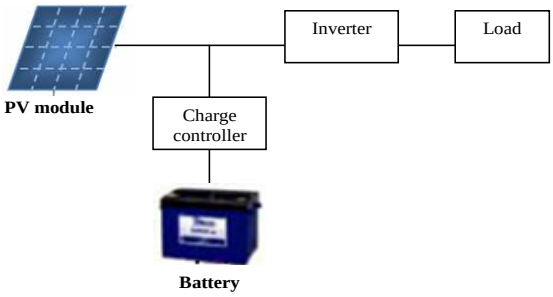

Figure 1. Earth station power system architecture

2.1. PV Generator Model

The traditional I-V characteristics of a solar array, when neglecting the internal shunt resistance, are given by the following equation [7]:

$$I_o = I_{ph} - I_{rs}\left(e^{qV_o/kTA} - 1\right) - \frac{V_o}{R_{sh}} \quad (1)$$

Where I_0 is the PV array output current (A), V_0 the PV array output voltage (V), q the charge of an electron, k the Boltzmann's constant in J/K, A the p–n junction ideality factor, T the cell temperature (K), and I_{rs} is the cell reverse saturation current. The factor A determines the cell deviation from the ideal p–n junction characteristics. The ideal value ranges between 1 and 5 [8].

The photocurrent I_{ph} depends on the solar radiation and the cell temperature as stated in the following equation:

$$I_{ph} = (I_{scr} + k_i(T - T_r))\frac{S}{S_t} \quad (2)$$

Where I_{scr} is the PV array short circuit current at reference temperature and radiation (A), ki the short circuit current temperature coefficient (A/K), S_t is the standard solar radiation, and S is the incident solar radiation (W/m^2).

The reverse saturation current I_{rs} varies with temperature according to the following equation:

$$I_{rs} = I_{rr}\left(\frac{T}{T_r}\right)^3 e^{\left(E_g / K'A\right)\left(\left(1/T_r\right)-\left(1/T\right)\right)} \quad (3)$$

Where T_r is the cell reference temperature, I_{rr} the reverse saturation current at T_r, and k' is the Boltzmann's constant in eV/K and E_{go} is the band gap energy of the semiconductor used in the cell.

Finally, Eq. (4) was used in the computer simulations to obtain the open circuit voltage of the PV array:

$$V_{oc} = \frac{AkT}{q}\ln\left(\frac{I_{ph} + I_{rs}}{I_{rs}}\right) \quad (4)$$

From Eqs. (2) to (4):

$$I_{rr} = \frac{\left(I_{scr} + k_i\left(T - T_r\right)\right)\frac{S}{S_t}}{e^{V_{oc}q/AkT} - 1}\left(\frac{T_r}{T}\right)^3 e^{-\left(E_g / K'A\right)\left(\left(1/T_r\right)-\left(1/T\right)\right)} \quad (5)$$

And from Eq. (1):

$$R_{sh} = \frac{V_{oc}}{-I_{rs}\left(e^{qV_{oc}/kTA} - 1\right)} \quad (6)$$

Eqs (1) to (6) are used in the development of computer simulations for the solar array. The MATLAB programming language is used.

2.2. Battery Bank Model

At any hour the state of battery is related to the previous state of charge and to the energy production and consumption situation of the system during the time from (t-1) to t. During the charging process, when the total output of PV and wind generators is greater than the load demand, the available battery bank capacity at hour t can be described by [9, 10]:

$$C_{bat}(t) = C_{bat}(t-1).(1-\sigma) + \left(E_{PV}(t) - \frac{E_L(t)}{\eta_{inv}}\right)\eta_{bat} \quad (7)$$

On the other hand, when the load demand is greater than the available energy generated, the battery bank is in discharging state. Therefore, the available battery bank capacity at hour (t) can be expressed as:

$$C_{bat}(t) = C_{bat}(t-1).(1-\sigma) + \left(\frac{E_L(t)}{\eta_{inv}} - E_{PV}(t)\right) \quad (8)$$

Where $C_{bat}(t)$ and $C_{bat}(t-1)$ are the available battery bank capacity (Wh) at hour (t) and (t-1), respectively; η_{bat} is the battery efficiency (during discharging process, the battery discharging efficiency was set equal to 1 and during charging, the efficiency is 0.65 to 0.85 depending on the charging current).

