World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:5, No:9, 2011

Spacecraft Neural Network Control System Design using FPGA

Hanaa T. El-Madany, Faten H. Fahmy, Ninet M. A. El-Rahman, and Hassen T. Dorrah

***Abstract*—**Designing and implementing intelligent systems has become a crucial factor for the innovation and development of better products of space technologies. A neural network is a parallel system, capable of resolving paradigms that linear computing cannot. 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 VLSI and DSP chips. So, many researchers have made great efforts on the realization of neural network (NN) using FPGA technique. In this paper, an introduction of ANN and FPGA technique are briefly shown. Also, Hardware Description Language (VHDL) code has been proposed to implement ANNs as well as to present simulation results with floating point arithmetic. Synthesis results for ANN controller are developed using Precision RTL. Proposed VHDL implementation creates a flexible, fast method and high degree of parallelism for implementing ANN. The implementation of multi-layer NN using lookup table LUT reduces the resource utilization for implementation and time for execution.

***Keywords*—**Spacecraft; Neural network; FPGA; VHDL

I. Introduction

ARTIFICIAL Neural Network (ANN) has promising applications in science and engineering. The main advantages of using ANN algorithms are simplifying the complicated algorithms, reducing heavy computation demands and improving fault tolerance. The ANN is particularly useful to implement nonlinear, time-varying input-output mapping [1]. Analog implementations have the potential for high densities and fast operations. Unfortunately, they are sensitive to noise; cross talk, temperature effects and power supply variations. Also long term weight storage requires special fabrication techniques. Another major drawback, which is very critical in ANNs is that conventional analog implementations are fixed (i.e. no programmability can be achieved). Digital integrated technology, in the other hand, offers very desirable features such as design flexibility, learning, expandable size and precision. Another advantage is that mature and powerful CAD tools support design of digital VLSI circuits.

Digital implementation of ANNs can make use of full custom VLSI, semi custom, ASICs (application specific integrated circuits) and FPGAs. Particularly, FPGA implementation of ANNs is very attractive because of the high flexibility that can be achieved through the re-programmability nature of these circuits [2, 3]. Also, FPGA is concurrent, which supports the massively parallel calculation of neural network.

Nowadays, with the increasing complexity of VLSI circuits, state of the art design is focused around high level synthesis which is a top down design methodology, that transform an abstract level such as the VHDL language into a physical implementation level.

VHDL based synthesis tools have become very popular due to mainly these reasons: the need to get a correctly working system at first time, technology independent design, design reusability, the ability to experiment with several alternatives of the design, and economic factors such as time to market. In addition, synthesis tools allow designers with limited knowledge, of low level implementation details to analyze and trade off between alternative implementations without actually implementing the target architecture [4]. Beside this, the VHDL language is well suited for high regular structures like neural networks. However, although all these advantages, seldom attention has been done to use synthesis for ANNs implementations.

The paper is organized as follow: Spacecraft power system architecture is given in section II. Section III reviews the basics of FPGA. Also, the theoretical background of artificial neural networks is given in section IV. Proposed spacecraft power control system architecture and data representation are introduced in sections and V and VI respectively. Section VII describes VHDL-implementation of NNC. Simulation and synthesis results are introduced in section VIII. Finally, conclusion is given in section IX.

II. Spacecraft Power System Architecture

Photovoltaic conversion of the sun's energy is the most common source of electrical power in space. An array of photovoltaic cells powers the load and charges a battery during sunlight. The battery powers the load during an eclipse. A typical solar panel–battery power system is shown in Fig. 1 [5, 6].

H. T. El-Madany, F. H. Fahmy, and N. M. A. El-Rahman are with Electronics Research Institute, National Research Center Building, Cairo, Egypt (phone: 202-33310512; fax: 202-33351631; e-mail: hanaa_tolba@yahoo.com).

