FAULT DIAGNOSIS ON VLSI ADDER CIRCUITS USING ARTIFICIAL NEURAL NETWORK

by

PUI MIN SAN

A Dissertation submitted for partial fulfilment of the requirement for the degree of Master of Science (Electronic Systems Design Engineering)

August 2015

ACKNOWLEDGEMENTS

First and foremost, I would like to take this opportunity to thank my supervisor, Assoc. Prof. Dr. Nor Ashidi Mat Isa, for the guidance, throughout this research. He dedicated his great efforts to constantly provide me advices and feedbacks, which enlightened me greatly throughout this research. Moreover, he handed me his supports and encouragement for my own ideas on this research, helping to make sure this research runs smoothly.

Besides that, I would like to express my gratitude to my manager, Mrs. Lee Jin Yee for providing the opportunity to enrol in this master programme and all the supports given.

Last but not least, I would wish to thank my parents and my family, who have been giving me unconditional love and freedom in my spirit.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND NOMENCLATURE

Abbreviation	Meaning
ANN	Artificial Neural Network
IC	Integrated Circuit
VLSI	Very-Large-Scale Integration
LM	Levenberg-Marquardt
BR	Bayesian Regularisation
SCG	Scaled Conjugate Gradient
MLP	Multilayer Perceptron
FPGA	Field-Programmable Gate Array
HDL	Hardware Description language
R	Regression
RTL	Register-Transfer Level
MSE	Mean Squared Error
ATPG	Automated Test Pattern Generation
VHDL	VHSIC Hardware Description Language

DIAGNOSIS KESALAHAN LITAR PENAMBAH VLSI MENGGUNAKAN RANGKAIAN NEURAL BUATAN

ABSTRAK

Diagnosis kesalahan litar VLSI digital adalah satu teknik untuk mengesan kesalahan dan lokasi kesalahan yang hadir dalam litar VLSI digital. Litar yang rosak dalam IC boleh menyebabkan IC menjadi rosak dan tidak boleh digunakan. Oleh itu, litar yang rosak mesti dikesan semasa proses pembuatan untuk mencegah IC rosak daripada diperolehi pelanggan, yang boleh membawa isu kebolehpercayaan. Diagnosis kesalahan secara manual yang pada masa ini digunakan dalam industri, adalah dijalankan secara analisis yang bergantung kepada manusia. Walau bagaimanapun, teknik ini yang bergantung kepada analisis secara besar-besaran oleh manusia, mempunyai beberapa kelemahan seperti masa diagnosis yang panjang untuk kes yang rumit, analisis secara manual, ketepatan bergantung kepada kesilapan manusia, kerumitan litar yang rosak, kecerdasan manusia dan lain-lain. Oleh itu, dalam kajian ini, dengan pendekatan diagnosis kesalahan baharu yang berasaskan rangkaian neural buatan, adalah dicadangkan untuk mengesan kerosakan dan lokasi kesalahan dengan lebih tepat dan diagnosis yang lebih menjimatkan masa jika dibandingkan dengan kesalahan diagnosis manual. Dalam kajian ini, diagnosis kesalahan yang dicadangkan telah direka berdasarkan rangkaian neural hierarki, yang mengubah setiap litar sepenuh penambah kepada rangkaian neural buatan individu. Diagnosis kesalahan yang dicadangkan mempunyai ketepatan 100% apabila ia digunakan pada 4-bit penambah, 5-bit penambah dan 6-bit penambah dalam masa diagnosis hanya satu saat. Sebaliknya, diagnosis kesalahan manual mungkin mengambil masa sehingga beberapa hari untuk mendiagnos litar penambah. Oleh itu, diagnosis

kesalahan automatik yang dicadangkan mempunyai potensi yang tinggi bagi diaplikasikan untuk litar digital VLSI di industri.

FAULT DIAGNOSIS ON VLSI ADDER CIRCUITS USING ARTIFICIAL NEURAL NETWORK

ABSTRACT

Fault diagnosis on VLSI digital circuit is a technique to detect a fault and the location of the fault that present in a VLSI digital circuit. A faulty circuit in an IC can cause the IC to be malfunctioning and unusable. Therefore, the faulty circuit must be detected during the manufacturing process to prevent the faulty IC from delivering to the customers, which can cause a reliability issue. Manual fault diagnosis, which is currently being used in industries, is carried out by the analysis of the human. However, the manual fault diagnosis, which relies heavily on the massive analysis by the human, has several disadvantages such as long diagnosis time for complicated fault diagnosis case, manual analysis, accuracy dependent to human errors, complexity of the faulty circuits, human intelligence, etc. Therefore, in this study, a new fault diagnosis approach, which is based on artificial neural network, is proposed to detect a fault and the fault location with better accuracy and diagnosis time if compared to the manual fault diagnosis. In this study, the proposed fault diagnosis is designed based on the hierarchical neural network, which transforms each of the full-adder circuits to an individual artificial neural network. The proposed fault diagnosis has 100% accuracy when it is applied to the 4-bit adder, 5-bit adder and 6-bit adder as well as one second diagnosis time. On the other hand, the manual fault diagnosis may take up to a few days to diagnose the adder circuits. Thus, the proposed automated fault diagnosis has high potential to be applied for VLSI digital circuits in industries.

CHAPTER 1

INTRODUCTION

1.1 Research Background

There is a big challenge for Very-Large-Scale Integration (VLSI) integrated circuit (IC) manufacturing industry to increase the manufacturing yield. The manufacturers are always trying their best to decrease as low as possible the number of defective ICs produced, since a single fault in an IC could cause the whole IC to be useless and unsellable. A low manufacturing yield will significantly reduce or cause a loss in the revenue to manufacturers. Hence, the reliability of ICs must be ensured to as high as possible. In order to improve the manufacturing process or circuit design which causes the faults in ICs, failure analysis engineers need to analyse the faulty ICs that fail any production test as well as determine the cause of failures. Detection of a fault and the location of the fault that present in a VLSI digital circuit is known as fault diagnosis on VLSI digital circuits (Abramovici et al., 1990, Gorlich et al., 1987). The detection of a fault is necessary to avoid any problem of the quality issues of IC among customers. Meanwhile, the detection of the location of the fault may be required to identify and then discard or replace the faulty circuit. Moreover, the location of the fault can also be used to analyse the defect causing the faulty circuit, which is important in improving the manufacturing process and yield.