σ is self-discharge rate of the battery bank. The manufacturer documentation gives a self discharge of 25 % over six months for a storage temperature of 20℃ [11].

$E_{PV}(t)$ is the energy generated by PV; $E_L(t)$ is the load demand at hour t and η_{inv} is the inverter efficiency.

2.3. The Inverter Model

The PV arrays and batteries produce DC and therefore when the stand alone PV system contains an AC load, a DC/AC conversion is required. An inverter is a converter where

the power flows from DC to AC side, i.e., having a DC voltage as input; it produces AC voltage, as output. The inverter is characterized by a power dependent efficiency. The role of the inverter is to keep the voltage constant on the AC side and to convert the input power P_{in} into the output power P_{out} with the best possible efficiency [11]. The inverter is characterized by a power-dependent efficiency η_{in} given by [11]:

$$\eta_{in} = \frac{P_{out}}{P_{in}} = \frac{V_{ac} I_{ac} \cos(\varphi)}{V_{dc} I_{dc}} \quad (9)$$

where I_{dc} is the current required by the inverter from the DC source in order to be able to keep the rated voltage on the AC side, V_{dc} is the input voltage to the inverter delivered by the DC source, V_{ac} and I_{ac} are the output voltage and current, respectively.

3. Field Programmable Gate Arrays

In 1985, Xilinx introduced a completely new idea: combine the user control and time to market of PLDs with the densities and cost benefits of gate arrays. Customers liked it, and the FPGA was born. Today Xilinx is the number one FPGA vendor in the world [12, 2]. A more advanced programmable logic than the CPLD is the Field Programmable Gate Array (FPGA). An FPGA is more flexible than CPLD, allows more complex logic implementations, and can be used for implementation of digital circuits that use equivalent of several Million logic gates [13, 14].

An FPGA is like a CPLD except that its logic blocks that are linked by wiring channels are much smaller than those of a CPLD and there are far more such logic blocks than there are in a CPLD. FPGA logic blocks consist of smaller logic elements. A logic element has only one flip-flop that is individually configured and controlled. Logic complexity of a logic element is only about 10 to 20 equivalent gates. A further enhancement in the structure of FPGAs is the addition of memory blocks that can be configured as a general purpose RAM. Figure 2 shows the general structure of an FPGA. It is an array of many logic blocks that are linked by horizontal and vertical wiring channels. FPGA RAM blocks can also be used for logic implementation or they can be configured to form memories of various word sizes and address space. Linking of logic blocks with the I/O cells and with the memories are done through wiring channels. Within logic blocks, smaller logic elements are linked by local wires.

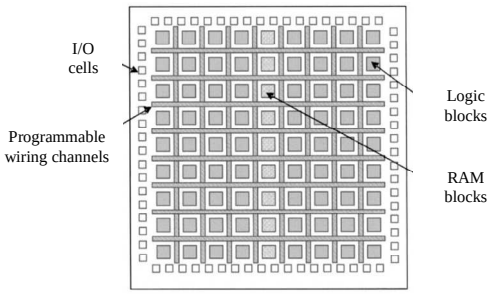

Figure 2. FPGA general structure

As shown in Figure 2, an FPGA is an array of many logic blocks that are linked by horizontal and vertical wiring channels. FPGA RAM blocks can also be used for logic implementation or they can be configured to form memories of various word sizes and address space. Linking of logic blocks with the I/O cells and with the memories are done through wiring channels. Within logic blocks, smaller logic elements are linked by local wires.

FPGAs from different manufacturers vary in routing mechanisms, logic blocks, memories and I/O pin capabilities. An FPGA is under your complete control. This means that you can design, program, and make changes to your circuit whenever you wish. There are two basic types of FPGAs: SRAM-based reprogrammable and OTP (One Time Programmable).

These two types of FPGAs differ in the implementation of the logic cell and the mechanism used to make connections in the device [12].

4. Proposed Neural Network Controller

Stand alone PV system for earth station is controlled using neural network. Figure 4 indicates the proposed power system controller for receiving earth station. As depicted in Figure 3, NN controller controls the battery charge current. The inputs for the controller are the error signal, and the ambient temperature.