H. T. Dorrah, is Electrical Power & Machines Dept., Cairo University, Egypt (e-mail: dorrahht@AOL.com)

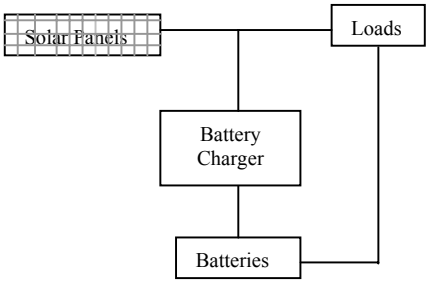

Fig. 1 Typical solar panel– battery system architecture

Using the equivalent circuit of a solar cell, the non-linear *I–V* characteristics of a solar array are extracted, neglecting the series resistance [5]:

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

Where: I_0 is the PV array output current (A), V_0 is the PV array output voltage (V), q is the charge of an electron, k is the Boltzmann's constant in J/K, A the p–n junction ideality factor, T is the cell temperature (K), and I_{rs} is the cell reverse saturation current (A).

The photocurrent I_{ph} depends on the solar radiation and the cell temperature as described in the following equation [5]:

$$I_{ph} = (I_{scr} + k_i(T - T_r))\frac{s}{1353} \quad (2)$$

Where: I_{scr} is the PV array short circuit current at reference temperature and radiation (A), T_r is the cell reference temperature, k_i the short circuit current temperature coefficient (A/K) and S is the solar radiation (W/m^2).

A generic model to most popular types of rechargeable batteries is represented as follows [5]:

$$E = E_o - K\left(\frac{Q}{Q - \int idt}\right) + C\exp\left(-D\int idt\right) \quad (3)$$

Where: E is no load voltage (V), E_0 is constant voltage (V), K is polarization voltage (V), Q is battery capacity (Ah), C is exponential voltage (V), and D is exponential capacity (Ah^{-1}). The state of charge (SOC) of the battery can be calculated as:

$$SOC = 100\left(1 - \frac{Q * 1.05}{\int idt}\right) \quad (4)$$

III. Field Programmable Gate Arrays

FPGAs are a form of programmable logic, which offer flexibility in design like software, but with performance speeds closer to Application Specific Integrated Circuits (ASICs) [6]. With the ability to be reconfigured an endless number of times after having been manufactured so FPGAs have traditionally been used as a prototyping tool for hardware designers [7].

A more advanced programmable logic than the CPLD is the 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 [8].

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 [9, 10].

As shown in Fig. 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 [11, 12].

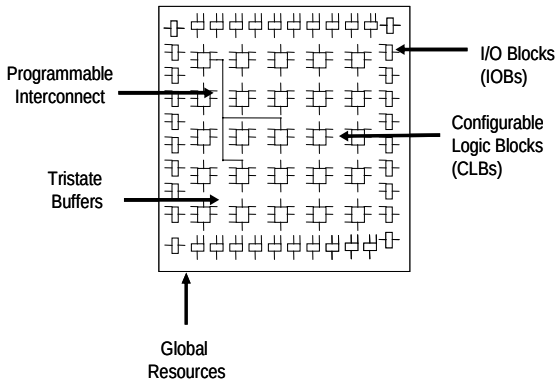

Fig. 2 Generic FPGA architecture.

IV. Artificial Neural Network

Artificial intelligence (AI) techniques are becoming useful as alternate approaches to conventional techniques or as components of integrated systems. They have been used to solve complicated practical problems in various areas and are becoming more and more popular nowadays. Today, considerable attention has been focused on use of ANN on system modeling and control applications [1, 13].