1.2 Current Method of Fault Diagnosis on VLSI Digital Circuits

There is a current fault diagnosis which is used in industries, i.e. the manual fault diagnosis by the human. The manual fault diagnosis consists of a series of massive analysis which is highly dependent on human intelligence. The manual fault diagnosis involves two main steps, i.e. fault isolation and fault localisation. Fault isolation is a technique of detecting a fault and minimising the number of suspects that may cause the fault. Fault localisation is a technique of identifying the location of the fault among the suspects.

1.3 Problem Statement

The testing of VLSI digital circuits has become more and more difficult due to the increase in complexity of VLSI digital circuits. The increase in complexity of VLSI digital circuits is due to a few factors, such as the increase in complexity of the architectural design, the increase in the number of transistors in the VLSI digital circuits, and the decrease in the size of transistors. On the other hand, the test cost becomes more expensive due to the long testing time, expensive test equipment and long debug time.

There is a manual fault diagnosis method currently being used in industries to determine the location of the defect and the root cause of failure in VLSI digital circuits. However, due to the increase in difficulty in the testing, the manual fault diagnosis becomes extremely difficult and time-consuming. In addition, the manual diagnosis result might lose accuracy due to certain factors such as human errors, limit of human intelligence, complexity of circuits, etc.

Overcoming the difficulties in the testing of VLSI digital circuits and the long testing time have become big challenges to failure analysis engineers to perform fault isolation and failure analysis in the VLSI digital circuits.

1.4 Objectives

Based on the above-mentioned problems, the objectives of this study are:

i. To design a fault diagnosis, which detects a fault and the location of the fault in VLSI digital adder circuits by using Artificial Neural Networks (ANN).
ii. To improve fault diagnosis performance in terms of the accuracy and the diagnosis time of fault diagnosis when compared to the manual fault diagnosis.

1.5 Scope of Research

The scope of this research is to design a fault diagnosis, which detects the fault and the location of the fault in VLSI digital adder circuits by using Multilayer Perceptron (MLP) neural network. There is a total of three adder circuits created and tested with the proposed fault diagnosis in this research. The adder circuits are 4-bit, 5-bit and 6-bit adder circuits.

This research will employ three learning algorithms, namely Lavenberg-Marquardt (LM), Bayesian Regularisation (BR) and Scaled Conjugated Gradient (SCG) to train the MLP networks, which are then applied to the proposed fault diagnosis to test the adder circuits. The results of the fault diagnosis with these algorithms are compared to determine a most suitable algorithm out of them as well as the optimum number of hidden neurons that can be applied as a fault diagnosis. In

order to detect a fault as well as the fault location, a fault diagnosis is transformed to hierarchical neural network and applied to the adder circuits.

Adder circuits are chosen for fault diagnosis in this research as they are widely used in the VLSI Integrated Circuits and microprocessors, such as arithmetic logic units, parts to calculate addresses, increment and decrement operators, etc. Thus, the probability of a fault occurring in an adder circuit from VLSI IC or microprocessor is relatively high due to its popularity. The high probability makes this research to have more contributions to industries. In addition, complexity of fault diagnosis in adder circuits is proportional to the number of bits. Therefore, adder is useful to test the performance and accuracy of fault diagnosis with different levels of complexity of fault diagnosis by applying the fault diagnosis on adders with different numbers of bits. 4-bit adder, 5-bit adder and 6-bit adder represent easy level, medium level and hard level of complexity of fault diagnosis respectively.

The design of the adder circuits is implemented in the FPGA platform, Altera Quartus II (Altera, 2015) by using Verilog Hardware Description language (HDL). On the other hand, the design of the fault diagnosis is implemented in the the Matlab and Neural Network Toolbox (MATLAB, 2014b).

1.6 Thesis Overview

This thesis consists of 5 chapters, namely Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 Methodology, Chapter 4 Results and Discussion, and Chapter 5 Conclusion and Future Works.

First, Chapter 1 introduces the research background, problem statement, objectives and scope of research.

Chapter 2 reviews basic concepts of fault modelling and current manual fault diagnosis used in industries. Then, basic concepts of Artificial Neural Networks (ANNs) and architecture of Multilayer Perception (MLP) are discussed. In addition, three learning algorithms, namely Levenberg-Marquardt (LM), Bayesian Regularisation (BR) and Scaled Conjugate Gradient (SCG) are reviewed as well.

Next, Chapter 3 describes the methodology of the proposed fault diagnosis. The chapter starts with the design of three adder circuits, followed by the design of the fault diagnosis. In addition, the chapter discusses the data generation for ANN training. Furthermore, the chapter discusses the training and learning of the fault diagnosis with MLP. Moreover, the chapter also discusses the performance measurements (Regression and training time) used to analyse and compare the learning algorithms on the fault diagnosis.

Then, Chapter 4 discusses the results of the fault diagnosis in the adder circuits. The optimal number of hidden neurons applied to the fault diagnosis is determined by comparing the results of the fault diagnosis with different numbers of hidden neurons. Furthermore, the chapter discusses the most suitable algorithm for the fault diagnosis among LM, SCG and BR by comparing the performance of each algorithm in the fault diagnosis.