The structure of the Multi Layer Perceptron (MLP) proposed for controlling the charge current of battery bank subsystem is shown in Figure 4. Different tests have been done in order to choose the number of neurons and the actual number selected produced the best results. It consists of three layers. The first one or input layer has two inputs as follows:

- error: error signal between the generated and load current.
- T_{air}: ambient temperature.

The second layer, also called the hidden layer, has two neurons or nodes. Finally, the last layer is called the output layer, has only one node. It represents the value of change in battery charge current. The training is done by the Levenberg-Marquart back propagation algorithm.

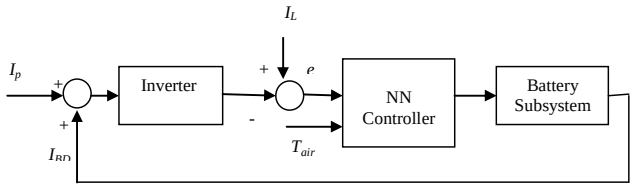

Figure 3. Block diagram of proposed NN controller for earth station power system

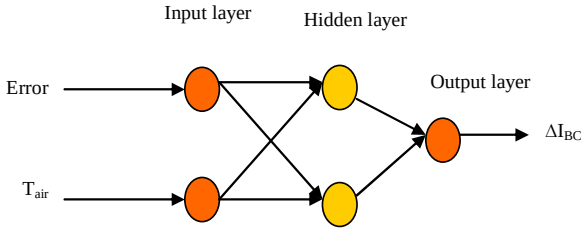

Figure 4. The architecture of the NN controller for earth station power system

5. VHDL Implementation of NNC

By the EDA tools such as FPGA Advantage® 8.1 [15], the design description can be created with one or more of the following five methods; the block diagram, state machine diagrams, flow charts, truth tables, VHDL and/or HDL code. The mentioned types of graphical descriptions are automatically converted by the tool to fast and efficient VHDL description. The hierarchal design capability of the EDA tools simplifies the design task.

At the top level of the hierarchy, a global design can be made in the form of system entity block that appears as block connected with all the inputs and all outputs of our design with its declaration and number of bits.

The second level represents the system components, in the form of block diagram, as illustrated in Figure 5 which appears as 2–2–1 multilayer feed–forward neural network design; the first two blocks show the hidden neurons, the final output of each neuron is determined based on the activation function. The hidden layer uses pure linear function as activation function. The output of these three blocks is connected to the final block which represents the output layer to get the control signal. Also the output layer uses pure linear function as activation function.

Neuron components are composed of three parts, multiplications, summation, and output calculation containing activation function as shown in Figure 6.each neuron in the neural network controller will perform the previous operations to inputs and weights.

