Universidad de Sonora

División de Ciencias Exactas y Naturales

Departamento de Investigación en Física

Maestría en Ciencias en Electrónica

Diagnostic of Resistive Faults in VLSI Circuits Applying Neural Networks

T E S I S

Presentada en cumplimiento de los requisitos para obtener el grado de **Maestro en Ciencias en Electrónica** presenta:

Melissa Tovar Pino

Director: **Dr. Roberto Gómez Fuentes**

Co-Director: **Dr. José Rafael Benito Noriega Luna**

Hermosillo, Sonora, México, Diciembre 2021

Universidad de Sonora

Repositorio Institucional UNISON

Acknowledgments

I would like to acknowledge the University of Sonora and the Academic Committee for accepting me into the graduate program. I especially want to recognize the guidance of Dr. Roberto Gomez Fuentes, Dr. Jose Rafael Benito Noriega Luna and Dr. Dainet Berman Mendoza.

Dedication

For my parents who have been there every time I have needed them and for giving me the space to grow. For my friends who cheered me on and understood when I disappeared for weeks. I am forever grateful to have you in my life and hope to find myself among your good company always. And for me, for surviving the year 2020, a pandemic and my thesis.

Contents

1 Introduction **7**
- 1.1 Hypothesis . . . 8
- 1.2 Justification . . . 8
- 1.3 Objectives and Goals . . . 8
  - 1.3.1 General Objective . . . 8
  - 1.3.2 Specific Objectives . . . 8
  - 1.3.3 Goals . . . 9
- 1.4 Fault Models . . . 9
  - 1.4.1 Structural Faults . . . 9
  - 1.4.2 Resistive-Opens . . . 12
  - 1.4.3 Resistive-Open Model . . . 13
  - 1.4.4 Methods of Diagnosing . . . 16
- 1.5 Chapter Organization . . . 17

2 Artificial Neural Networks **19**
- 2.1 Artificial Neural Networks . . . 19
  - 2.1.1 Perceptron . . . 20
  - 2.1.2 Multilayer Perceptron . . . 21
  - 2.1.3 Propagation in an Artificial Neural Network . . . 22
  - 2.1.4 Types of Learning . . . 23
  - 2.1.5 Activation Function . . . 24

3 Methodology and Experimentation **27**
- 3.1 Methodology . . . 27
- 3.2 Experimentation . . . 36

4 Results **43**
- 4.1 Simulations . . . 43
- 4.2 ANN Training . . . 51
- 4.3 Diagnostic of Resistive Faults . . . 53

5 Conclusions **63**

Chapter 1

Introduction

Chips are getting smaller due to the reduction in transistor size. The reduction in the size of transistors means smaller technologies are used, which in turn results in smaller dies in size. This means an increase in the amount of dies able to fit on a silicon wafer. Very large-scale integration (VLSI) is a process where an integrated circuit is created by hundreds of thousands of transistors on a single chip [1]. The number of transistors allowed on these chips has increased highly due to years of advancement in the manufacturing process of transistors. While all new chips are tested [2] and the design for one of these chips is adequate to fulfill the function it has, some chips will have an error in one of the many connections these chips have.

Some of these errors will stem from structural failures due to issues in the integrated circuits manufacturing process, materials, or chemical composition [1] [3]. It may mean that a redesign in the fabrication process must be considered in order to reduce the number of defects. This could indicate a small change in doping, the use of a different mask in the lithography phase or having to change a different step in the methodology when fabricating the chip [4]. Chips are tested after manufacturing in batches and knowledge about faults is common among individuals conducting tests to find defective chips. These testers will normally depend on machines dedicated to testing chips for certain faults, but updating or repairing these machines may get costly over the years.

As technology becomes smaller and faster, the number of transistors increases. It is far easier for conventional testing methods to fail as the structure of the circuit becomes more complex and traditional testing methods are left outdated. Some testing methods can be adapted and enhanced, but those methods become more costly monetarily over time when having to upgrade machinery [5] [1]. The cost of maintenance for such machinery is also worth acknowledging, not to mention the time taken to learn using a new machine is testing time lost and impacts the manufacturing process.

This project is about an option for finding resistive-open defects with the application of artificial neural networks. This first chapter will go over the the justification of this project and the main concepts to better understand the resistive-open fault. A hypothesis

will be presented afterwards along with the objective and goals of the project. Finally, the chapter will end with an explanation of the organization of the chapters for this thesis.

1.1 Hypothesis

It is possible to create a method for the diagnostic of resistive faults in VLSI circuits using neural networks.

1.2 Justification

Various interconnections allow the function of an integrated circuit, yet it is possible for the electronic system to be faulty. On a far more reduced scale, it is fair to mention that is also expected with VLSI circuits. In the variety of manufacturing processes several different faults may present themselves involving connections between gates, levels of metal or tracks [1]. The probability of finding resistive-open faults in VLSI circuits increases with interconnection density. Therefore, it is important to have a reliable method to be able to diagnose specific faults, such as the resistive-open fault. This project proposes the application of neural networks to make a diagnosis of resistive-open faults in VLSI circuits.

Neural networks have the useful advantage of "learning" and strengthening every time they comply with their function following the programmed algorithm [6]. This is a vital element for a diagnostic that will possibly be repeated on a great number of integrated circuits that have been fabricated. The testing process would be facilitated and more automatized. A neural network diagnostic will reduce testing time and facilitate the testing procedure drastically over time. This application of neural networks may allow for human error to be reduced and for there to be an increase in quality.

1.3 Objectives and Goals

1.3.1 General Objective

The general objective is to create a method for the diagnostic of resistive faults with neural networks for future application in VLSI circuits.

1.3.2 Specific Objectives

The specific objectives of this thesis are as follows:

- Simulate interconnections with resistive faults.

- After gathering data from simulations and training an artificial neural network, verify the neural network's results from training to report its success.
- Analyze larger circuits such as benchmark circuit ISCAS 85.

1.3.3 Goals

The goals to achieve the main objectives are:

- Simulate circuits which have fault-free and faulty interconnections.
- Organize the data to train the artificial neural network.
- Test the artificial neural network's performance.
- Establish the bases for circuit tests of a larger scale.

1.4 Fault Models

This chapter will define the terms of structural faults in order to better comprehend the resistive-open fault.

Fault models are used to mimic manufacturing defects within an integrated circuit [1]. Resistive-open faults are structural faults that can be found in VLSI and ULSI circuits. As previously mentioned and as defined by [7], VLSI chips contain thousands or millions of metal oxide semiconductor field effect transistors (MOSFETs). Ultra large-scale integration or "ULSI" chips generally contain billions, or more, MOSFETs [7]. As the technology becomes smaller, the number of defects and faults increase. Thus, having knowledge of these structural faults for testing is necessary. In order to diagnose a resistive-open fault, the tester has to understand the conditions for this particular fault to be present.

Research is conducted in VLSI circuits usually aided by software in order to design, test, diagnose faults, simulate and process signals [5] [1].

In order to have a diagnostic using artificial neural networks (ANNs) for resistive-open faults, there are some terms that should be reviewed before getting to the part of training an ANN. All of these matter when it comes to testing and understanding the main issues that may be found within integrated circuits. The concepts related to ANNs will be viewed in the next chapter.

1.4.1 Structural Faults

Fault models are representations of defects. A defect in an electronic system, is considered the unintended difference between the implemented hardware and its intended design [1]. These defects are classified into four categories: process defects, material defects, age

defects, and package defects [8]. These devices are failures, which have their origin in the manufacturing process or from the use of the device [1].

An error is known to be the resulting output signal, which indicates that the circuit behavior has departed significantly from its intended operation. This error has its origin from a defective system, which means it is a direct effect of a defect. A fault represents the defect at an abstract function level [1]. This definition is stating that a fault is, in this case, an electronic system not functional or being able to function completely as expected.

In many circuits, defects may suddenly manifest and hinder the circuit's performance. These may be structural faults, which are caused by different anomalies in a physical part of the circuit, such as its topology or its physical geometry. The term structural faults is typically applied to in gate-level interconnections and not to faults modeled in layouts. Some examples of structural faults are single stuck-at faults and bridging faults [1].

These structural faults are modeled by the following:

Stuck-at Fault: This fault is modeled by assigning a fixed (0 or 1) value to a signal line in the circuit. A signal line is an input or an output of a logic gate or a flip-flop [1].

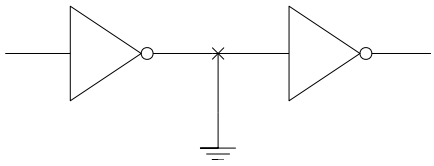


Figure 1.1: The stuck-at model.

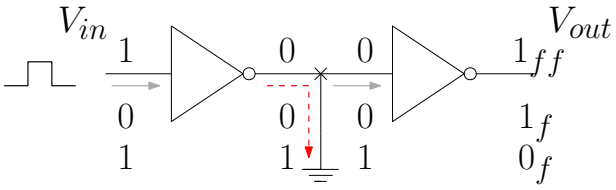


Figure 1.2: The effect of the stuck-at model.

In Figure 1.1, the stuck-at model is shown, but in Figure 1.2 the effect that defines this fault is demonstrated. As a signal enters the circuit through the input V_{in}, it passes through two inverters while splitting in between the two and being grounded, finally the corresponding output signal is given. Initially, the first output value seen is 1_{ff} which is a value that is faulty free (ff) while the following outputs are faulty, since a fixed value was unexpectedly assigned to a signal line.

Stuck-open Fault: The effect is to produce a floating state at the output of the faulty logic gate [1].

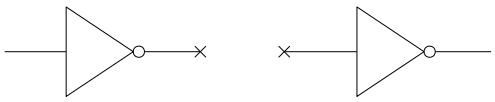


Figure 1.3: The stuck-open model.

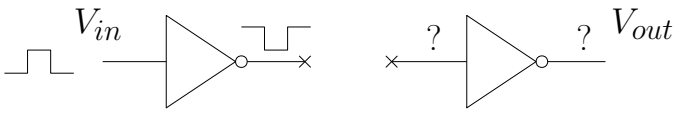


Figure 1.4: The effect of the stuck-open model.

Figure 1.3 is the model used for stuck-open faults. Figure 1.4, demonstrates the effect produced by this fault, as the output value is left with a floating value. The signal is able to enter through V_{in}, but fails to proceed past the second inverter.

Stuck-short or Stuck-on Fault: The effect of this fault is to produce a power supply to ground conducting path. It may occur that a path conducts to a ground and the signal can not reach the next part of the circuit [1].

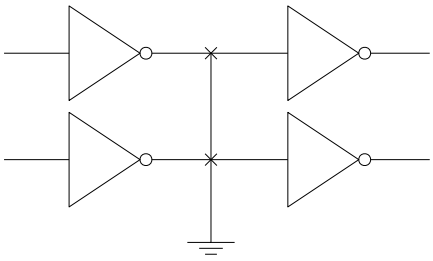


Figure 1.5: The stuck-short or stuck-on model.

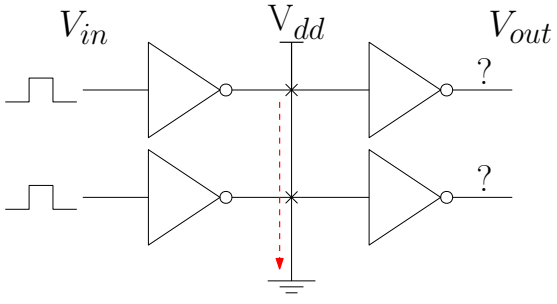


Figure 1.6: The effect of a stuck-short or stuck-on model.