The basic processing elements of neural networks are called artificial neurons, or simply neurons or nodes. As indicated in Fig. 3, the effects of the synapses are represented by connection weights that modulate the effect of the associated input signals, and the nonlinear characteristic exhibited by neurons is represented by a transfer function. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm. The learning situations in neural networks may be classified into three distinct sorts. These are supervised learning, unsupervised learning, and reinforcement learning.The total synaptic input, u, to the neuron is given by the inner product of the input and weight vectors:

$$u = \sum_{i=1}^{I} w_i x_i \tag{5}$$

The output activation, y, is given by:

$$y = \phi(u) \tag{6}$$

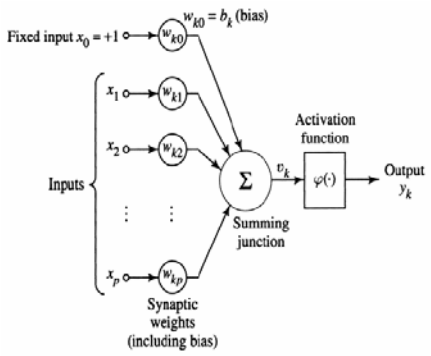

Fig. 3 Nonlinear model of a neuron.

The main advantages of the neural network technique are

- Nonlinearity.
- Mapping input signals to desired response.
- Adaptivity.
- Evidential response: confidence level improves classification.
- Contextual information: Knowledge is represented by the very structure and activation.
- Fault tolerent: graceful degradation of performance if damaged.
- Uniformity of analysis and design.
- Neurobiological analogy.

V. Proposed Spacecraft Power Control System Architecture

Fig. 4 shows the block diagram of the control subsystem using NNC. In this diagram, the NNC controls whether the system is in peak power or in eclipse conditions. Comparing the solar array current with the load current, the change in battery charge current is considered as the difference between them.

Fig. 5 indicates the proposed multi-layer perceptron network architecture. The inputs of this controller are the load current (I_L) and the error signal (E) while the output is the change in battery charge current (ΔI_{BC}). The input and the output are fixed initially however the number of hidden layers and the neurons within these layers are optimized during the learning process based on the good performance of root mean square error (RMSE). A two layer feed-forward network with "purelinear" hidden neurons and "purlinear" output neurons are be used. The network will be trained using MATLAB–SIMULINK.

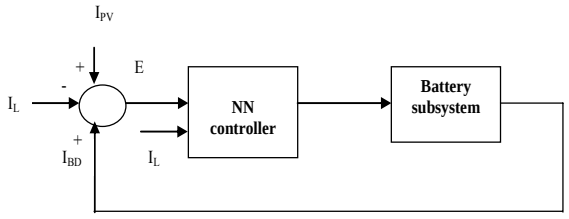

Fig. 4 Block diagram of NN controller

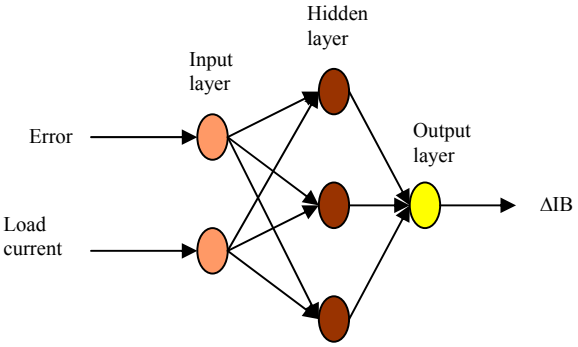

Fig. 5 The architecture of the NN controller model

VI. Data Representation

There are two problems during the hardware implementation of ANNs. How to balance between the need of reasonable precision (number of bit), that is important for ANN and the cost of more logic area associated with increased precision. How to choose a suitable number format that dynamic range is large enough to guarantee that saturation will not occur for a general-purpose application. So, before beginning ANN's based FPGAs system design with VHDL, number format (floating point, fixed point etc.) and

precision which used for inputs, weights and activation function must be considered. Floating point offers the greatest amount of dynamic range, making it suitable for any application so it would be the ideal number format to use.