Finally, Chapter 5 concludes and summarises the works done in the research and discusses works that could be done in the future to improve the research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

There are several factors for faults to occur in an integrated circuit (IC) such as component degradation, design bugs, physical defects, etc. Among the factors, physical defect is the main contributor of electrical faults in an IC (Shaw et al., 2006). Physical defects are produced due to the instability of manufacturing processes. Examples of physical defects are open via, open metal line, dielectric breakdown, cracks, crystal imperfections, shorted metal line, etc. In order to detect faults and the location of a fault, fault diagnosis was introduced in manufacturing industries.

In this chapter, basic concepts of fault modelling and current manual fault diagnosis used in industries are discussed in Section 2.2. Next, basic concepts of Artificial Neural Networks (ANNs) and architecture of Multilayer Perception (MLP) are discussed in Section 2.3. In addition, Levenberg-Marquardt (LM), Bayesian Regularisation (BR) and Scaled Conjugate Gradient (SCG) are reviewed in the section as well. Then, the applications of ANN in fault diagnosis of VLSI digital circuits are reviewed in Section 2.4. Finally, a summary of the literature review is made in Section 2.5.

2.2 Fault Diagnosis of VLSI Digital Circuit

A fault in an integrated circuit is defined as a deviation of a circuit from its specified behaviour (Shaw et al., 2006). A fault could be caused by a physical defect

in the circuit. Faults are undesirable as they result in a yield loss of fabrication of integrated circuits and an increase of fabrication cost. Thus, it is important to detect the fault and isolate the location of a fault in the circuit. Fault diagnosis is commonly used to model and detect a fault in circuits. In this regard, fault modelling and fault diagnosis are discussed in this section to understand the concept of fault diagnosis.

2.2.1 Basic Concepts of Fault Modelling

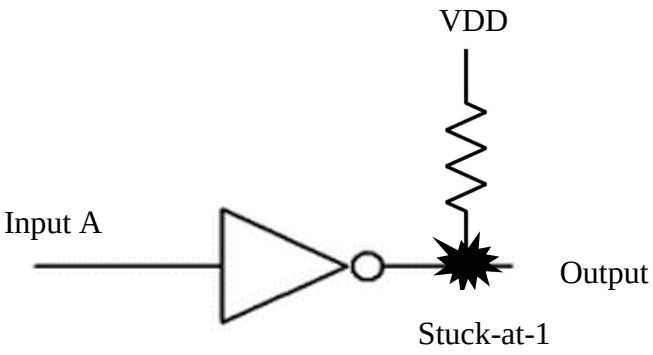
It is necessary to understand the fault modelling of circuit before going to the discussion of fault diagnosis, as the fault diagnosis is used to model and detect a fault in circuits. The basic fault models include stuck-at fault, delay fault, bridging fault, open fault, etc. In this section, stuck-at fault and bridging fault are discussed as they are the major contributors of most of the faults (Ferguson and Larrabee, 1991, Ferguson and Shen, Nov 1988, Sousa et al., 1991).

2.2.1.1 Stuck-at Fault

Stuck-at fault consists of two kinds of components, stuck-at-1 and stuck-at-0. Stuck-at-1 is a fault that causes a node in the circuit to be always stuck at logical ‘1’ regardless of the inputs to the circuit (Breuer and Friedman, 1976). Similarly, stuck-at-0 is a fault causing a node in the circuit to be always stuck at logical ‘0’ independent of the inputs to the circuit.

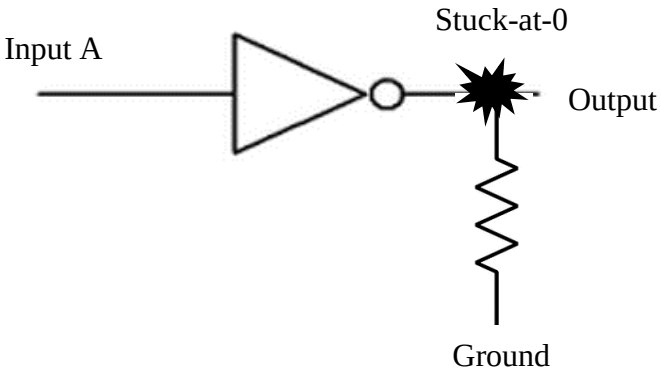
Figure 2.1 illustrates the stuck-at-1 fault. An inverter should invert the input ‘1’ to ‘0’. However, a stuck-at-1 fault causes the inverter to produce output ‘1’ although the input is ‘1’. As illustrated in this figure, stuck-at-1 fault is equivalent to a pull-up circuit to voltage VDD. Similarly, stuck-at-0 fault is equivalent to a pull-

down circuit to ground as shown in Figure 2.2. Due to the stuck-at fault, the inverter loses the functionality of inverting the input. Example of defect that can cause a stuck-at fault is a short between the node and the power supply or ground.



Input A	Output O
0	1
1	**1**

Figure 2.1: Illustration of the stuck-at-1 fault



Input A	Output O
0	**0**
1	0

Figure 2.2: Illustration of the stuck-at-0 fault

2.2.1.2 Bridging Fault

Bridging fault is a connection between two nodes where they should not be connected in the design (Vierhaus et al., 1993). Figure 2.3 illustrates an example of a bridging fault. Bridging fault can be represented as a resistor connecting two nodes

as shown in the figure. Bridging fault is dormant when the two connected nodes are driving the same logical value (Shaw et al., 2003). However, a logic contention occurs when the two gates at node 1 and node 2 are driving different logical values. For example, the OR gate is driving ‘1’ at node 1 while the inverter gate is driving ‘0’ at node 2 in the figure. There are a few factors that determine the final logical value at the nodes, whether the nodes adopt either logical value of ‘1’, ‘0’ or intermediate voltage level which is undistinguishable by digital circuits. The main factors are the difference in the drive strength of the two gates, the characteristics of the bridging fault and the input patterns. For example, the final logical value at the node 1 may adopt the logical value at the node 2 if the drive strength of gate at node 2 is much greater than the drive strength of gate at node 1.