The stuck-short or stuck-on model is presented in Figure 1.5. Figure 1.6 illustrates the defined fault by demonstrating this fault's effect. As shown, a power supply is produced

that conducts to ground, as a result the signal can not pass through the following inverters and there is no output value.
Structural faults also include resistive-open faults, but will be further defined in the next section.

1.4.2 Resistive-Opens

Before defining what a resistive-open fault is, there are a few terms that should be known as well. Such terms are defined as follows:

Open Defect: This defect is physical discontinuity of a line that, in the original design, connects two nodes [9].

Full-open Defect: This defect occurs when the open completely interrupts the electrical connection between two nodes [3].

Resistive-open Defect: The defect happens when the open doesn't completely interrupt the electrical connection between two nodes [3]. As the form of the missing line is difficult to predict, the resistance value of the defect is modeled as continuous resistance [10].

A resistive-open defect can be modeled as a an abnormally large interconnecting resistor or interpreted as a defect resistor between two circuit nodes that should be connected in between two logic gates [11]. To clarify this interpretation, the connection between the two nodes is actually a small resistance metallic conductor, which is made up of multiple metal sheets. This defect may occur when a part of an interconnection path is far thinner than the rest and causes an effect similar to that of a resistor [11]. This means that a resistive-open fault is a model that represents the behavior caused by a defect in an interconnection. This defect makes it act as a resistor by reducing the amount of space the electrons have in order to move across between two nodes.

As mentioned by José Luis García Gervacio [12], defects related to interconnections have become an outstanding issue when it comes to technology on a nanometer scale. This is due to the increase in the number of transistors in newer technologies. With this increase, there is a higher number of metal layers used and consequently lead to an increase of parasitic capacitance and inductance [13]. Another cause of resistive-opens may be dishing and erosion in the materials [14]. While there are some open defects that cut off the electricity between two nodes, the ones that do not are considered to be weak open defects. These weak resistive open defects are the cause of small delays [15].

García also mentions previous research about the classification of resistive open defects. Accordingly, strong opens were larger than 10 MΩ and weak opens were less than 10 MΩ [15]. Since weak opens allow for the circuit to function adequately, they are harder to

detect by typical testing methods [12]. Also, while the circuit may continue to function it does so with delays that affect its signals.

1.4.3 Resistive-Open Model

The resistive-open fault model is represented as shown in Figure 1.7. It is represented by two inverters and a defective resistor in between them, which represents the effect resistive-open faults cause.

The fault model is a way to visualize the resistive-open fault's behavior. As can be recalled, any defect within an interconnection may result in the circuit having delays, it is the reason for the resistance set between two inverters. This is to indicate that the electrons will have more trouble advancing through that interconnection, which is best represented by there being resistance between two nodes.

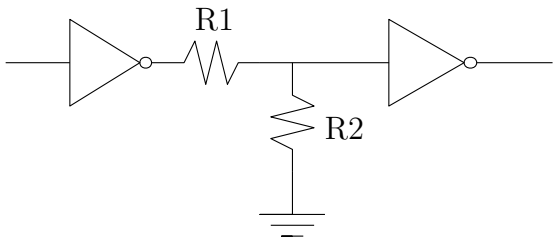


Figure 1.7: The resistive-open fault model.

In the fault model in Figure 1.7, there are two resistors between two inverters. The first resistor is there to represent a defect that may be found in the interconnection between two nodes. This resistor will serve to model a defective interconnection. The second resistor, R2, is connected to the ground in order to compensate for any parasitic conditions.

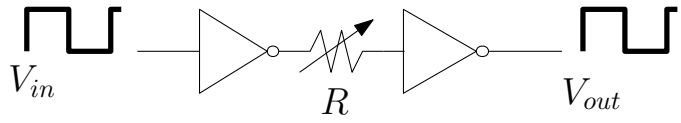


Figure 1.8: The main resistive-open fault model used.

The resistor R in Figure 1.8, is the defective interconnection. If a pulse signal goes through the first inverter, logically the exact same signal should be the output signal of the second inverter. When the resistor happens to be of a value closer to zero, then it might not alter the output signal too much. The real issues appear when that resistance value is higher and glitches appear in the output signal or the integrity of the signal is lost. A high resistance in the interconnection usually results in the output signal being delayed or not responding when it should. One of these results could include the output signal not reaching a high state when the input signal did.

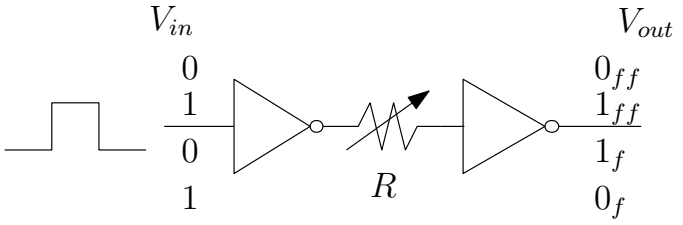


Figure 1.9: The effect of the resistive-open fault model.

The effect of the resistive-open fault is demonstrated in Figure 1.9 and is represented with the different values. The faulty free (ff) output values are the resulting output values that are expected, while the faulty (f) output values are results that may be obtained due to a resistive-open fault. Another result may be a delay or a glitch in the output signal, as previously mentioned, which may be seen in Figure 1.10.

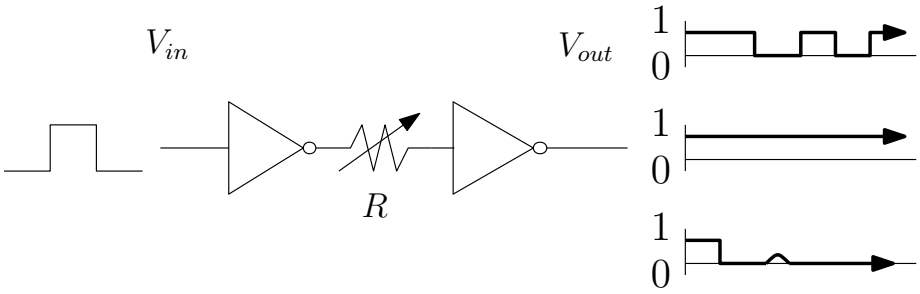


Figure 1.10: Other effects of the resistive-open fault model.

In order to better understand this fault model, the concept of line resistance should be studied. An interconnection's resistance is the result of several metal sheet lined up from one node in the circuit to the next. This line resistance is calculated using the following formula:

$$R_{line} = \rho \frac{d}{A} = \rho \frac{d}{hw} \tag{1.1}$$

In which ρ is the resistivity of the material in units of [Ω-cm] and $A = hw$ is the cross-sectional area of the line [4]. Metals are used in interconnect lines and all of them can be categorized by a low value of resistivity, which is the reason behind their use for interconnections. The width is represented by w and the distance as d. The thickness or height of the material layer is seen as h.

The sheet resistance (R_S) is the end-to-end resistance of a square section of the material with $d = w$ and is defined as [4][16]:

$$R_S = \frac{\rho}{h} \tag{1.2}$$

Therefore, line resistance is computed from:

$$R_{line} = R_s n \tag{1.3}$$

The following formula is the number of squares (n) of dimensions encountered by the current [4]:

$$n = \frac{d}{w} \tag{1.4}$$

The units for n is ohms per square and is observable from the top view of an interconnect. This may be noted in Figure 1.12, where the concept of number of squares and the dimensions are illustrated. In Figure 1.11, a representation of an interconnect line is given from the side to appreciate the location and the attributes of an interconnect.

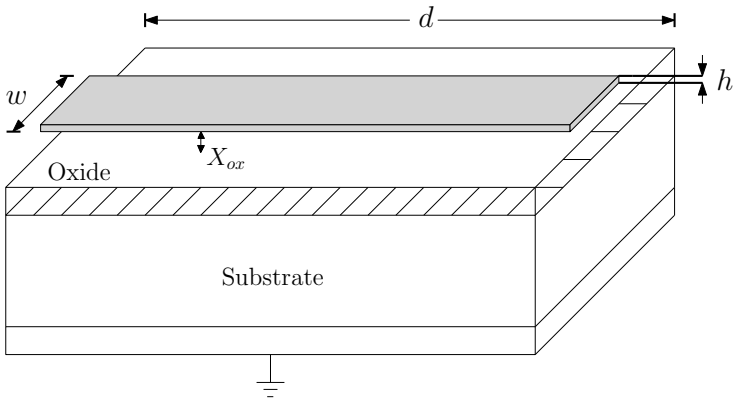


Figure 1.11: The geometrical structure of an interconnect line [4].

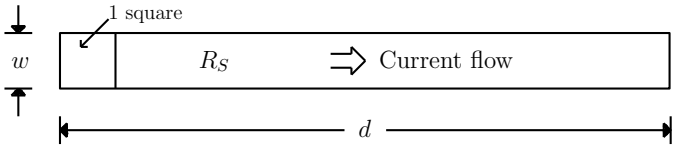


Figure 1.12: The top view of an interconnect [4].

After taking note of the aspects of an interconnect line, it is easy to consider that the interconnect's functionality may be affected under several circumstances. One of these circumstances being in relation to the material not adhering correctly or becoming damaged. It would only take a section to cause issues and would likely affect any signal passing through the interconnect. Knowing the measurements for the interconnect's attributes will allow anyone to find the resistance expected and use it to acknowledge when a resistance value may be considered to influence faulty effects.

1.4.4 Methods of Diagnosing

A crucial part for quality control is to test for faults in chips. In the fabrication of VLSI circuits there can be many complications due to defects. Over the years, some investigations have been done on how to improve this step in the production and eventual sale of integrated circuits. Due to the complexity of the interconnections existing between millions of transistors it is difficult to find all the issues these may present during a quality check. Defining one particular issue may prove to become a complicated task, but this has not dissuaded investigators from continuing their research.

An article [11] explains a method involving the use of transitory current (I_{DDT}) from a power supply source for resistive-open faults with the use of simulation. The results indicated that the testing with I_{DDT} is able to present information about the defects like a locating instrument. At the same time, the team extracted a fault function and configured a dictionary of faults within the Wavelet Analysis, a tool from Matlab. The authors conclude that the combination of transitory current (I_{DDT}) and the wavelet technology in a digital circuit for testing resistive-open defects is effective [11].

Some of the strategies mentioned are found in [3]. Here, it is explained that for the analysis, the use of Middle-of-Line (MOL) interconnections as well as multi-fin and multi-finger design strategies are taken into account. The defective delay behavior is modeled using RC networks for nominal process conditions and under process variations [3].

As stated in [1], a parametric test is important in order to decide whether the chip pins meet various rise and fall times. A parametric test also informs about setup and hold times, low and high voltage thresholds, and low and high current specifications. Functional tests are used to determine if the internal digital logic and analog sub-systems behave as planned in every chip tested [1]. Digital and analog functional tests are considered to have the most cost in testing. The parametric test, in contrast, has a lesser part of the total cost for testing. Since the parametric test takes several seconds to be done, it does not amount to a large sum independently. Yet, the parametric test cost is proportional to the amount of time the tester may need to execute this test, which may vary and be influenced by external factors [1].

The automated test equipment, also referred to as “ATE”, has an increasing cost, but the cost for the actual testing process is decreasing due to improvements in the equipment [1]. This equipment becomes more expensive as its performance is raised. This performance involves storing more vectors than in the past and operating at higher frequencies [1]. As a result, designing a tester probe head becomes harder due the larger number of pins and clock rates. During a digital test, this results in more problems with inductance and electrical noise, which is what all manufacturers wish to avoid.

1.5 Chapter Organization

The following chapters, including this one, are organized as follows and go further into what this work entails:

In this first chapter, an introduction was made to resistive-open faults and concepts relating to it to understand the rest of the chapters when these get mentioned. This also served to state the hypothesis, objectives and goals. As for the second chapter, it will go over terms of artificial neural networks. These first two chapters introduce the main concepts needed to understand the project's necessity and the knowledge needed to make the application of an artificial neural network a reality.