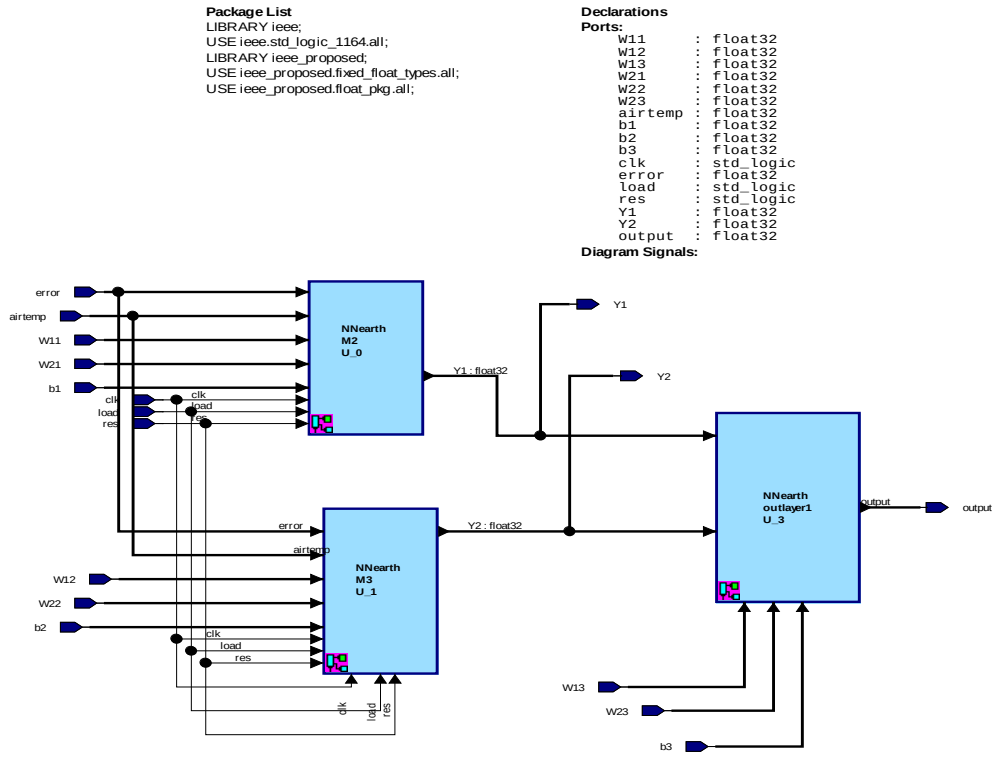

Figure 5. System components block diagram of earth station controller

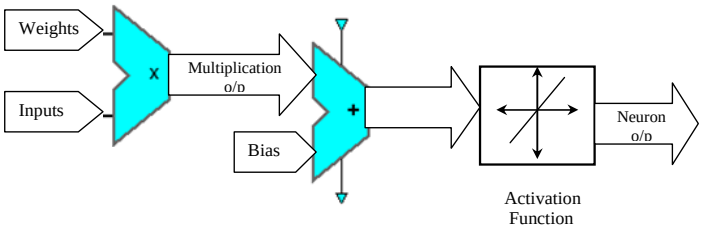

Figure 6. Neuron architecture

6. FPGA Results

Logic simulation in FPGA design environment plays a very vital role in verifying the functionality of the designs. Simulation is a powerful way to test the system on a computer, before it is turned into hardware. Simulators let designer to check the values of signals inside the system. The complete design is simulated using ModelSim® PE 6.6 simulator tool [16].

There are standard packages which required for ModelSim® computation and synthesis. The STD_LOGIC_1164 is used a standard IEEE package in the NNC design which allows to use the data types (STD_LOGIC and STD_LOGIC_VECTOR) in design and basic arithmetic operations. Neurons components besides standard IEEE packages uses float_pkg and fixed_float_pkg packages from IEEE proposed library which allow synthesis of floating and fixed point arithmetic operations. Specification, usage and source code of those packages can be obtained from fixed and floating point VHDL-2008 support library [17].

A test bench is written in VHDL where the set of inputs (error signal & ambient temperature) can be changed for NNC. Also, the inputs like Clock, Reset, and Load signal are

defined and the output is observed in the simulation window. Once all the signals are taken into the waveform window, the simulation is run for 6400 ns and the changes in the signals are observed in the waveform window as shown in Figure 7. The result indicates a good agreement between MATLAB and VHDL.