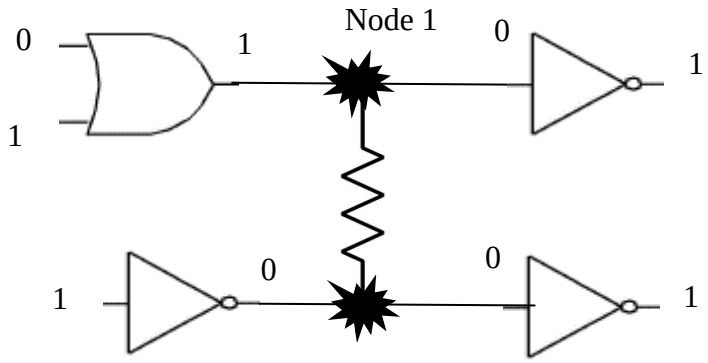


Figure Node 2 ing fault

2.2.2 Manual Fault Diagnosis in Industries

Figure 2.4 shows the methodology of fault testing and manual fault diagnosis used in industries (Kheng, 2010). The method starts with inputting the test pattern through an Automatic Testing Equipment (ATE). The test pattern contains an input sequence that can differentiate the correct behaviour and the faulty behaviour of a

circuit. ATE is an equipment that performs test on a device, known as Device Under Test (DUT). Then, the circuit is under test for output data collected from the circuit. The data collection is necessary for analysing and modelling a fault in the circuit.

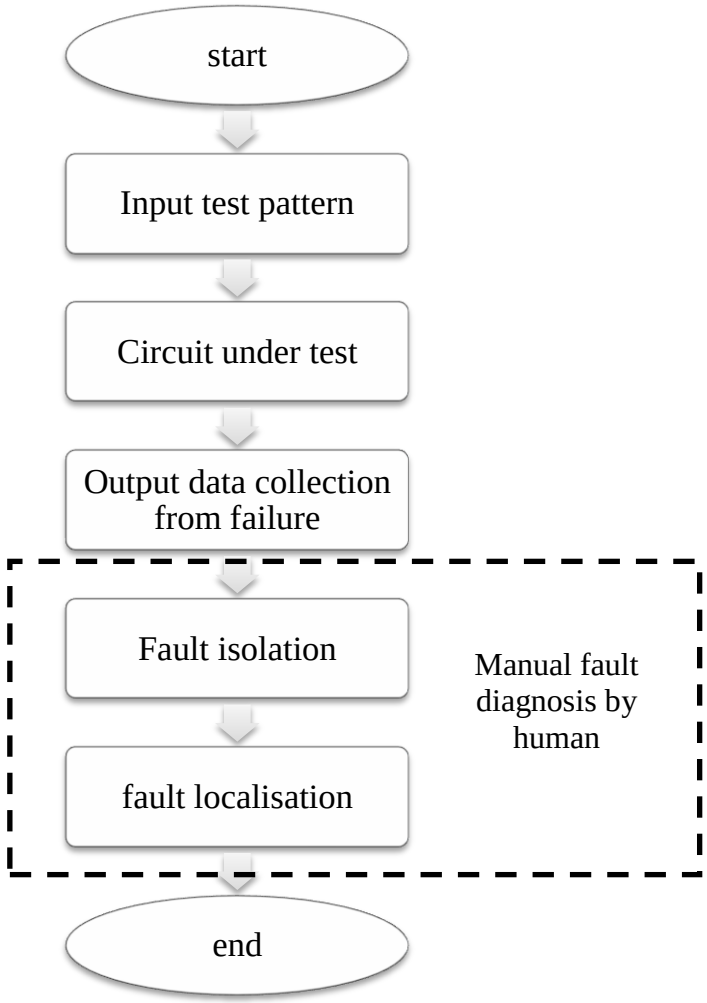


Figure 2.4: The methodology of fault testing and manual fault diagnosis in industries

After the data collection, the data is then used for the manual diagnosis by the human. The manual fault diagnosis has two main steps, i.e. fault isolation and fault localisation. Fault isolation is a technique of detecting a fault and minimising the number of suspects that may cause the fault. Fault localisation is a technique of identifying the location of the fault among the suspects. Finally, the fault is localised

by the manual fault diagnosis result and used to improve the manufacturing process and yield.

In order to have better understanding about the manual fault diagnosis, let's consider the case study of manual fault diagnosis for a faulty full adder. Figure 2.5 illustrates a full adder circuit which consists of 2 XOR gates, 2 AND gates and 1 OR gate. On the other hand, Table 2.1 shows the truth table of a full adder and the failure's outputs of the case study when stuck-at-1 fault occurs. The table compares the expected outputs and failure's outputs for the purpose of manual analysis.

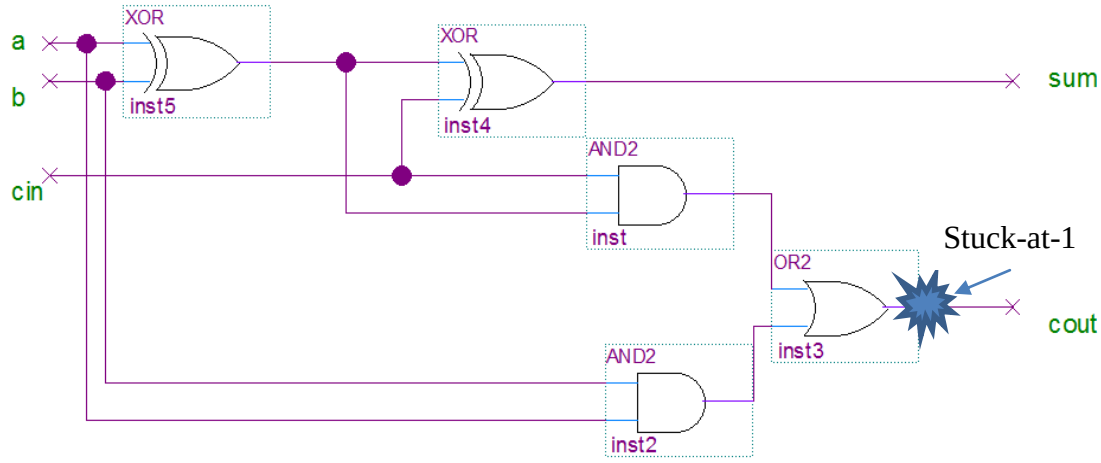


Figure 2.5: Case study of manual diagnosis for a faulty full adder

Table 2.1: The truth table of a full adder and the failure's outputs of the case study