The third chapter will provide details about the methodology for creating a diagnosis for resistive-open fault, the implementation of this methodology, and also go over the experimental phase that influenced the methodology. While the fourth chapter will share the results of the different aspects that were needed to conclude this project, for example the graphs obtained from simulations. Finally, the last chapter will present the conclusions and mention any future work that may be done to further improve this project.

Chapter 2

Artificial Neural Networks

The terms for understanding an artificial neural network will be viewed and defined in this chapter. These terms are necessary in order to understand the function of an artificial neural network and all the parts required to build one of these networks.

2.1 Artificial Neural Networks

Artificial neural networks, also known as ANNs, are programmable algorithms. These algorithms choose between various vectors in order to determine the correct output depending on the criteria it learned from information it was given during the ANN's training. If an ANN is trained to compare various factors and learns to find faults, over time it will improve and each time it compares between these factors it will become more accurate.

Initially, the idea for artificial neural networks arose from the human brain, it was an inspiration due to its structure. Since an ANN is assumed to work in a similar way as the human brain, some of the same terms are used, but their definitions are now in relation to machine learning. Artificial neural networks are a part of artificial intelligence, but are a sub-field of deep learning which happens to be a sub-field of machine learning [17]. Machine learning is constituted by a set of algorithms and these algorithms parse data. From the parsed data, the algorithms learn and discover patterns. An ANN is used in machine learning for modeling data using graphs of neurons [18].

In general, the steps to creating a neural network can be summarized as follows [6]:

- Through a learning process, the ANN gains "knowledge" from the environment it is introduced to.
- Interneuron connection strengths are known as synaptic weights, which are used for the storage of information obtained.

- The learning process modifies the synaptic weights in a systematical manner to achieve the desired objective.

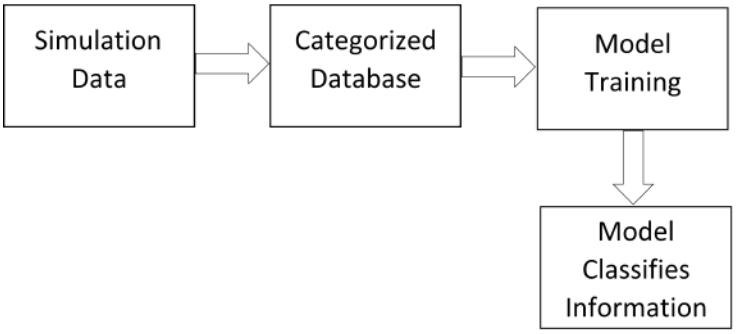


Figure 2.1: The block diagram of the general process for obtaining the neural network for this project.

An artificial neural network is defined by its focus. The focus for this project will be to create a diagnosis for resistive faults using artificial neural networks. The reason it was necessary to investigate resistive faults in VLSI circuits and diagnosis applying ANNs.

2.1.1 Perceptron

A single perceptron is a processing unit of an artificial neural network. It may contain various inputs, an equal number of synapses as inputs, and a single output [19].

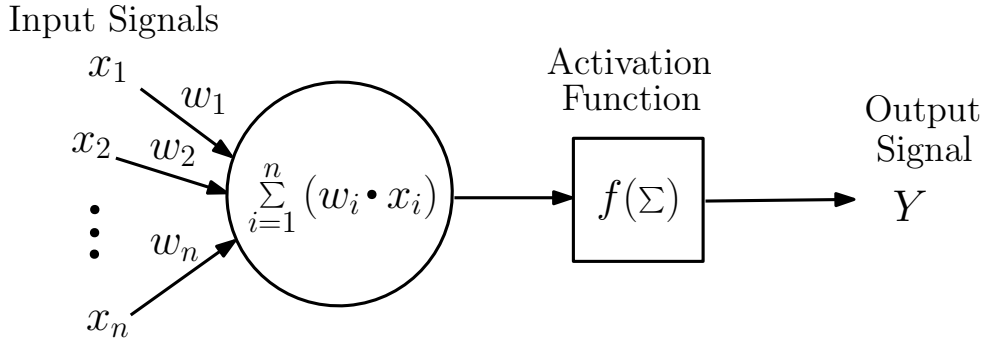


Figure 2.2: A representation of a perceptron [19][20].

In Figure 2.2, x_1, x_2, ..., x_n represent all the input signals going into the neuron. These input signals are multiplied individually by different weight values. Weights represent the strength of the connection between neurons in the network [21]. A bias is added to the sum of the weighted inputs. In order to obtain an output, an activation function is used

on the weighted sum of the input values. It is important to note that during the learning phase, these weights may get adjusted.

2.1.2 Multilayer Perceptron

With a multilayer perceptron, there are now more neurons at work and there are hidden layers involved. The ANN accepts the input signals, then has these signals pass through hidden layers accordingly and finally turns this information into outputs. A hidden layer is a layer between the input and output layers, where the neurons take in a set of weighted inputs and produce an output through an activation function [22]. In backpropagation, these weighted inputs become adjusted and calibrated [6] [21].

Precision in a neural network is achieved through training, proper selection of the number of neuron per layer, selection of the number of layers and other factors. Although perceptrons are better suited in situations that are less complicated, a multilayered perceptron is more suitable when calculations handling larger amounts of data are needed [6]. The hidden layers are defined by the user and may be helpful in some cases, while in others these layers might not be necessary.
Two kinds of signals are identified in this network [6]:

- Function signals
- Error signals

A signal goes in through the input end of the ANN, this input signal propagates forward from neuron to neuron through the network until it reaches the output end as an output signal [6]. As it propagates forward, it is known as forward propagation. The computation of the function signal appearing at the output of each neuron is expressed as a continuous nonlinear function of the input signal and synaptic weights associated with that neuron [6].

While an error signal comes from an output neuron and propagates backward through the ANN. As the error signal propagates backward, it is referred to as backpropagation and in order to compute this signal, every neuron in the network is affected by an error-dependent function. The computation of an estimate of the gradient vector (*i.e.*, the gradients of the error surface with respect to the weights connected to the inputs of a neuron), which is needed for the backward pass through the network [6].

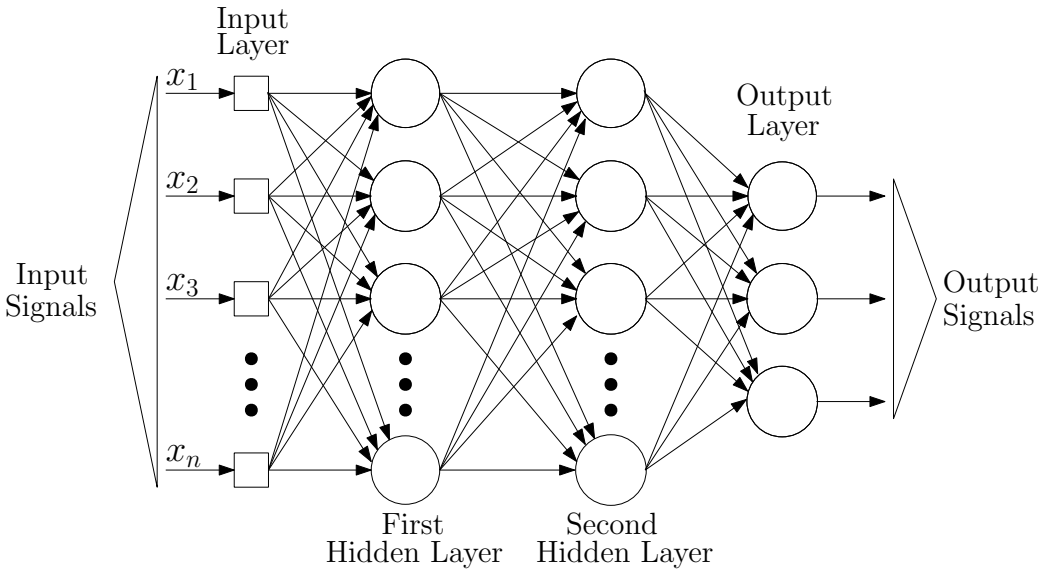


Figure 2.3: Architectural graph of a multilayer perceptron with two hidden layers [6].

A multilayered perceptron is more complex than a simple perceptron, as is noticeable in Figure 2.3. There are several elements that must be considered for the artificial neural network to operate in the optimal conditions and to not have many issues during its training phase. An artificial neural network's success depends on how well trained it is. This is why the estimated completion time (ETA), the number of epochs, and the number of input and output neurons matter. The term "ETA" ordinarily means estimated time of arrival, but in machine learning it stands for the estimated completion time [23]. This ETA is equal to one epoch and is an estimate of how long the network has to train. An epoch is a round in which all the data samples given to the artificial neural network are used for training.

2.1.3 Propagation in an Artificial Neural Network

The function of forward propagation is to produce an output and to compare with the expected output from training data. This comparison will produce an error.

Backpropagation aims to optimize the weights used in the neural network. This is needed for the training of the neural network, as these weights will help correctly make the connection between arbitrary inputs to outputs [6]. As the loss is calculated, that information will propagate backwards from the output layer to the neurons in the hidden layer that directly affect the output. These neurons only receive a fraction of the total loss signal, depending on how much each neuron contributed to the original output signal [6]. This is done continuously until all the neurons in the network have obtained their fraction of the total loss signal.

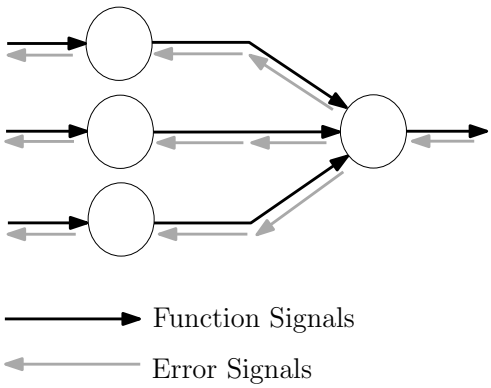


Figure 2.4: Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back propagation of error signals [6].

2.1.4 Types of Learning

In the learning phase, an artificial neural network is given information for its input layer. Yet, there are different ways for it to learn. There are two types of learning: unsupervised and supervised. In this section, both will be looked over.

Unsupervised Learning

In unsupervised learning, algorithms are used for analyzing and clustering unlabeled sets of data. The algorithms used find patterns that are hidden within the data without the need of a user (supervisor) intervening. Since this type of learning is so self-reliant, it's known as unsupervised. The models used for this type of learning have three main tasks: clustering, association and dimensionality reduction [24].

Supervised Learning

By using supervised learning, the artificial neural network will be guided along due to the user deciding the data inputs and the correct outputs before being introduced to the artificial neural network. Supervised learning entails a user being directly involved with the knowledge the ANN learns. The user introduces specific data to the network as input signals, this data allows the neural network to train while calculating an error, then weights and bias are adjusted afterwards. This process continues until the error reaches a value the user deems adequate. If the neural networks happens upon any problem, the user will be there to adjust the ANN. After the training phase, the neural network should have the criteria to decipher between new information that is being introduced.

Batch learning is a part of supervised learning in which adjustments to the synaptic weights of the multilayer perceptron are performed after the presentation of all the N ex-

amples in the training sample that constitute one epoch of training [6]. These adjustments to the synaptic weights are done on an epoch to epoch basis. This indicates that the cost function for this method will be defined by the average error energy.