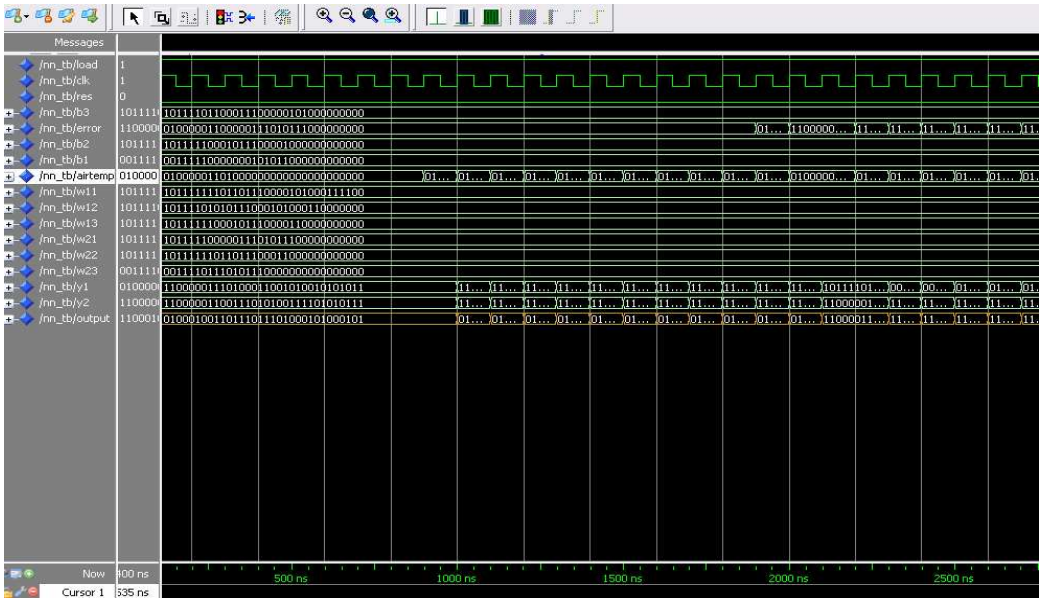

Figure 7. The observed signals are in the waveform window

After results are verified, the next step is to obtain the circuit diagram which will be used in hardware. We have found Xilinx FPGA's is an efficient tool for the design of neural networks. Their reconfigurability and desktop programmability allow design changes at the user's terminal, thereby avoiding the fabrication cycle times and non-recurring engineering charges.

Once the functionality is verified, the VHDL – RTL (Register Transfer Level) code is used for synthesis. At this level, the RTL description can be transformed to a netlist in term of configurable logic blocks (CLB) depending on the target technology transforms. Tile synthesis tool proceeds to estimate area in terms of CLBs. Synthesis is a two step process with an optional third step:

a. Translate synthesizable RTL–HDL (Hardware Description Language) to generic gate level netlist, such as technology independent gates.
b. Optimize and map generic gate level netlist to technology gates utilizing any special architectural features wherever possible. This optimization can be for area and/or speed.
c. Timing optimization if timing constraints not met (optional step).

The output from synthesis is an EDIF netlist ready for vendor place and route tools. In the synthesis phase the target technology must be determined. Two devices for Xilinx FPGA are used (VIRTEXII 2V2000bf957 & VIRTEX5 5VFX100TFF1136). The design summary and the percentage of available resources is explores in Table 1. Xilinx VIRTEX5 is less than VIRTEXII in resource utilization so it is selected in design implementation.

Precision^{TM} RTL Synthesis is a synthesis platform that maximizes the performance of FPGAs. Precision^{TM} RTL Synthesis is a comprehensive tool suite, providing design capture in the form of VHDL, Verilog and System Verilog entry, advanced register- transfer-level logic synthesis, constraint-based optimization, state-of-the-art timing analysis, schematic viewing and encapsulated place-and route [18].

Figure 8 describes the Precision^{TM} RTL schematic of the earth power system NNC using Xilinix FPGA Virtex5 device technology. It consists of basic logic gates (AND, OR, etc.) for a particular fabrication process. These are connected using wires, and due to the size of the final schematic, specific details can only be seen by zooming in a particular part of the design. At this point, it is necessary to consider cell delays due to interconnect and gate loading effects.

Technology schematic using Xilinix FPGA VIRTEX5 device technology which indicates the connections between the lookup tables is depicted in Figure 9.

Table 1. Area Report for A 2-2-1 NN for Xilinx VirteXII, And Virtex5 Device Technology

	VIRTEXII			VIRTEX5		
Resource	Available	Used	Utilization %	Available	Used	Utilization %
IOS	624	450	72.12	640	450	70.31
Global Buffers	16	1	6.25	32	1	3.13
Function Generators	21504	13921	64.74	64000	10335	16.15
CLB Slices	10752	6961	64.74	16000	2584	16.15
Dffs or Latches	23376	64	0.27	65280	64	0.10
Block RAMs	56	0	0.00	228	0	0.00
Block Multipliers	56	28	50.00	–	–	–
Block Multiplier Dffs	2016	0	0.00	–	–	–
DSP48Es	–	–	–	256	14	5.47