Inputs			Expected outputs		Failure's outputs	
a	b	Cin	Cout	sum	Cout'	sum'
0	0	0	0	0	**1**	0
0	0	1	0	1	**1**	1
0	1	0	0	1	**1**	1
0	1	1	1	0	1	0
1	0	0	0	1	**1**	1
1	0	1	1	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

From the table, it is clear that there is a fault presents in the full adder circuit, which causes the outputs to be different from the expected outputs. In order to manually diagnose the failure to localise the fault, certain hypothesis and conclusion can be made by analysing the failure's data. By the data analysis at the failure's outputs, there are two observations obtained:

1. The Sum output behaves as the expected output according to the changes in inputs.
2. The Cout output is always stuck at 1 regardless of the changes in inputs.

From the observation 1, since sum output always behaves correctly according to the changes in inputs, the hypothesis that could be made is the upstream of sum (two XOR gates) and sum are not faulty, as faultless gates should give correct circuit behaviours. Thus the XOR gates can be eliminated as our suspects of fault. On the other hand, another hypothesis is made from the observation 2. The fault must present either from the upstream of Cout or Cout itself (two AND gates and one NOR gate). Two AND gates are eliminated as the suspects because if either one of them is faulty, obtained sum should be only partially stuck at 1 with certain changes in inputs.

Therefore, the manual fault diagnosis result and conclusion of this case study is the output of OR gate, which is always stuck at 1, is the only suspect of this failure.

2.2.3 Problems of Fault Diagnosis in Industries

The manual fault diagnosis is still manageable for humans to solve simple cases such as the previous case study of a full adder. Since there are only 3 inputs and 2 outputs, which give only 8 possible combinations of input data and 4 possible combinations of output data, a person may solve the manual fault diagnosis case study in 5 to 10 minutes, depending on the fault diagnosis experience and intelligence of the person. Moreover, there are only 5 logic gates in the circuits that are possible to cause the fault. However, it may be very difficult and time-consuming for a person to solve a case with a large number of inputs, outputs and logic gates. As an example, for a 6-bit adder, it consists of 6 connected full adders with 6 bits of input a, 6 bits of input b, 6 bits of output sum and 1 bit of output Cout. There are a total of $2^{12} = 4096$ combinations of inputs, $2^7 = 128$ combinations of output and $5X6 = 30$ logic gates. The person has to analyse all the possible combinations and logic gates to obtain the manual fault diagnosis result, which is very difficult and time-consuming. If we make an assumption that the case of 1-bit adder with 8 combinations of inputs takes 10 minutes of manual fault diagnosis, a 6 bit adder case with 4096 combinations of inputs may take approximately $\frac{4096}{8} x\ 10min = 85\ hours$ of manual fault diagnosis. In addition, the manual diagnosis result might not be accurate due to certain factors such as human errors, complexity of circuits, etc.

Therefore, an approach using Artificial Neural Network to develop a fast and accurate fault diagnosis is proposed in this research to overcome the difficulties in manual fault diagnosis in industries.

2.3 Artificial Neural Networks (ANNs)

Due to the ability to learn between input variables and output variables, ANNs have the capability of pattern recognition and machine learning. There are several types of ANNs such as multilayer perception network (MLP), Hopfield network, etc. In this section, the biological neuron, basic concepts of ANNs, Multilayer Perceptron ANN and three learning algorithms, namely LM, BR and SCG are discussed.

2.3.1 Biological Neuron

The human brain has over 100 billion of interconnected neurons. Neurons use the interconnected networks to transfer information with each other using electrical and chemical signals. They are separated by a tiny gap, namely synapse. Each neuron processes and exchanges information with as many as 50000 neurons.

Figure 2.6 illustrates a biological neuron. A neuron has three main components, i.e. the dendrite, the cell body and the axon (Hagan et al., 2014). The dendrites are tree-like receptive networks of nerve fibres receiving electrical and chemical signals into the cell body. The cell body sums and thresholds all the signals received by the dendrites. The axon is a single long fibre transmitting the signals from the cell body to other neurons.

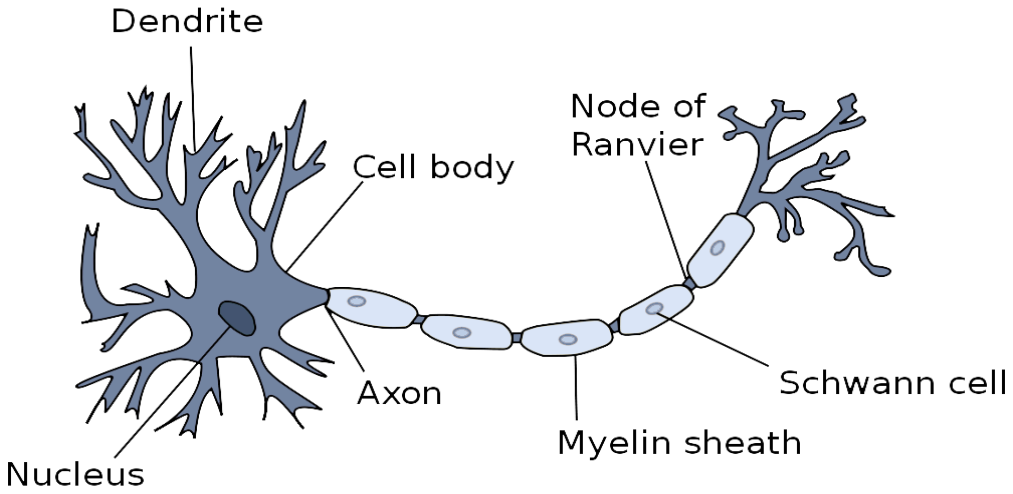


Figure 2.6: Illustration of a biological neuron (Kriesel, 2005)