With batch learning, the learning process of the artificial neural network is done by batches of data being introduced to the ANN. As these batches are introduced, the ANN starts to learn by comparing the data and calculating the error. The artificial neural network will try to reach the error training average established. Such as if there is a certain signal that needs to be replicated, like the initial signal in a circuit, the data gathered allows the network to understand the behavior of certain simulations done beforehand and what the final signal should be. If the goal is to maintain the initial signal, then it stands to reason that both signals should be exactly the same or as close to equal as possible.

2.1.5 Activation Function

Data travels as input signals, which pass through the hidden layers until reaching the output. An activation function is used to define each neuron's output in the network and again at the output end to define the output signal. Also, the activation function may be used to present non-linearity to the network when modeling. As follows, the most basic types of activation functions.

Threshold Function

The threshold function is used when the output depends on a threshold value. Figure 2.5 is of a graph for the visualization of the threshold function and it presents the limitation between the two values of 0 or 1.

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{2.1}$$

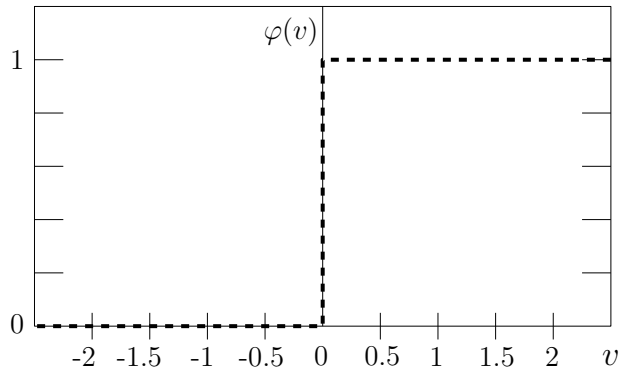


Figure 2.5: A graph of the threshold function [6].

Sigmoid Function

The sigmoid function is used when the model or the neural network is going to be used to predict probability. While a threshold function assumes the value of 0 or 1, a sigmoid function does this with a continuous range of values from 0 to 1 [6]. The sigmoid function is a differentiable function, while the threshold function is not. Figure 2.6 demonstrates the sigmoid function in a graph and its range of various values between 0 and 1.

$$\varphi(v) = \frac{1}{1 + e^{-av}} \tag{2.2}$$

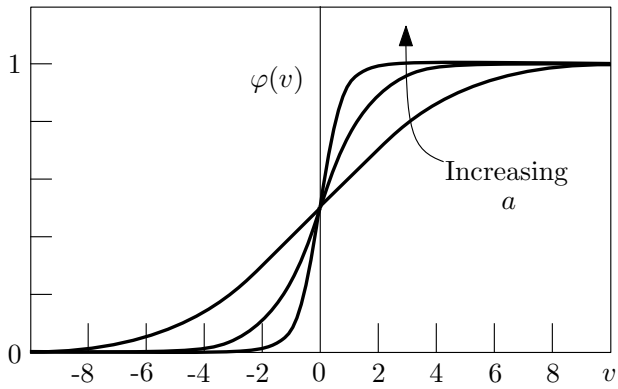


Figure 2.6: A graph of the sigmoid function [6].

Hyperbolic Tangent

In some cases, the activation function should range from -1 to $+1$ which is where the hyperbolic tangent is needed. This activation function is often used in backpropagation. In Figure 2.7, the hyperbolic tangent function is shown limited between -1 and $+1$ while maintaining a continuous range of values.

$$\varphi(v) = \tanh(v) \tag{2.3}$$

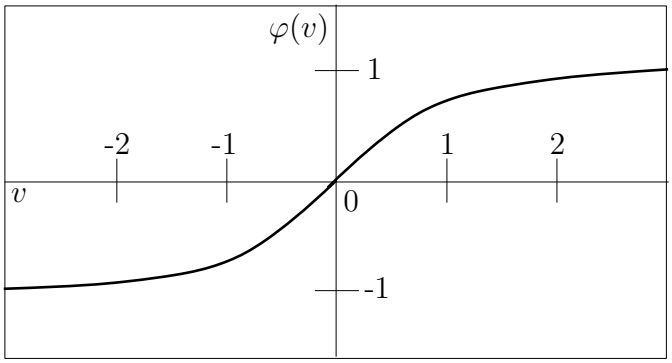


Figure 2.7: A graph of the hyperbolic tangent [6].

Chapter 3

Methodology and Experimentation

In the methodology, the steps taken to produce a trained neural network for resistive faults are described. The implementation of line resistance will be commented on. The experimentation will provide information of everything that happened during the process of following the methodology and highlight some of the issues that occurred.

3.1 Methodology

The methodology for this project will be reviewed in this section and will be more described after an overview of the activities realized.

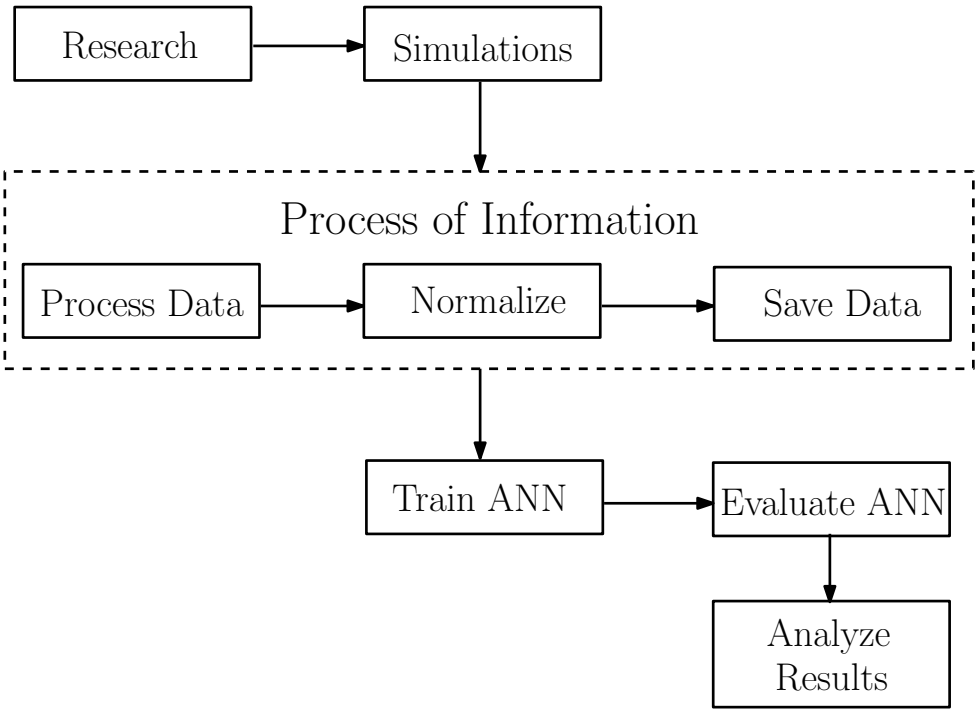


Figure 3.1: A flow chart of the activities realized.

Figure 3.1 are the activities done throughout the project and is a very reduced summary of the methodology followed. These condensed activities will be explained more as the section continues.

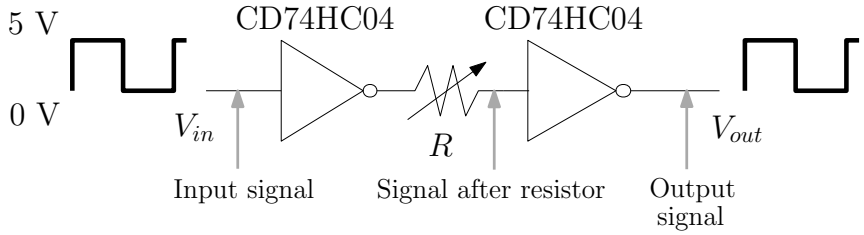


Figure 3.2: An example of a circuit simulated with the gray arrows indicating the origin of the signals.

It is important to mention that during initial simulations, the process technology is not considered at this point, only the time response of the circuit. In Figure 3.2, there is a simulated circuit with arrows indicating where the probes were placed for the transient analysis. These probes allowed the input signal, the signal after the resistor and before the second inverter, and the output signal to be visualized on a graph. It was necessary to monitor signal after the resistor since it demonstrated on many occasions that the signal was already having difficulty and losing its form.

As the simulations were produced, the process technology was decided on. After obtaining the parameters for this process technology, these were applied to the simulations in order to find the fault model that was the more realistic representation of the defect. The process technology for this project is 0.35 μm. A 0.35 μm CMOS inverter should be used for these simulations while varying the resistance values. In this case, the component is the CD74HC04 CMOS inverter and both inverters in the fault model are replaced with this component. This component and the 0.35um technology were chosen because enough information of the technology characteristics and parameters were found to be widely available. The information was required to conduct realistic simulations. The model for CD74HC04 was downloaded from Texas Instruments [25] and imported into LTspice. The model should remain without alteration and be used with the characteristics the model has. After consulting the datasheet of the CD74HC04 inverter, the max frequency can be found along with information about the threshold voltages when the input voltage varies from 0 volts to 5 volts. The improved circuit model was simulated continuously until the correct data for training an artificial neural network was gathered.

It was decided to have circuits still considered to be healthy use resistance values beginning with 1 Ω, increasing every 100 Ω until reaching 2.2 kΩ. It was decided that the biggest indicators of a fault being present began with the rise and fall times of the output signal of circuit varying far from the expected times of the input signal. These expected times included the 17 ns delay the CD74hC04 inverter already included. At this point, these circuits were working with a maximum frequency of 12.5 MHz. The transient analysis had a stop time of 330 nanoseconds and a maximum time step of 0.5 nanoseconds. In order to be able to save this data for healthy circuits, the following steps had to be taken in LTspice to save the data after doing a transient analysis:

1. After running the transient analysis, the graphed signal will be displayed in a separate tab in LTspice. Click the right button on a mouse, a window should appear and pass the cursor over “View”.

2. Then in the new options that appear, click on FFT. A new window should open to select the waveforms to be included in the FFT. The number of data point samples in time should be adjusted to match that of the stop time.

3. In “Binomial Smoothing done before FFT and windowing”, the number of points should be changed to a multiple of the stop time. After the multiple has been selected, click “Okay” to continue.

4. A new window called, “Select Visible Waveforms”, should appear and to select the signals the user must press the control button then select the signals. After the signals have been selected, click “Okay”.

5. In the new tab that appears for the FFT, repeat the second step. This time a smaller window will show up. The number of data point samples in frequency should be

changed to the stop time value and then the user may click "Okay".

6. Finally, final small window will pop up and the signals must be selected as they were in the fourth step. After another click on "Okay", the signals should closely resemble the original signals before the Fourier transformations. If the signals do not look similar, start from the first step again and vary the stop time or alter the value of the multiple.

7. Once the signals resemble their original versions, confirm the graph is the one that went through the Fourier transformations and go to "File" to save the data as a text file.

8. Wherever the LTspice schematic was saved, that is also where the text file can be found. Open this file to confirm the time step is now consistent in its advancement. If the time step still varies or has many decimals, begin at the first step again.

The data in the text file should be copied and pasted in an Excel spreadsheet. Then this information must be normalized by using the highest voltage value, which was 5 volts. The process technology functions with either 3.3 volts or 5 volts, but it was decided on 5 volts since most components and devices work with 5 volts. The spreadsheet is saved again as a text file for MS-Dos. The text file has to be in MS-Dos since the version of Scilab used does not recognize newer text file formats and it is needed to make a DAT file for the ANN to be able to use the information as vectors in Scilab.