So ANN's architecture was developed using VHDL with 32 bit floating point arithmetic. Unfortunately, there is currently no clear support for floating-point arithmetic in VHDL. As a result, a VHDL library was designed for using ANN's on FPGAs. The library supports to the IEEE-754 standards for single-precision (32-bit) floating point arithmetic, and it is referred to, fixed_pkg.vhdl, float_pkg.vhdl, and fixed_float_types.vhdl packages [14].

The single precision floating point numeric representation supports to IEEE-754 standard is shown in Fig. 6.

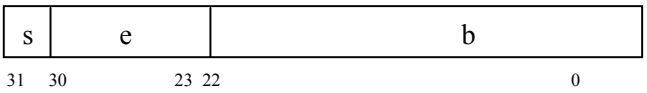

Fig. 6 32 bit Floating Point Format

The floating point number (n) is computed by:

$$n = -1^{s} 2^{|e-127|} (1.b) \quad (7)$$

In Fig. 6, sign field is referred to 's' is bit 31 and is used to specify the sign of the number. Exponent field is referred to 'e' which ranges from bits 30 down to 23. The bias of 127 is used because of 8 bit quantity is a signed number representation. The bits 22 down to 0 are used to store binary representation (b) of floating point number.

VII. VHDL-Implementation of NNC

VHDL is an industry standard language used within the design of digital circuits and systems. Toolsets based on the language allow the designer to model, simulate and ultimately synthesis into hardware logic complex digital designs commonly encountered in modern electronic devices. A number of benefits are derived from using VHDL as a neural network representational framework. First VHDL provides a powerful language suitable for modeling, simulation, and behavioral representation. VHDL also provides an interface to powerful circuit simulators used to predict responses and convergence of circuits and the neural networks they represent. The designer can embody both circuits and mathematical model constraints in the same VHDL representation. In this way, VHDL neural network design simulations represent not only the simulation of the mathematical model itself, but also the actual behavior of the intended device [15].The first level in the design of the spacecraft power system controller is the top-level block diagram using FPGA as indicated in Fig. 7. The previous figure describes a 2–3–1 multilayer feed forward neural network. The input layer neurons pass the input signal to the hidden layer based on multiplexing.The neurons in the hidden and output layer perform the computation. Each neuron in the hidden and output layers calculates its net output by determining the product of input and weight of each connection. The final output of each neuron is determined based on the activation function. The hidden and output layers use pure linear function as activation function.