The functions and architecture of a neural network are established by the arrangement and the strengths of the synapses. There are two types of synapse. The first type of synapse is the electrical synapse, which receives and transfers electrical signals. The electrical synapse is the simpler variant. There is a direct, strong and unadjustable connection between the transmitter and the receiver. The second type of synapse is the chemical synapse, which transmits and receives chemical signals (Kriesel, 2005). It is more complex and powerful than the electrical synapse. It is a one-way connection, due to the fact that there is no direct electrical connection between the synaptic areas. Moreover, the chemical synapse is adjustable. The adjustability enables a large number of different nerotransmitters that can be released in various quantities in a synaptic cleft.

2.3.2 Basic Concepts of Artificial Neural Networks (ANNs)

An Artificial neural network ANN) is a mathematical model of biological neural structure. The basic concepts of a single perceptron were introduced by Rosenblatt (Rosenblatt, 1957). A model of single-input neuron is shown in Figure

2.7. It is a simple mathematical model which consists of three sets of mathematical rules, i.e. multiplication, summation and transfer function. The scalar input p is multiplied by the scalar weight *w* to form *wp*. On the other hand, the other input, 1, is multiplied by a bias *b*. Then, term *wp* and *b* are passed to the summer, The summer output *n*, which is the net input, goes into a transfer function *f*, which produces the scalar neuron output *a* (Hagan et al., 2014).

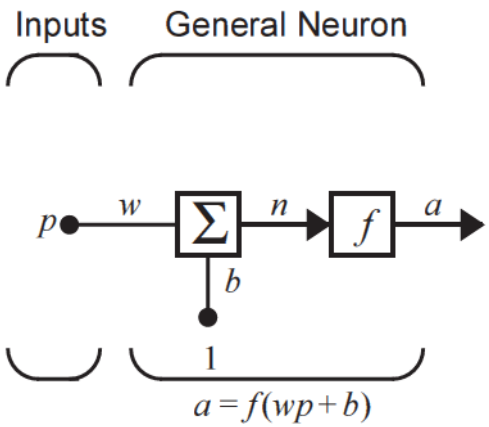


Figure 2.7: Model of a single-input neuron (Hagan et al., 2014)

The equation below shows the neuron output, *a*:

$$a = f(wp + b) \qquad (2.1)$$

where *w* and *b* are both adjusted by some learning algorithms applied to the neural network so that the neuron input and output relationship meets the specified requirement.

The neuron output depends on the transfer function *f* that is chosen for the neural network. The transfer function, *f* may be a linear or a nonlinear function of *n*. In order to solve the problem that the neuron is trying to solve, a particular transfer function is chosen to satisfy the specifications of the problem.

There are many types of transfer functions used in neural networks. For example, Figure 2.8 shows the hard limit transfer function, which sets the output of the neuron to 0 if the function argument is less than 0, or 1 if the argument is greater than or equal to 0.

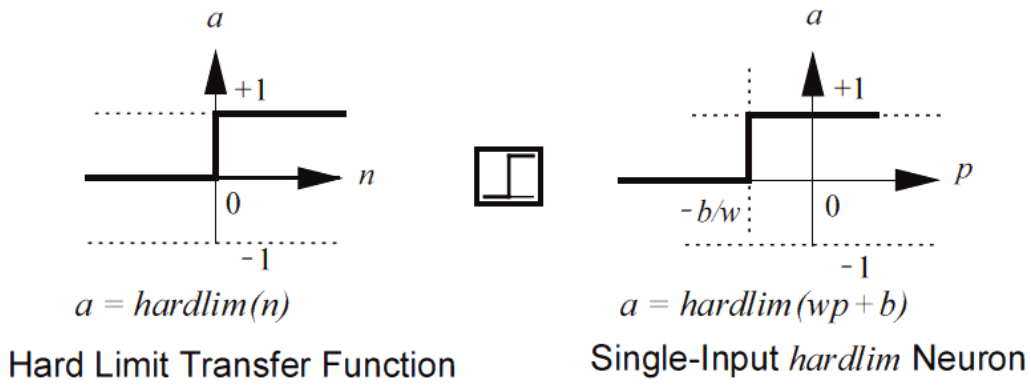


Figure 2.8: Hard limit transfer function (Hagan et al., 2014)

The equation below shows the neuron output, a with hard limit transfer function:

$$a = hardlim(wp + b) \qquad (2.2)$$

Figure 2.9 illustrates another type of transfer function used in neural networks, log-sigmoid transfer function.

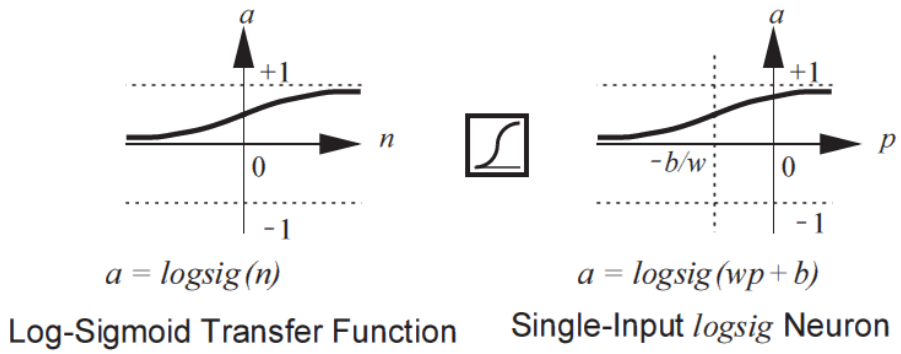


Figure 2.9: Log-sigmoid transfer function (Hagan et al., 2014)

The log-sigmoid transfer function takes the input and squashes the output into the range of 0 to 1, as shown as equation 2.3:

$$a = \frac{1}{1+e^{-n}} \qquad (2.3)$$

A neuron with R inputs is shown in Figure 2.10. The input p_1, p_2, … p_R are weighted by w_1, w_2,… w_R respectively of the weight matrix W.