Then, using the data acquired in the appropriate format for its use in Scilab, train the ANN until the resulting output signal from the network matches the input signal of the fault model circuit. It is expected that the output of a trained ANN should look similar to the output of the circuit. This is, the ANN would predict the output of a healthy circuit by producing a signal closely resembling that of the circuit. Since the circuit has two inverters connected, in this case, the output signal is expected to be an exact copy of the input signal, ideally. The ANN should make a similar prediction or attempt to approximate the same form as the input signal. The training should continue until both signals almost overlap entirely. While these signals should overlap for the most part, it does not signify that the signals should match each other completely. The output signal should not have too many glitches and its form should share most of the same values as the input signal. It should be noted that the ETA, the number of epochs, the number of neurons or other parameters may need to be adjusted for better results. The learning phase of the ANN may take a long time, but it may also happen in a short amount of time if the parameters are balanced accordingly.

After the training of the ANN has been completed, test the diagnosis of faults to evaluate it. This evaluation can be accomplished using the Root-Mean-Square Error (RMSE) in the program used for the ANN. Once the values have been compared, it should be noticeable which error signal belongs to a faulty interconnection and which belongs to a

healthy interconnection. The signals containing these error values are plotted against each other.

Finally, the results should be analyzed to ensure the success of the diagnosis and if any future improvements may be made.

Effect of Interconnection on Circuit Performance

As mentioned in the methodology, 0.35 μm was the process technology chosen for the project and the information came from the sites of the following institutes: University of California (Berkley), University of Lund and MOSIS. Using the resistive-open fault model as a base, the CD74HC04 inverters replaced the inverters in the circuit used in LTspice. The parameters were used to consider when the interconnection could be labeled "healthy" or without damage.

The following are the parameters for 0.35 μm technology [26][27][28]:

- t_{m1}= 665 nm
- w= 1.6 µm
- l= 1.6 -160 µm
- R_S= 0.08 Ω

where t= thickness, w= width, l= length, R_S= resistance sheet.

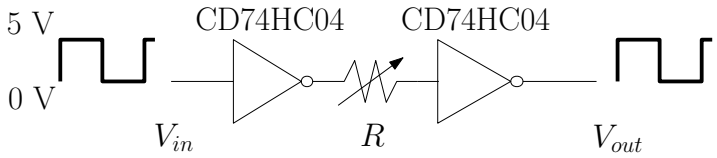


Figure 3.3: The circuit used for simulations with a variation of resistance values.

Figure 3.3 is the fault model which was applied in a circuit for simulations. Both inverters are replaced by a CMOS inverter CD74HC04 and the resistor will be varying. The resistor varying in value will be shown in Figure 3.4.

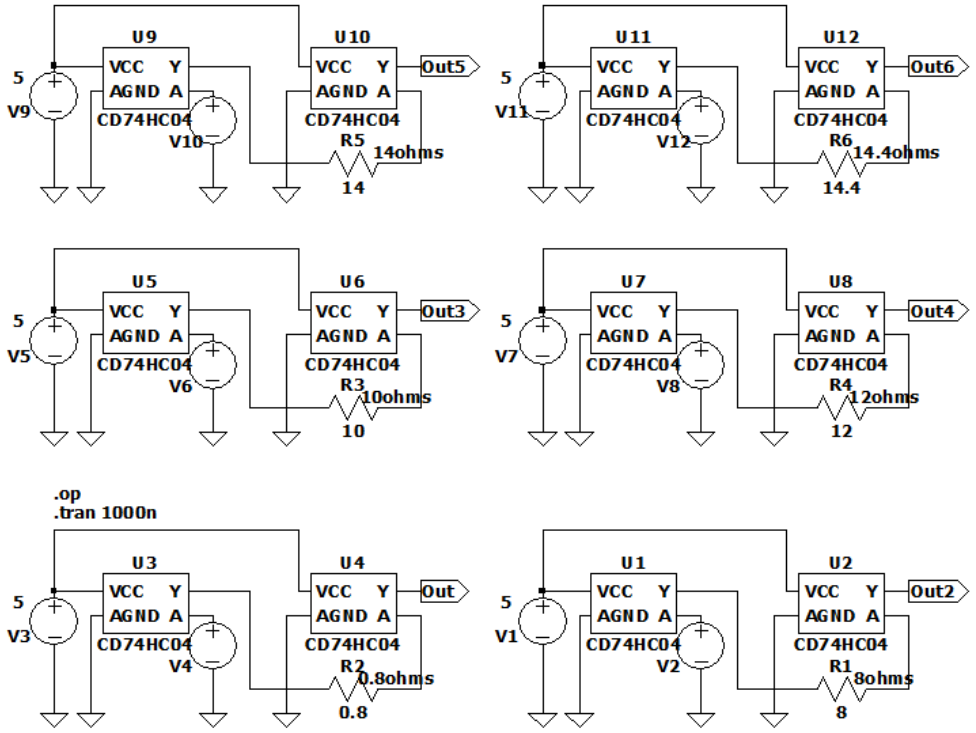


Figure 3.4: The circuits used for simulations to demonstrate the effect of small differences between slight changes in resistance.

The circuits in Figure 3.4 were used to create a simulation. The "healthiest" resistance values were 0.8 Ω and 8 Ω, two separate cases with varying lengths. The equation (3.1) is for line resistance, where n is the number of squares and R_S is the sheet resistance.

$$R_{line} = nR_S \tag{3.1}$$

An example for one of the cases was calculated as follows:

$R_{line} = (0.08\Omega)(10)$
$R_{line} = 0.8\Omega$

Considering the R_S is 0.08 Ω, it was decided to have two cases in which the circuits would be "healthy". In one case, the R_S given by the found parameters for 0.35 μm technology is multiplied by 10 squares and in another the R_S is multiplied by 100 squares. Both cases are then compared to other resistance values that represent faulty interconnection lines.

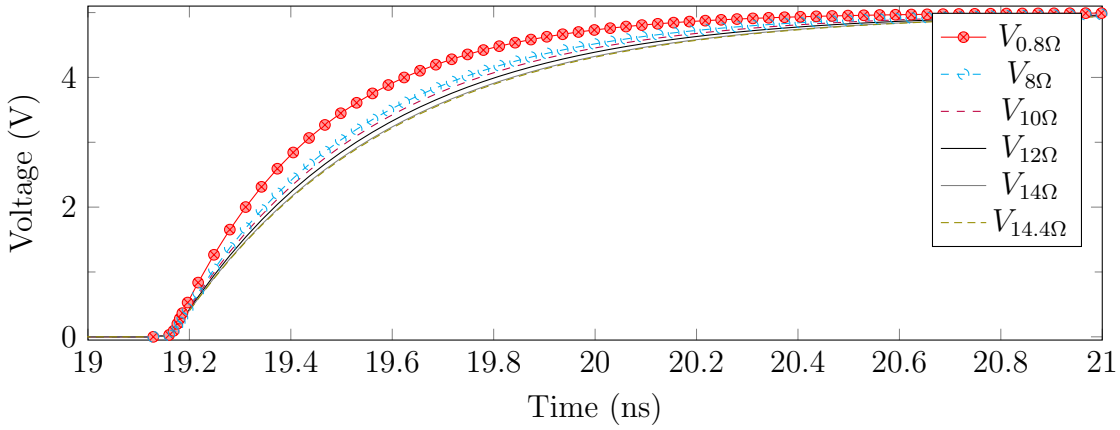
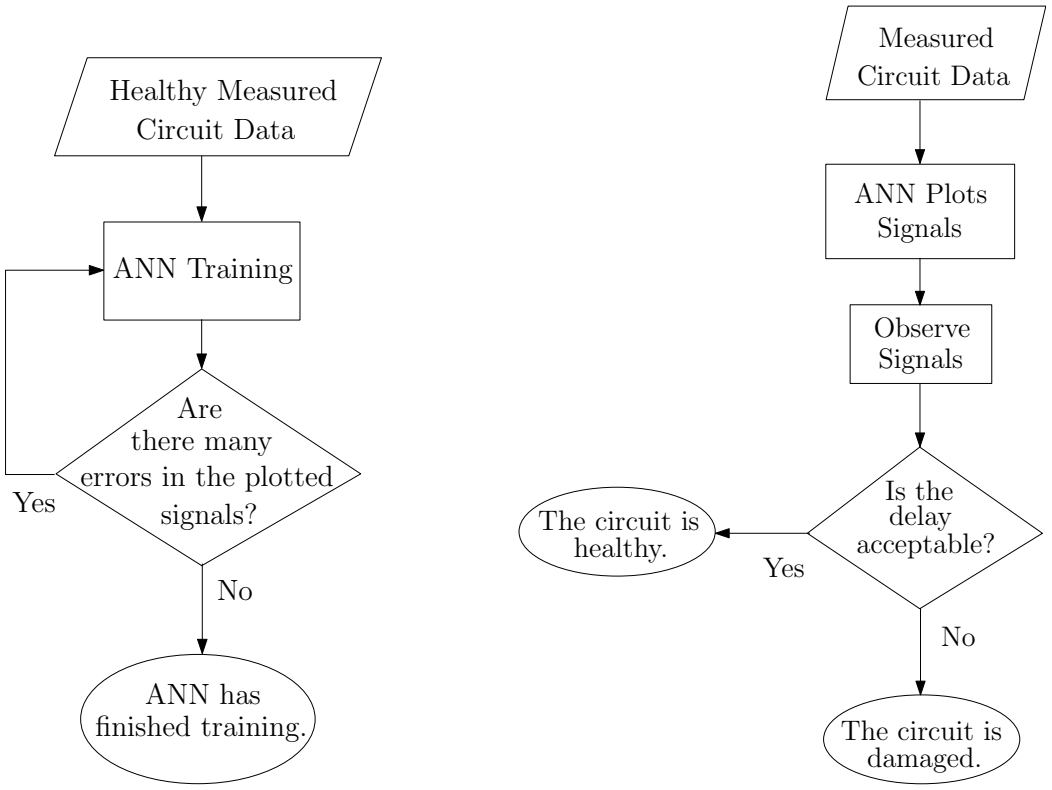


Figure 3.5: The resistive effects of the circuits in Figure 3.4.

The signals in Figure 3.5, illustrate the effect of different values for the resistance of the interconnecting line. As stated before, the signals from the circuits with resistance values of 0.8 Ω and 8 Ω are considered to be healthy interconnection lines. As seen in Figure 3.5, with increasing the resistance the signals are affected. The higher value of resistance affects the rise time and the pulse's form when comparing these signals to those of 0.8 Ω and 8 Ω.

Diagnostic of Resistive-open Faults using ANN

In order to make an artificial neural network and use supervised learning as a training method, the first task was obtaining the correct training data. In the case for this ANN, it depended on the voltage of the input signal passing through the circuit and the corresponding output signal voltage for every circuit. Each circuit had a different resistance value in the interconnection line betwen the two inverters, but all of these circuits had the same input signal. Once the information was attained, this data was transferred and saved into a DAT file which would allow Scilab to read and use the data without any hassle.



(a) ANN Training (b) ANN Validation

Figure 3.6: Flow charts generalizing the processes for the artificial neural network.

The main idea for the resistive fault diagnosis to function is for the ANN to compare between a healthy circuit and one that is faulty, which is illustrated by Figure 3.6. This was done by determining which resistance values could be considered healthy. After observing the behavior of the output signal and comparing it to the input signal of the fault model circuit that was simulated, the resistance values could be divided into "healthy" or "faulty" categories. The data saved from the simulations for the healthy circuits was used to train the ANN using supervised learning. In the learning phase, a few parameters had to be changed through several trials to find the best approximation to the data given to the ANN. The resulting output signal had to be as similar as possible to the input signal the ANN recognized from the data. Once the ANN had an output signal reach the best approximation, the training ended.