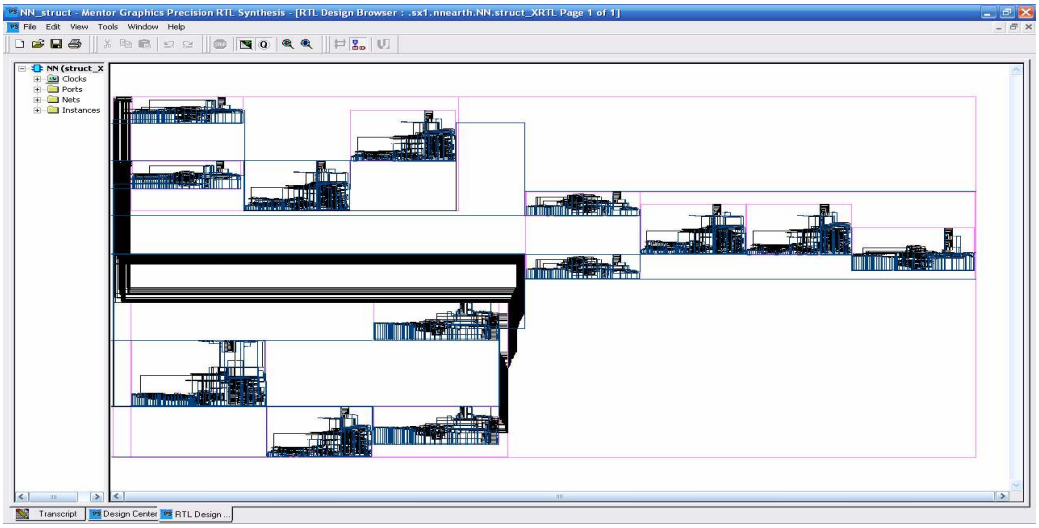

Figure 8. The RTL view of the synthesized architecture of earth station NNC using XILINX VIRTEX5

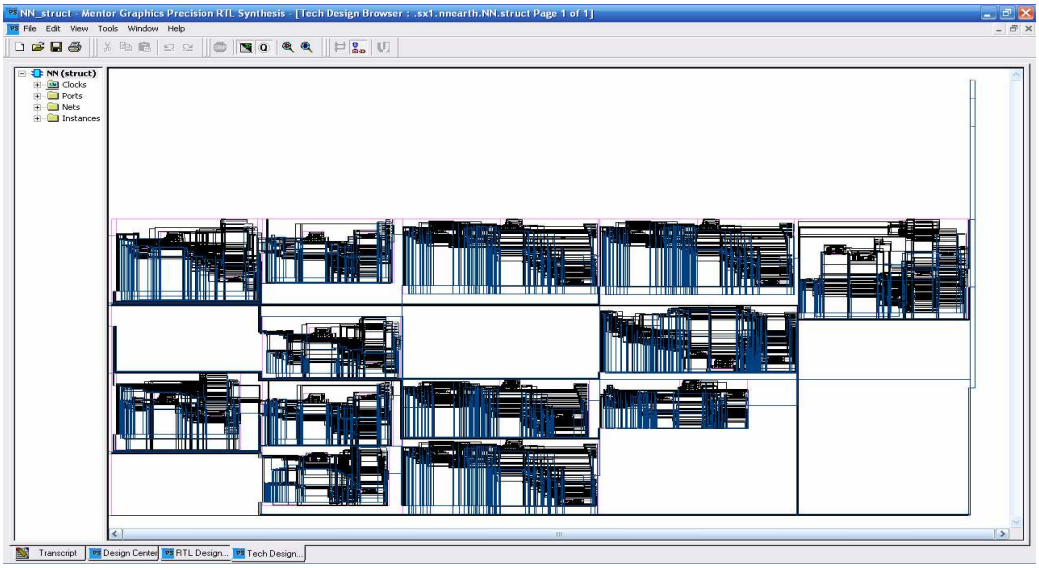

Figure 9. Technology schematic of earth station NNC using XILINX VIRTEX5

7. Conclusion

In this paper, implementation of NNC using hardware description language VHDL is performed. The first reason for using VHDL is to increase the productivity by making it easy to build and use libraries of commonly used VHDL modules. Another important reason is the rapid pace of development in electronic design automation (EDA) tools and in target technologies. Using a standard language such as VHDL can greatly improve your chances of moving into more advanced tools without having to re-enter your circuit descriptions. The ability to retarget circuits to new types of device targets will also be improved by using a standard design entry method. This implemented digital controller was designed to be simple so as to be fast and not require large hardware resources. The integration between MATLAB and VHDL is achieved. The network of digital neurons can then be synthesized into digital hardware using a commercial synthesis tool.