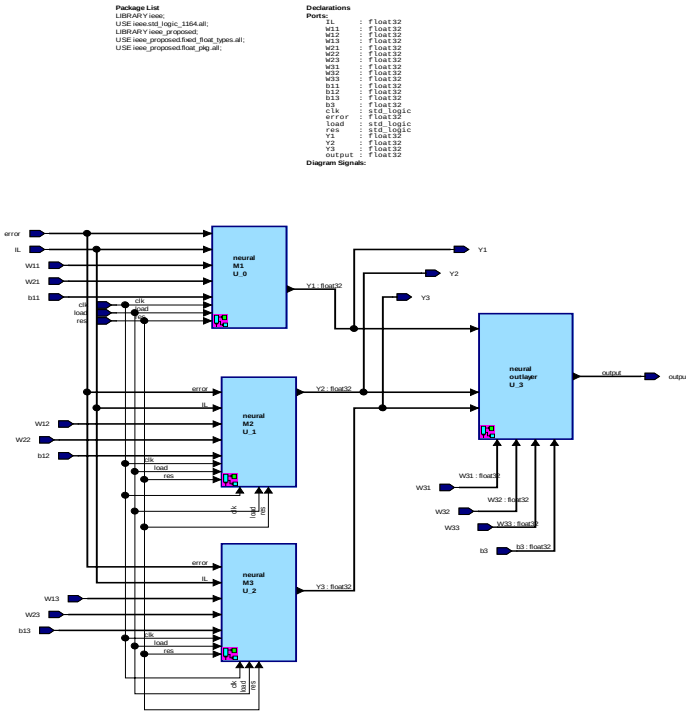

Fig. 7 Hardware implementation of spacecraft power system controller using NN

VIII. Simulation and Synthesis Results

The ANN computation can be divided in two phases: learning phase and recall phase. The learning phase performs an iterative updating of the synaptic weights based upon the error back-propagation algorithm. It teaches the ANN to produce the desired output for a set of input patterns. The recall phase computes the activation values of the neurons from the output layer according to the weighted values (computed in the learning phase).In our control subsystem, the recall phase of a neural network is implemented which has been previously trained on MATLAB–SIMULINK where the final synaptic weights are obtained, i.e. "off- chip training".After the design description is created using FPGA Advantage® 8.1, the second procedure is to simulate the total design using Modelsim.A design process is incomplete without design verification. There are several ways to verify a VHDL design. However the most popular way is to use a test bench. A test bench is an environment where a design (called a design or unit under test, UUT) is checked by applying

stimulus to its inputs and monitoring the output responses. Test bench is described using VHDL code. Figure 8 represents test bench Modelsim results of the ANN. The final results of the designed NNC can be obtained during 100 ns. This value is not comparable with the microcontroller or the DSP speed of the algorithm execution. This is due to the very long time of sequential execution of the NNC algorithm on this type of devices. Output results show that, the required functionality is well achieved. It is indicated that, hardware implementation will result in very fast digital neurons compared to a biological system. Where biological neurons respond in milliseconds, our digital neurons will respond in tens of nano-seconds using a modest 100 MHz clock. This is a speed enhancement of at least a factor of 10^4, and a large-scale neuron network such as the human brain operating at these speeds is an interesting conjecture.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.

1. Translate synthesizable RTL–HDL (Hardware Description Language) to generic gate level netlist, such as technology independent gates.
2. 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.
3. 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.The synthesis results are taken with the help of XILINX SPARTAN3 (device 3S4000fg900), VIRTEXII (device 2V2000bf957), and VIRTEX5 (device 5VFX100TFF1136) device technologies which contain 27648, 10752, and 16000 CLBs respectively as depicted in Table I. Synthesis results indicate that XILINX VIRTEX5 has the lowest utilization percentage of resources than the other device technologies. So the target technology used to implement our design is the XILINX FPGA VIRTEX5 (device 5VFX100TFF1136). Precision™ RTL Synthesis is a synthesis platform that maximizes the performance of FPGAs. Precision™ RTL Synthesis is a comprehensive tool suite, providing design capture in the form of VHDL, Verilog and SystemVerilog entry, advanced register- transfer-level logic synthesis, constraint-based optimization, state-of-the-art timing analysis, schematic viewing and encapsulated place- and route.Figure 9 describes the RTL schematic of the spacecraft power system NNC using Xilinix FPGA Virtex5 device technology. The previous figure 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 Fig. 10.