Initially, the data used was time, the input signal of the fault model circuits, and the output signal to every circuit. The neural network had five neurons in the hidden layer and only one output neuron. The ETA was 0.0001 and the number of epochs was varied until reaching 30,000, but it was clear that the resulting output signal of the ANN would not be able to become more like the input signal's waveform. With each change made to

the neural network, a new version of the code was saved. This was done to easily identify which codes had certain changes made and to find the most successful code among the rest. The final code used in the learning phase of the network had an ETA of 0.00101 and used 90,000 epochs. Not only this, but it had: two input delays, three input signals, the number of neurons was six, one output neuron and one output signal.

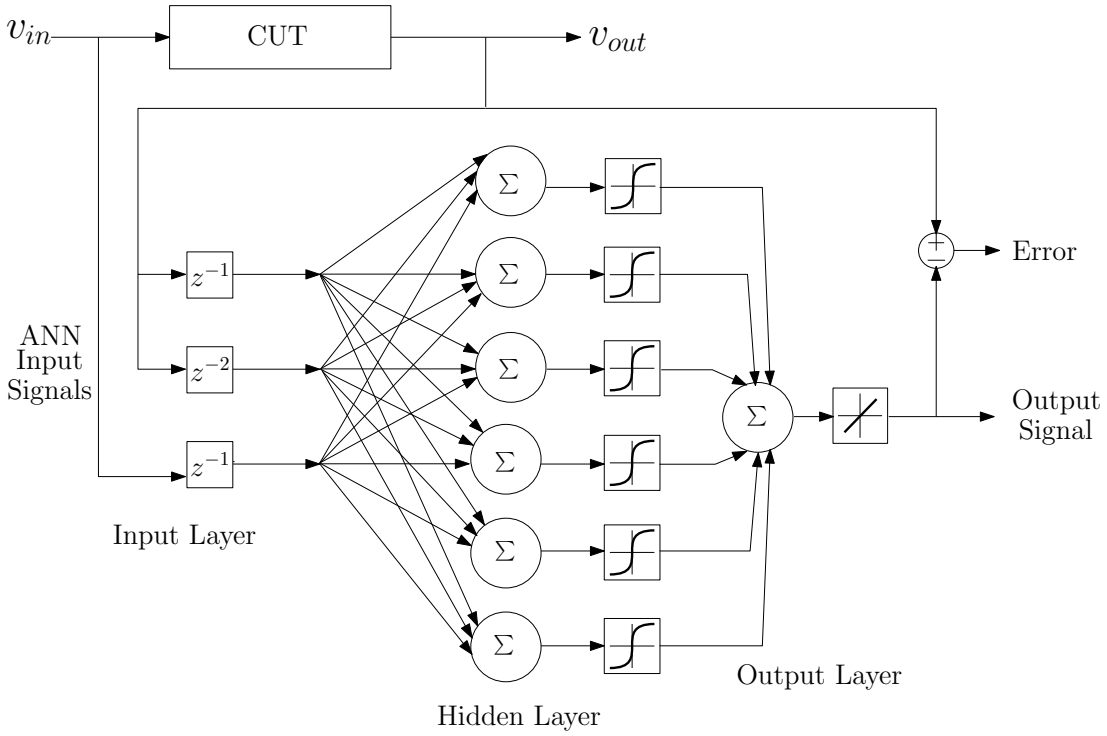


Figure 3.7: The applied artificial neural network.

In Figure 3.7, the applied ANN is presented with three input signals. The normalized output voltage (V_{out}) of the circuit under testing (CUT) going into the blocks of z^{-1} and z^{-2} are two input signals for the ANN. The block z^{-1} are the delayed samples after the original samples from V_{out}. Block z^{-2} are the delayed samples after the samples from block z^{-1} of V_{out}. The normalized input voltage (V_{in}) of the CUT follows into a separate block z^{-1} and is the third input signal for the ANN. This separate block z^{-1} indicates the samples will be delayed once from the original samples from V_{in}. The normalized voltage values are multiplied by the weights and a bias is added to each product upon passing through the neurons in the hidden layer. These values are now activated by a hyperbolic tangent function before reaching the output layer. Before reaching the output layer, the values from the activation function are multiplied by weights once more. Upon reaching the output layer, a bias is added to the products and the resulting values are summed. The output signal is obtained and an error is calculated. The output signal is the result predicted by the ANN.

The error is calculated by the subtraction of the expected output voltage defined by the CUT's data by the estimated output value of the ANN. The backpropagation algorithm uses the error to adjust the weights during the training phase.

After the training was completed, the data the ANN received came from designated fault model circuits. The same process to obtain the "healthy" circuit data was also applied to obtaining the data for faulty circuits. This data was new for the ANN and by plotting the the predicted output signals against the ones from the "healthy" circuit data, it was clear the data was very different.

3.2 Experimentation

One of the first problems encountered was finding articles relating to resistive-open faults and a test method using ANNs was not common. Another difficulty when researching was a lot of information found was not in published articles, but this information could be found in books and dissertations. Finding the information was for the most part a success, but a lot of these sources could be considered outdated.

Another issue was the program used to simulate the resistive-open fault model. Unfortunately, the first program that was used did not give the best results. This program was Multisim and it had been chosen for its various tools. It was clear Multisim would not serve the purpose of simulating faults because the signals in the oscilloscope were not simulated as expected when applying a large resistor to the circuit. It is also important to note that there were no indicators for the x-axis or the y-axis on the graphs. Also, there was no option to add an axis label or values. Only a few simulations were done with Multisim before deciding to try a different program.

The next program chosen was OrCAD as it not only had several tools, but many components as well. Simulations with OrCAD had been done before and was a better choice than Multisim. OrCAD did not meet all the requirements due to more idealistic results from simulations and not showing the output voltage. While the schematics and graphs had a better presentation than with Multisim, the results were close yet far from what was expected. There were a few technical issues with this program as well. These issues consisted of the program completely stopping or not saving the changes that had been made.

In Table 3.1, the programs tried are compared to better summarize all that was favorable and state any points lacking that were found. LTspice proved to be the better choice.

Table 3.1: Program Comparison

Program Comparison		
Multisim	OrCad	LTspice
Easy to use	Not straightforward	Extremely easy to use
Simulations with different oscilloscopes	Difficult installation	Easy installation
Not easy to graph or save results	Overly complicated	Easy to graph and save results
Graph has no values in any axis	Simulations did not give output voltage value	Simulations give output voltage value
Models can be added	Has many tools and components	Simulations demonstrated the effects in the resistor in the circuit
	Models can be added	Models can be added

Finally, LTspice was installed and it became the program for simulations. Not only did LTspice result in the expected outcomes for the circuits being used, but it was also more user-friendly and easier to customize. The customization of commands allowed for tasks to be finished quicker and be less complicated. The schematics and graphs were far superior than the previous trials obtained from OrCAD and Multisim. The graphs had the adequate values, axis labels were permitted and additional adjustments were also allowed in LTspice, unlike in Multisim. It should be noted there were no technical issues like there were with OrCAD. LTspice files were easy to save and close, so the possibility of losing any data became small.

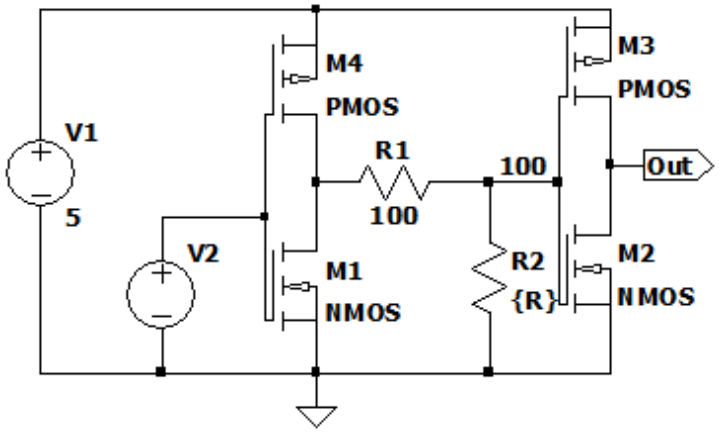
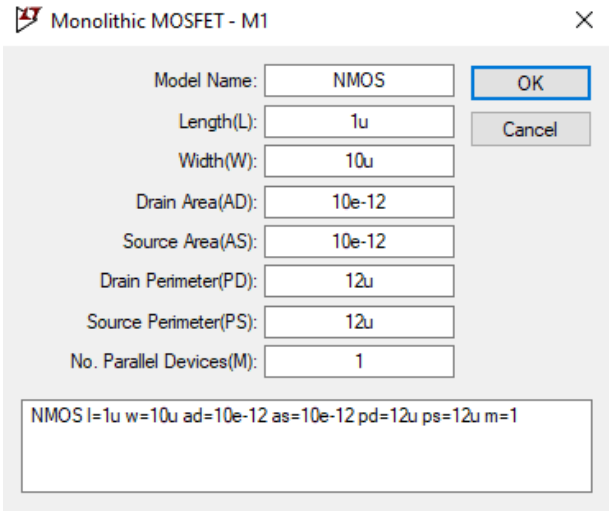


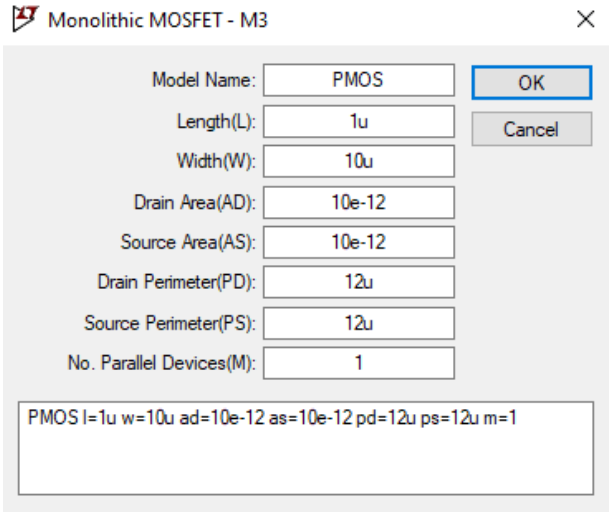
Figure 3.8: An initial attempt with MOSFET transistors.

In the initial stages, MOSFETs were used. Individual NMOS and PMOS transistors were

configured into inverters and the circuit was designed to represent the resistive-open fault [29], as shown in the example in Figure 3.8.



(a) NMOS Transistor



(b) PMOS Transistor

Figure 3.9: The parameters for NMOS and PMOS transistors [29] used in Figure 3.9.

Figure 3.9 shows the parameters for the MOSFETs. These transistors were used before the process technology was decided. Once 0.35 μm was chosen and an inverter from this same process technology was found, these inverters made of MOSFETs were no longer used in simulations.

Table 3.2: Variation of Input Signals used for Varying Resistors

Input Signals	Resistance							
00001	1	100	1K	2K	3K	4K	5K	5.38K
00011	1	100	1K	2K	3K	4K	5K	5.38K
00111	1	100	1K	2K	3K	4K	5K	5.38K
01111	1	100	1K	2K	3K	4K	5K	5.38K
10000	1	100	1K	2K	3K	4K	5K	5.38K
11000	1	100	1K	2K	3K	4K	5K	5.38K
11100	1	100	1K	2K	3K	4K	5K	5.38K
11110	1	100	1K	2K	3K	4K	5K	5.38K

Table 3.2 organizes all the input signal patterns used for some of the simulations done. The first column is an indication of the repeating signal patterns and the second column is for the variation of resistance values for the resistor in the fault model circuit. For every resistance value, there was a circuit simulated using the corresponding input signal.

Figure 3.10 is an example of a fault model circuit that had a varying input signal. These simulations with varying input signals were used to visualize differences in the affected output signals and were initially chosen to have more variation. The issue with these simulations was that the frequency also varied and so many signals would result in too much data to train the ANN, which is the reason these input signal patterns for simulations were not continued.