ModelSim® PE 6.6 simulator tool is used to simulate the NNC for both power systems. A comparison between MATLAB–SIMULINK and VHDL has been occurred. The results give a good agreement between VHDL and MATLAB software. The VHDL-FPGA combination is shown to be a very powerful embedded system design tool, with low cost, high reliability. Power system controller is suggested which is designed, modeled, and simulated using FPGA Advantage® 8.1. We have found Xilinx FPGA's is an efficient tool for the design of neural networks. Their reconfigurability and desktop programmability allow design changes at the user's terminal, thereby avoiding the fabrication cycle times and non-recurring engineering charges. Several XILINX technologies are compared with respect to the utilization of the resources. It is found that XILINX VIRTEX5 has the lowest utilized resources than others. The hardware architecture of neural network with two input, one output and two hidden neurons occupies only 16.15% of available CLB slices using XILINIX VIRTEX5 target technology.

References

[1] Kalogirou SA. *Artificial intelligence in energy and renewable energy systems*. Nova Publisher; 2007.
[2] Steven T. Karris. *Digital Circuit Analysis and Design with Simulink® Modeling and Introduction to CPLDs and FPGAs*. Second Edition, Orchard Publications.
[3] Zainalabedin Navabi, *Digital Design and Implementation with Field Programmable Devices*, Kluwer Academic Publishers, Boston, 2005.
[4] Janardan Misra , IndranilSaha , "Artificial neural networks in hardware: A survey of two decades of progress", *Journal of Neurocomputing*, 2010; 74: 239–255.
[5] Alexander Gomperts, Abhisek Ukil, and Franz Zurfluh, "Development and Implementation of Parameterized FPGA-Based General Purpose Neural Networks for Online Applications", *IEEE Transactions on Industrial Informatics*, 2010.
[6] W. X. Shen. Optimally sizing of solar array and battery in a standalone photovoltaic system in Malaysia. *Renewable Energy*. 2009; 34: 348-352.
[7] C. Hua, and C. Shen. Study of maximum power tracking techniques and control of DC/DC converters for photovoltaic power system. *Proceedings of the 29th Annual IEEE Power Electronics Specialists Conference*, 1998.
[8] G.J Yu, Y.S Jung, J.Y Choi, and G.S Kim. A Novel Two-Mode MPPT Control Algorithm Based on Comparative Study of Existing Algorithms. *Solar Energy*. 2004; 76: 455–463.
[9] Bogdan, S. B. and Salameh, Z. M. 1996. Methodology for optimally sizing the combination of a battery bank and PV array in a wind/PV hybrid system. *IEEE Transactions on Energy Conversion*. 1996; 11(2): 367-375.
[10] Bin A., Hongxing Y., Shen H., Xianbo L. Computer aided design for PV/Wind hybrid system. *Renewable Energy*. 2003; 28(10): 1491-1512.
[11] Hansen D., Sorensen P., Hansen H., and Henrik B., Models for a Stand-Alone PV System. Rio-R-1219 (EN)/SEC-R-12, 2000.
[12] Programmable Logic Design Quick Start Handbook, Xilinx, Inc., 2006.
[13] A. Mellit, H. Rezzouk, A. Messai, B. Medjahed. FPGA-based real time implementation of MPPT-controller for photovoltaic Systems. *Renewable Energy*. 2011; 36: 1652-1661.
[14] Mellit A, Mekki H, Messai A, Salhi H., FPGA-based implementation of an intelligent simulator for stand-alone photovoltaic system. *Expert Systems with Applications*, 2010; 37(8): 6036-6051.
[15] FPGA Advantage®, Software Version 8.1, Mentor Graphics Corporation 1999-2008.
[16] ModelSim® PE Reference Manual, Software Version 6.6, Mentor Graphics Corporation 1991-2010.
[17] David W. Bishop, VHDL-2008 Support Library, published by EDA Industry Working Groups and Accellera, http://www.vhdl.org/fphdl/, online 2010.

[18] Precision® RTL Synthesis User's Manual, Release 2008a update 1, September 2008, Mentor Graphics Corporation, 2008.