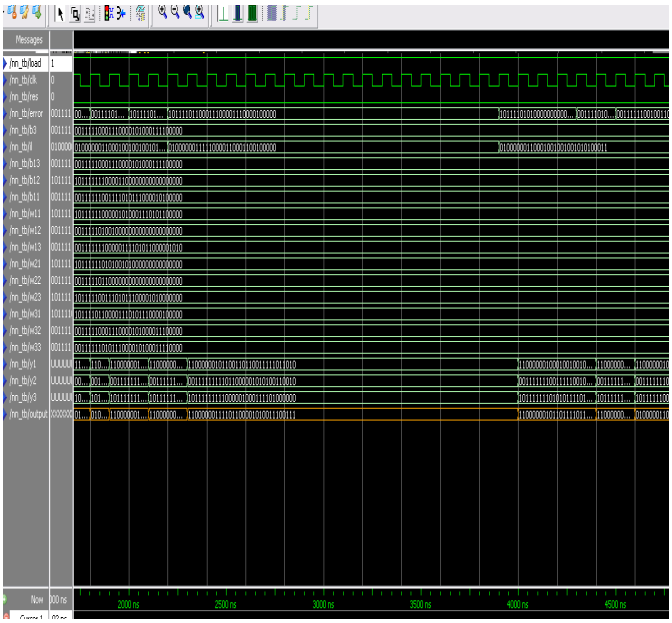

Fig. 8 Simulation results of NNC for implementation on FPGA

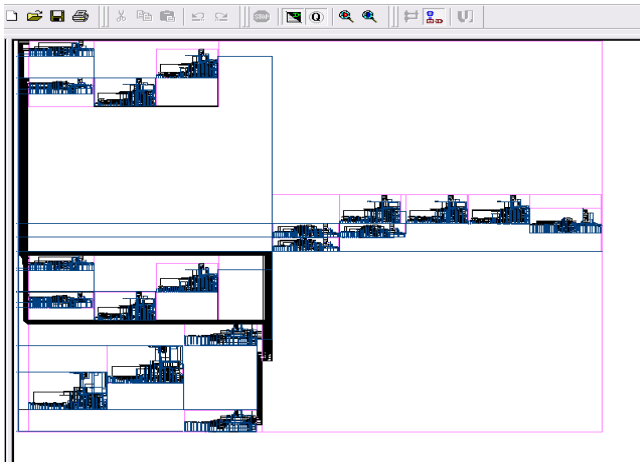

Fig. 9 The RTL view of the synthesized architecture of NNC using XILINX VIRTEX5

TABLE I
AREA REPORT FOR A 2-3-1 NN FOR XILINX SPARTAN3, VIRTEXII, AND VIRTEX5 DEVICE

Resource	SPARTAN3			VIRTEXII			VIRTEX5		
	Available	Used	Utilization %	Available	Used	Utilization %	Available	Used	Utilization %
IOS	633	610	96.37	624	610	97.76	640	610	95.31
Global Buffers	8	1	12.50	16	1	6.25	32	1	3.13
Function Generators	55296	25390	45.92	21504	20059	93.28	64000	15066	23.54
CLB Slices	27648	12695	45.92	10752	10030	93.28	16000	3767	23.54
Dffs or Latches	57195	96	0.17	23376	96	0.41	65280	96	0.15
Block RAMs	96	0	0.00	56	0	0.00	228	0	0.00
Block Multipliers	96	12	12.50	56	42	75.00	–	–	–
Block Multiplier Dffs	3456	0	0.00	2016	0	0.00	–	–	–
DSP48Es	–	–	–	–	–	–	256	21	8.20