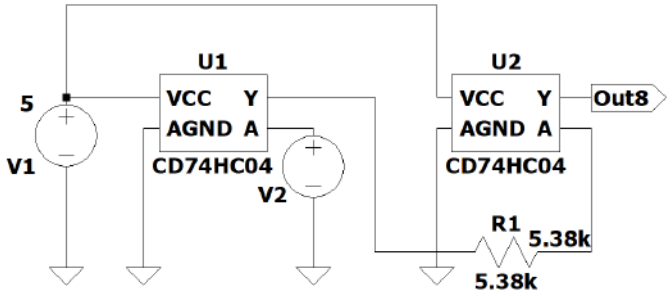


Figure 3.10: A resistive-open fault model of 5.38 kΩ with a repeating input signal of 00001.

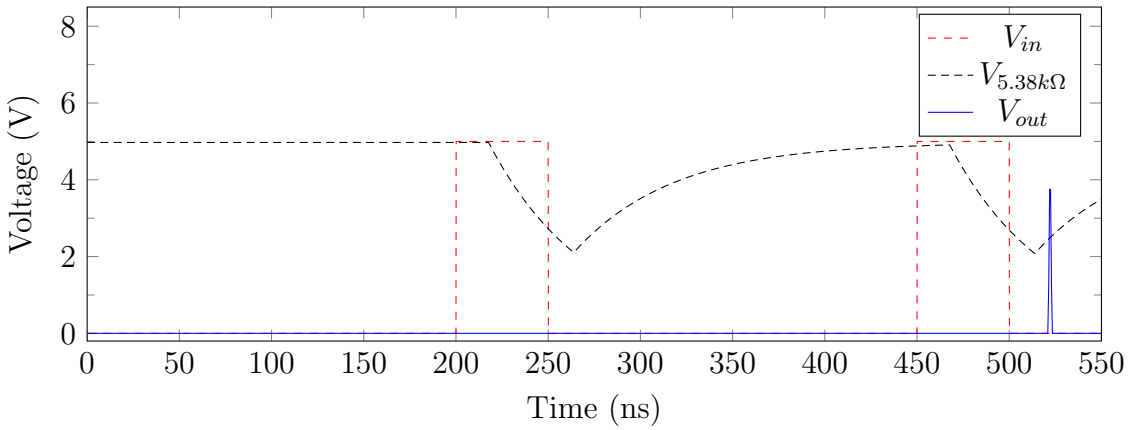


Figure 3.11: The graph for the resistive-open fault model of 5.38 kΩ with a repeating input signal of 00001.

In Figure 3.11 serving an example for one of the trials, the solid blue signal is the output signal of the entire fault model circuit and the dashed red signal is the input signal of the fault model circuit. The input signal repeats itself after every "00001". In the output signal of the circuit, it is noticeable that it contains a glitch, but otherwise remains at zero volts. The dashed black signal is the input signal of the second inverter and is also unrecognizable as it lacks the form of a pulse from the initial input signal of the first inverter.

These simulations with varying input signals were not used. When a circuit is tested, it is usually at its highest frequency. By using the varying input signals, these signals' frequencies would not be at maximum. Another issue that had not yet been resolved was that of the inconsistent time step in LTspice. The time step consistency became a priority because the artificial neural network would not be possible to train without the correct and concise data.

One main issue was figuring out the max frequency of the CD74HC04 inverter since initially it was thought to work at 10 MHz considering the circuit's family. From looking at the datasheet it was decided to use 4 MHz as the "highest" frequency in order to have more variations of input signals. A problem with this is that more variations of input signals leads to a more difficult training phase. Another problem were the results of the simulations, which were not a reliable source of data to train the ANN. The change was also brought about with the decision to work with relatively "healthy" circuits. This way, the network would have data of circuits with resistors which could still be considered as "healthy" interconnections by using their output signals. The resistors used for "healthy" circuits were chosen to be 1 Ω to 2.2 kΩ, which were based on previous results from past simulations and considering the hypothetical case done.

With the new set of data from the output signals after a transient analysis, the only issue remaining was to fix the inconsistency with the time step. In order to have a constant

increment in the time step, LTspice suggests to do two transformations of Fourier. As explained in the methodology, this would be to first do one transformation of Fourier and then do another transformation of Fourier on the results of the previous transformation. After this is done for all signals, this information can be saved in a text file, which is what was done. Since the data had a consistent time step now, all that was left to do was normalize the voltages for each output signal and the input signal.

In order to normalize the data, voltages were divided by the highest voltage value, 5 volts. The normalized data was then put into a new text file in order to be saved as a DAT file with a program made in Scilab. This simplified summoning the normalized data into the artificial neural network. From then on, it was about making adjustments to the neural network to better suit its training process. The ETA was modified, the number of epochs and even the number of input neurons was altered in order to have a more approximate response to the input signal.

Training the artificial neural network can be a lengthy period, but by observing its graphed output it gets easier to notice if something should be changed to have it work more optimally. With each modification the ANN has to undergo training and this usually means an increment to the number of epochs. The greater the number of epochs, the longer the wait for the ANN to finish training. The properties of the computer the ANN is training on may make a large difference in processing this data and training with it. The computer's specifications may influence the speed of the training phase, resulting in less wait time. It is necessary to change certain criteria such as the number of input neurons or the ETA as well, which will improve the ANN's training. After several attempts and changing the criteria as was needed, the training phase for the ANN ended when the result was a prediction similar to the input signal.

The next step was to evaluate the artificial neural network, this was done by using the error the ANN had calculated and saved. The difference being that the data for "faulty" circuits was now presented to the network in order to have a comparison. The data of the faulty circuits each indicated a resistive-open fault was present, which was helpful in deciding if the ANN was doing well when comparing the results.

Chapter 4

Results

With this chapter, the results achieved during this project will be introduced and explained. The simulations of the fault model circuit will be presented first. Then, the results from the ANN training will follow after the section on the simulations. Finally, this chapter will end with the results for the resistive fault diagnosis.

As a reminder, all simulations were done in LTspice and the CD74HC04 inverter already has a delay of about 17 ns on its own and it is taken into account.

4.1 Simulations

Simulations of a resistive-open fault serve to have an understanding of the defect's effect in an interconnection line and subsequently provided data to train a neural network to distinguish it. In addition, the simulations gave a visual representation of the behavior in order to demonstrate that it is in fact the resistive-open fault and not another fault.

The circuit in Figure 4.1 is the model for a stuck-at fault. This circuit will serve to demonstrate that the output of a circuit with a resistive-open fault may appear similar to that of the output for a circuit with a stuck-at fault. The effects that these two faults may be different, but the outcome in some cases may be almost identical and one could be mistaken for the other. The input and the output signals are shown in Figure 4.2, which has a similar output signal to one of the possible results in Figure 1.10 from Chapter 1.

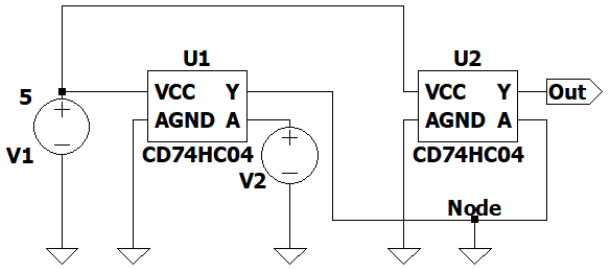


Figure 4.1: The stuck-at model for comparison.

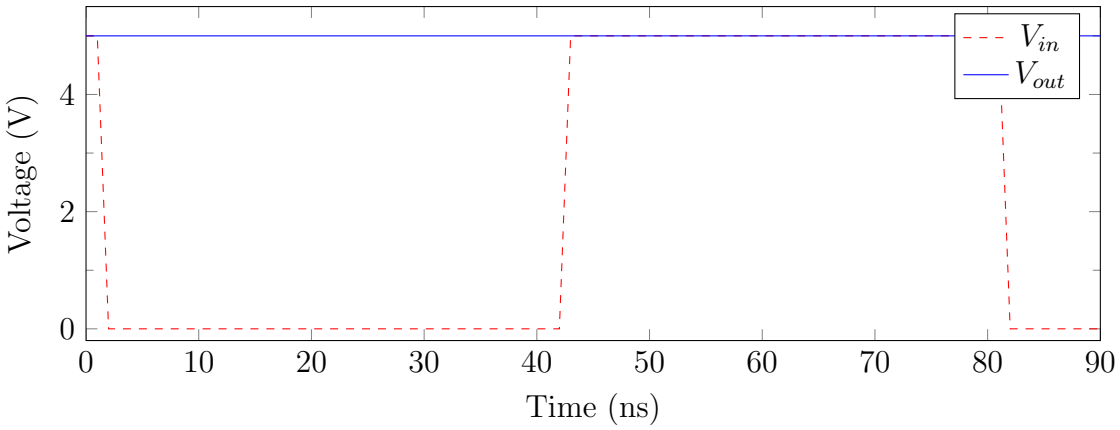


Figure 4.2: The graph for the stuck-at model for Figure 4.1.

The following simulations are those of the resistive-open fault model. These are some of the initial simulations showing the resistance values causing the worst effects on the circuits and visualizing the fault more easily. Figure 4.3 is a graph of an input signal used for the resistive-open fault model simulations.

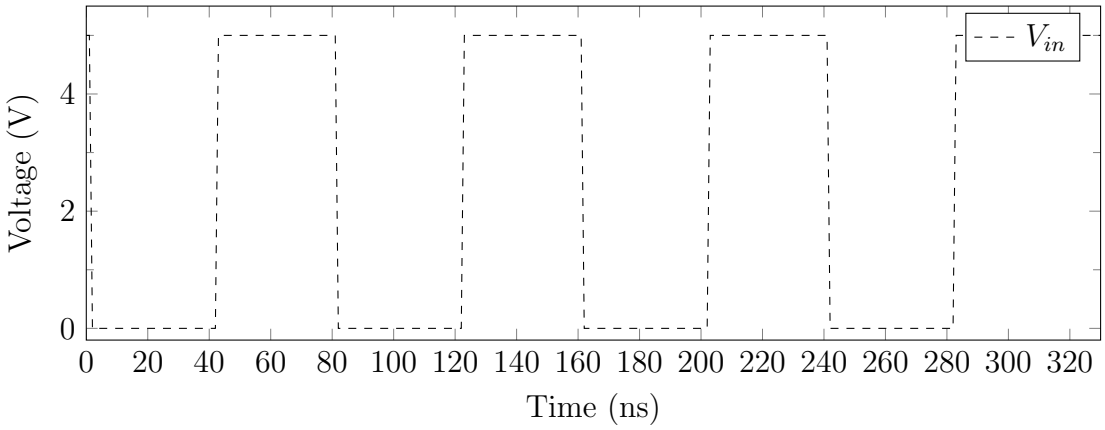


Figure 4.3: The input signal for the following simulations.

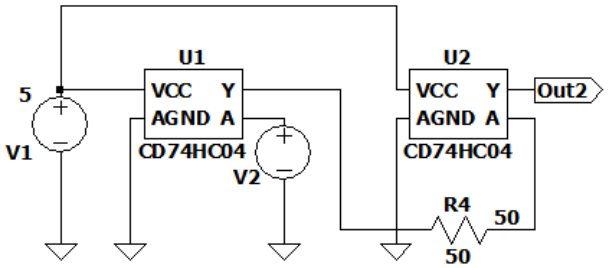


Figure 4.4: The resistive-open fault model with resistance of 50 Ω.