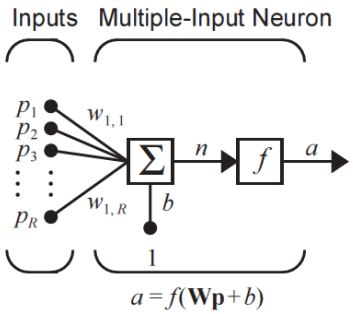


Figure 2.10: Multiple-input neuron (Hagan et al., 2014)

The net neuron input, *n* can be expressed by:

$$n = w_1 p_1 + w_2 p_2 + \cdots + w_R + b \qquad (2.4)$$

n in matrix form:

$$n = Wp + b \qquad (2.5)$$

and the neuron output, *a* is expressed as

$$a = f(Wp + b) \qquad (2.6)$$

A single perceptron is not very useful due to its limited mapping ability. Moreover, single perceptron is not able to solve complex problems. Hence, multilayer perceptron (MLP) is created to overcome the limitations of single perceptron.

2.3.3 Multilayer Perceptron (MLP) Neural Networks

A neuron network, which has multiple layers of neurons, is known as a multilayer perceptron (MLP) (Hagan et al., 2014). The following section explains basic concepts and architecture of MLP.

2.3.3.1 Architecture of Multilayer Perceptron (MLP)

Figure 2.11 shows the architecture of a MLP with an input layer, two hidden layers and an output layer. A layer whose output is the network output is an output layer, a layer whose receives the input from network input is an input layer and the other layers are hidden layers. Each layer has its own weight matrix *W*, its own bias vector *b*, a net input vector *n* and an output vector *a*.

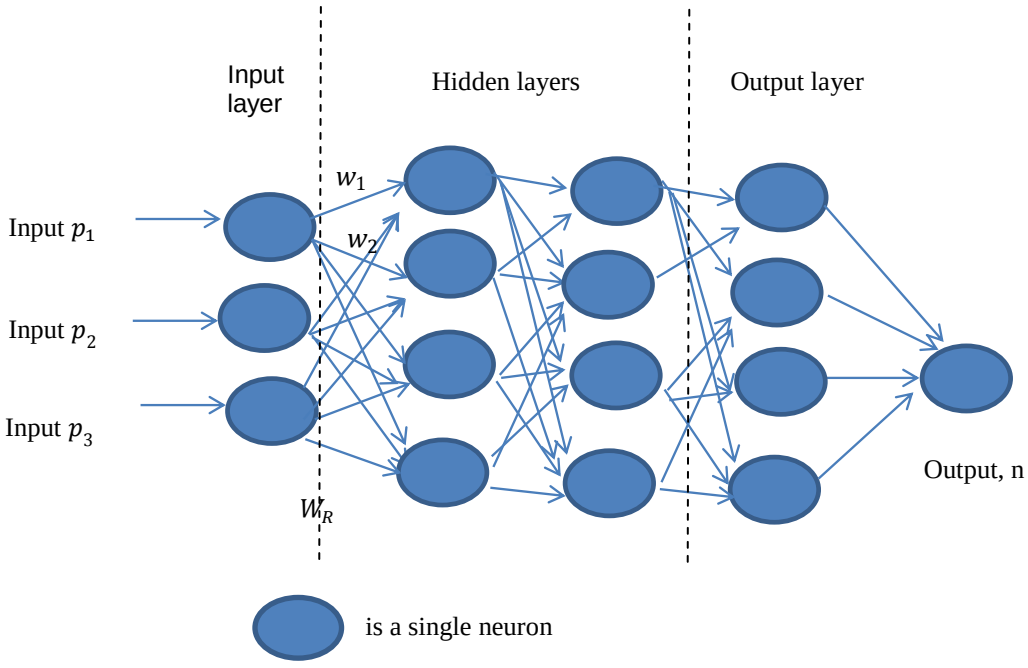


Figure 2.11: Multilayer perceptron (MLP)

There are R inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, and S^3 neurons in the third layer. Different layers can have different numbers of neurons. The output neurons can be expressed as:

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) \qquad (2.7)$$

MLPs are more powerful than single perceptron neural networks. Each of the layers may have different type of transfer function, which makes MLPs can be specified and used to solve complicated problems. The architecture of a MLP is determined by problem specifications, including the specific number of inputs and outputs, and the output signal characteristic. For example, if there are eight variables to be the inputs, there must be eight input neurons in the input layer. Similarly, if there are five outputs from the neural network, there must be five output neurons in the output layer. If the outputs have to be either 0 or 1, then a hard limit transfer function could be used to give the outputs. The number of hidden layers may be determined by the optimal number of neurons needed in a hidden layer.

2.3.3.2 Learning Algorithms

ANNs have the capability of solving problems due to its capability of learning. ANNs can be trained by a number of learning algorithms. There are three main learning algorithms, i.e. supervised learning, unsupervised learning and reinforcement learning. In this section, three supervised learning algorithms include Levenberg-Marquardt (LM) algorithm, Bayesian Regularisation (BR) algorithm as well as Scaled Conjugate Gradient (SCG) algorithm are discussed.