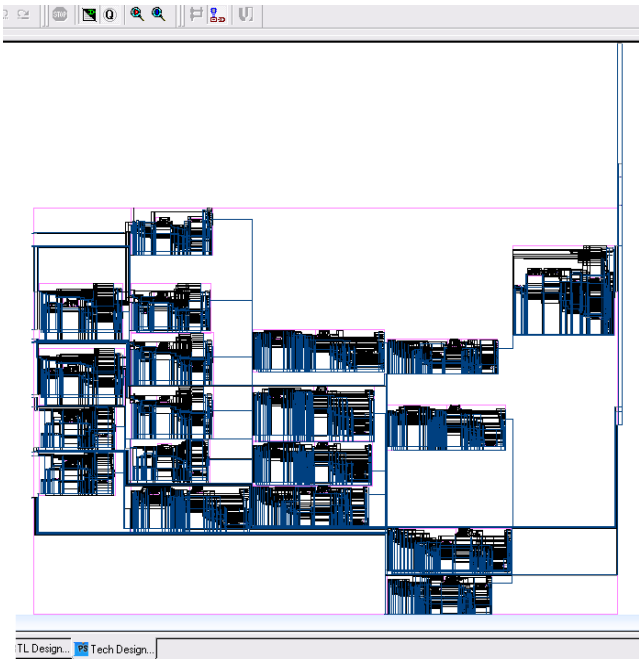

Fig. 10 Technology schematic of NNC using XILINX VIRTEX5

IX. Conclusion

This paper presents a spacecraft power system controller which is designed, modeled, and simulated using FPGA Advantage [R] 8.1. Synaptic weights can be obtained from MATLAB – SIMULINK which save computation time. A fast and flexible feed forward neural network can be obtained which is capable of dealing with Floating point arithmetic operations using VHDL programming language. The final results of the designed NNC can be obtained in very low simulation time as compared to microcontroller and DSP.Model was synthesized into Precision RTL. Three technologies (XILINX SPARTAN3, VIRTEXII, and VIRTEX5) 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 three hidden neurons occupies only 23.54 % of available CLB slices using XILINIX VIRTEX5 target technology.

References

[1] Amos R. Omondi, and Jagath C. Rajapakse, "FPGA Implementations of Neural Networks", Springer publisher, 2006.
[2] Zainalabedin Navabi, "Digital Design and Implementation with Field Programmable Devices", Kluwer Academic Publishers, New York, 2005.
[3] Mehmet Tukel And Mustak Erhan Yalcin, "A New Architecture for Cellular Neural Network on Reconfigurable Hardware with an Advance Memory Allocation Method, in proc. of IEEE 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA), 2010
[4] Douglas L. Perry , "VHDL: Programming by Example", The McGraw-Hill Companies, Inc. , 2002.
[5] Wiley J. Larson, and James R. Wertz, Spacecraft Mission Analysis and Design, Micrcosm Press, Elo, Segrund, California, 2008.
[6] Mukund R. Patel, Spacecraft Power Systems, CRC Press, Boca Raton, Florida, 2005.
[7] 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.
[8] A. Mellit, H. Rezzouk, A. Messai, B. Medjahed, "FPGA-based real time implementation of MPPT-controller for photovoltaic Systems", Renewable Energy Vol. 36, PP. 1652-1661, 2011.
[9] Mellit A, Mekki H, Messai A, Salhi H., "FPGA-based implementation of an intelligent simulator for stand-alone photovoltaic system. Expert Systems with Applications", Vol. 37, No.8, PP. 6036-6051, 2010.
[10] Deming Chen, Jason Cong, and Peichan Pan, "FPGA Design Automation: A Survey",now Publishers Inc. , The Netherlands, 2006.
[11] Ian Grout, "Digital Systems Design with FPGAs and CPLDs", Elsevier (Newnes press) publications, 2008.
[12] Nasri Sulaiman, Zeyad Assi Obaid, M. H. Marhaban and M. N. Hamidon, "Design and Implementation of FPGA-Based Systems - A Review", Australian Journal of Basic and Applied Sciences, Vol. 3, No. 4, PP. 3575-3596, 2009.
[13] Janardan Misra , IndranilSaha , "Artificial neural networks in hardware: A survey of two decades of progress", Journal of Neurocomputing, Vol. 74, PP. 239–255, 2010.
[14] David W. Bishop, VHDL-2008 Support Library, published by EDA Industry Working Groups and Accellera, http://www.vhdl.org/fphdl/, online 2010.
[15] H. Mekki, A. Mellit, H.Salhi, and B. Khaled, "Modeling and simulation of photovoltaic panel based on artificial neural networks and VHDL-language", IEEE, 2007.

Hanaa T. El-Madany obtained her B.Sc. degree in Electronics and Communication Engineering from Faculty of Electronic Engineering, Menofia University in 2003. She completed her MSc degree in communication engineering from Faculty of Electronic Engineering, Menofia University in 2007. She is an assistant Researcher in the Electronic Research Institute, Photovoltaic Cells Department. She published many papers in spacecraft power systems. Her interests are in renewable energy sources systems, especially photovoltaic systems and their applications.