2.3.3.2.1 Levenberg-Marquardt (LM) Algorithm

There has been many researches on methods to accelerate the convergence of the backpropagation (BP) learning algorithm since the algorithm was created (Rumelhart et al., 1986). Levenberg-Marquardt (LM) algorithm was introduced by Levenberg and Marquardt to improve the convergence of the quasi-Newton method (Hagan and Menhaj, 1994). LM is a method which minimises functions that are sums of squares of other nonlinear functions (Hagan et al., 2014).

Since the LM algorithm is a variation of Newton's method from BP algorithm, BP algorithm will be introduced first. BP is a learning algorithm which uses the method of gradient descent by finding the minimum of the error function in weight. The training of BP neural network is used to obtain a balance between the ability to respond correctly to the input variables that are used for training and the ability to provide a good response to the input variables that are similar.

Consider a MLP neural network, the net input to unit *i* in layer *k*+1 is (Hagan and Menhaj, 1994)

$$n^{k+1}(i) = \sum_{i=1}^{Sk} w^{k+1}(i,j)a^k(j) + b^{k+1}(i) \qquad (2.8)$$

The output of unit *i* is

$$a^{k+1}(i) = f^{k+1}(n^{k+1}(i)) \qquad (2.9)$$

The system equations for an *M* layer network in matrix from are

$$a^0 = p \qquad (2.10)$$

$$a^{k+1} = f^{k+1}(W^{k+1}a^k + b^{k+1}), \quad k = 0,1,..,M-1 \qquad (2.11)$$

The performance index of this network is

$$V = \frac{1}{2}\sum_{q=1}^{Q}(t_q - a_q^M)^T(t_q - a_q^M) = \frac{1}{2}\sum_{q=1}^{Q} e_q^T e_q \quad (2.12)$$

where a_q^M represents the output of the network for qth input and $e_q^T = t_q - a_q^M$ represents the error for the qth input. By approximate steepest descent rule, the performance index is approximated by

$$V = \frac{1}{2} e_q^T e_q \quad (2.13)$$

where the total sum of squares is represented by the squared errors for a single input-output pair. Hence, the steepest descent algorithm is then

$$\Delta w^k(i,j) = -\propto \frac{\partial V}{\partial w^k(i)} \quad (2.14)$$

$$\Delta b^k(i) = -\propto \frac{\partial V}{\partial b^k(i)} \quad (2.15)$$

Where $\propto$ is the learning rate.

The sensitivity of the performance index to changes in the net input is expressed as:

$$\delta^k(i) \equiv \frac{\partial V}{\partial n^k(i)} \quad (2.16)$$

Further derive from the equations, the sensitivity is equal to

$$\delta^k(i) = \frac{\partial V}{\partial b^k(i)} = F^k(n^k) {W^{k+1}}^T \delta^{k+1} \quad (2.17)$$

where

$$F^k(n^k) = \begin{bmatrix} (1)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f^k(n^k(Sk)) \end{bmatrix} \quad (2.18)$$

and

$$f^k(n) = \frac{df^k(n)}{dn} \quad (2.19)$$

This recurrence relation is then initialised at the final layer

$$\delta^k(i) = -F^M(n^M)(t_q - a_q) \quad (2.20)$$

LM algorithm improved the backpropagation with Netwon's method instead of steepest descent algorithm (Marquardt, 1963). By using Newton's method, a function which we want to minmise can be expressed as

$$\Delta x = -[\nabla^2 V(x)]^{-1} \nabla V(x) \quad (2.21)$$

where $\nabla^2 V(x)$ is the Hessian matrix and $\nabla V(x)$ is the gradient.

The sum of squares function is equal to

$$V(x) = \sum_{i=1}^{N} e_i^2(x) \quad (2.22)$$

Thus,

$$\nabla V(x) = J^T(x) e(x) \quad (2.23)$$

$$\nabla^2 V(x) = J^T(x) J(x) + S(x) \quad (2.24)$$

where $J(x)$ is the Jacobian matrix

$$J(x) = \begin{bmatrix} \frac{\partial e_1(x)}{\partial x_1} & \frac{\partial e_1(x)}{\partial x_2} & \cdots & \frac{\partial e_1(x)}{\partial x_n} \\ \frac{\partial e_2(x)}{\partial x_1} & \frac{\partial e_2(x)}{\partial x_2} & \cdots & \frac{\partial e_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(x)}{\partial x_1} & \frac{\partial e_N(x)}{\partial x_2} & \cdots & \frac{\partial e_N(x)}{\partial x_n} \end{bmatrix} \quad (2.25)$$

and

$$S(x) = \sum_{i=1}^{N} e_i(x)\, \nabla^2 e_i(x) \qquad (2.26)$$

$S(x) \approx 0$ is the assumption of the Gauss-Newton method, therefore

$$\Delta x = [\mathrm{J}^T(x)J(x)]^{-1} J^T(x)e(x) \qquad (2.27)$$

LM algorithm modified the Gauss-Netwon method to

$$\Delta x = [\mathrm{J}^T(x)J(x) + \mu I]^{-1} J^T(x)e(x) \qquad (2.28)$$

where μ is the parameter that is multiplied by a factor of β whenever a step will result in an increase in $V(x)$.

The key step in the LM algorithm is the computation of the Jacobian matrix. It can be computed by a simple modification to the backpropagation (BP) algorithm. The standard BP algorithm is

$$\frac{\partial V}{\partial w^k(i,j)} = \frac{\partial \sum_{m=1}^{SM} e_q^2(m)}{\partial w^k(i,j)} \qquad (2.29)$$

And the terms needed to be calculated for the Jacobian matrix

$$\frac{\partial e_q(m)}{\partial w^k(i,j)} \qquad (2.30)$$

These terms can be computed using the standard BP algorithm with the modification at the final layer

$$\Delta^M = -F^M(n^M) \qquad (2.31)$$

LM algorithm can be sumarised into five steps.