Figure 4.4 is a resistive-open fault model used in LTspice with a low value of resistance. The voltage source "V5" is a DC voltage supply that feeds 5V to both CD74HC04 inverters. The second voltage source "V6" is a pulse voltage supply which varies the pulse from 0 to 5V. The resistor between the output of the first inverter and the input of the second inverter, is there to represent a resistive-open defect found in the interconnection.

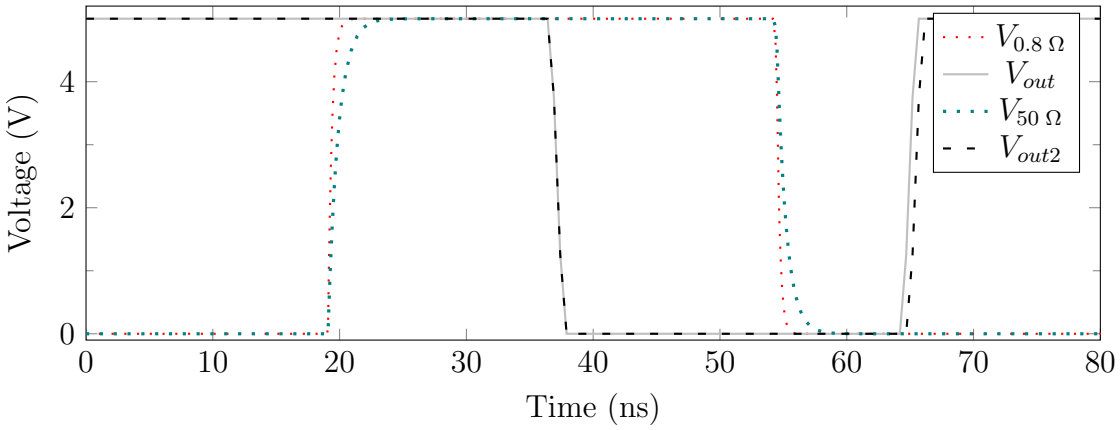


Figure 4.5: The graph for the resistive-open fault model with resistance of 50 Ω.

Figure 4.5 is the result of the transient analysis of the circuit in Figure 4.4 and a circuit with a typical healthy interconnection. The circuit with a typical healthy interconnection is the same model circuit only with resistance of 0.8 Ω (this circuit's signals will also be present in Figure 4.7 and Figure 4.9). The dotted signals are the input signals of the second inverter in each model used. The red-dotted signal is for a model with 0.8 Ω and the teal-dotted signal corresponds to 50 Ω. The difference between them is $V_{50\ \Omega}$ is becoming affected in its form, while $V_{0.8\ \Omega}$ keeps its rectangular pulse form and is not affected by a delay. The parasitic capacitance from each gate in the first inverter charge the gates in the second inverter. The delay is caused by the resistance created by the drain and source, which are now included. As the resistance grows, not only will the signal input of the second inverter be affected, but the output signal as well.

The output signal of the second inverter for 50 Ω is the black-dashed pulse. The output signal V_{out2} and the output signal V_{out} (solid gray signal) initially do not seem to have a noticeable difference. Yet, there is slight delay at around 64 ns when V_{out2} does not share the same rise time as V_{out}.

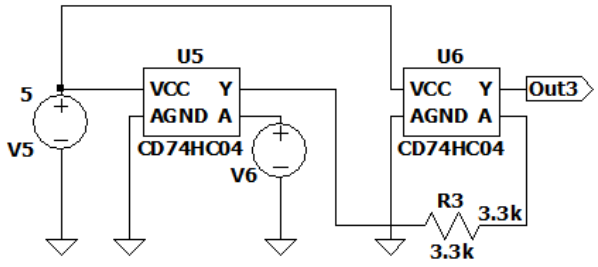


Figure 4.6: A resistive-open fault model with resistance of 3.3 kΩ.

Figure 4.6 presents another fault model circuit with resistance of 3.3 kΩ. In this circuit, the resistance is of a higher value than the one seen in the past circuit in Figure 4.4. The

resistor of 3.3 kΩ is the only change made to the fault model. The input pulse signal for the first inverter in the 3.3 kΩ model is also the same as in Figure 4.3.

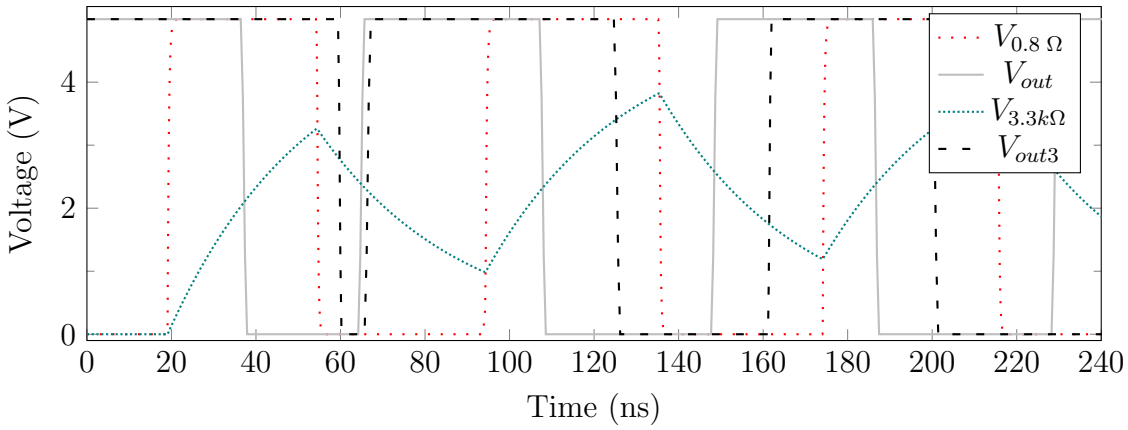


Figure 4.7: The graph for the resistive-open fault model with resistance of 3.3 kΩ.

The same colors and styles of the lines representing the signals described in Figure 4.5 describe the signals in Figure 4.7. The exceptions are the signals for the input of the second inverter and the output of the fault model with 3.3 kΩ. The input of the second inverter for the fault model is the teal-dotted signal. The output to the fault model is the black-dashed signal.

The input signal of the second inverter ($V_{3.3\ k\Omega}$) is affected more than seen previously as the pulse's form becomes saw-like. In the output signal V_{out3} of the circuit, the pulse width of each positive pulse is wider than the pulses of the output signal (V_{out}) of the model with 0.8 Ω. By the rise time of the second pulse of V_{out3} at approximately 65 ns, the delay is of one or two nanoseconds. In the rising time for the pulse at 160 ns, V_{out3} is delayed by about 12 ns. The CMOS inverter CD74HC04 has a delay around 17 ns which is part of the component, but this is with an ideal interconnection and not a typical intrinsic resistance value as is compared in these simulations. If compared to the ideal case, the output signal V_{out3} will have a delay of 29 ns.

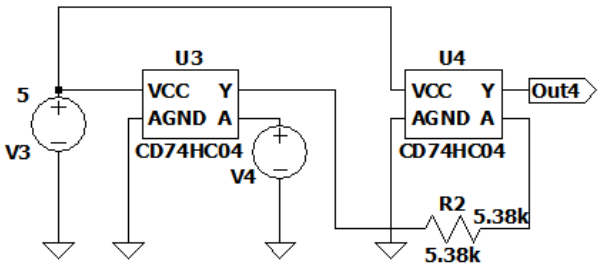


Figure 4.8: A resistive-open fault model with resistance of 5.38 kΩ.

By replacing the resistor with another that has a value of 5.38 kΩ in Figure 4.8, the fault model circuit is affected more. This fault model circuit is the most affected and the results from its simulations are shown in Figure 4.9.

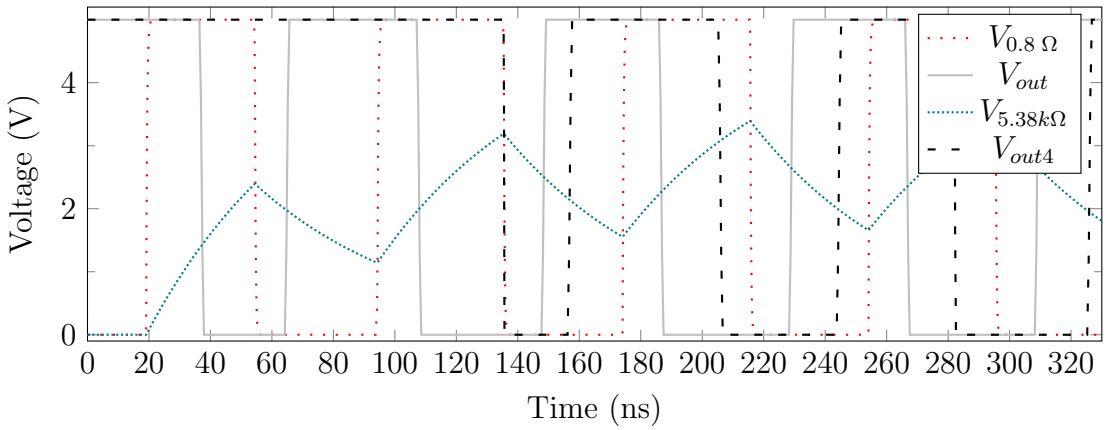


Figure 4.9: The graph for the resistive-open fault model with resistance of 5.38 kΩ.

Figure 4.9 displays the same colors and styles used for the signals in Figure 4.5 for the 0.8 Ω model's signals. The change comes with the second two signals. The input signal to the second inverter with a resistance of 5.38 kΩ is the teal-dotted signal and the output signal for this fault model is the black-dashed signal.

The output signal (V_{out4}) is delayed. The initial spike in the input signal ($V_{5.38k\Omega}$) did not reach the threshold voltage to make the switch to the state it should. It was calculated from the data sheet of the inverter CD74HC04, that in order to have logic one the voltage must be above 3.3 V approximately. For a logic zero, the voltage should be below 3.2 V. The input signal ($V_{5.38k\Omega}$) of the second inverter has lost the pulse's rectangular form due to being affected by a the large resistance, which means (V_{out4}) having a larger delay as previously mentioned.

As previously mentioned in the past chapter, once the time step issue was resolved, new simulations were made to gather acceptable information for the ANN to be able to determine the adequate circuit behavior. After having defined up to which resistance would be considered healthy for the circuit's interconnection, the simulations began. In testing, components are usually tested at their maximum conditions. In this case, the maximum frequency chosen to work with is 12.5 MHz. The resistors considered healthy start at 1 Ω, go up to 100 Ω and continue increasing by a 100 until reaching 2.2 kΩ.

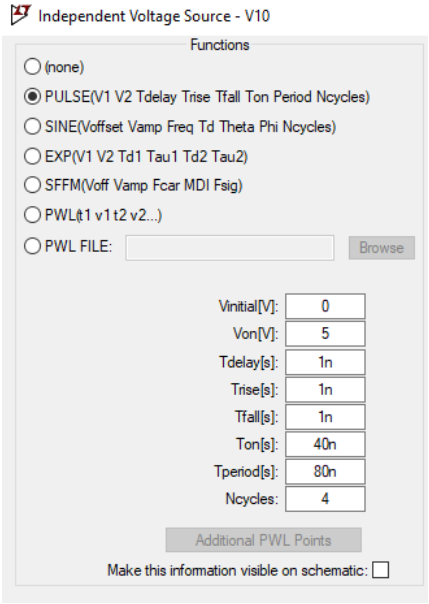


Figure 4.10: The pulse voltage supply used in LTspice.

The conditions for the transient analysis were decided on the information that was most important to the data the needed to be gathered. Such as, having enough cycles to determine there was a fault or that the delay's effects were notable. It was crucial the signals maintained their shape as a pulse, despite working at a maximum frequency. The conditions for the input signal's pulse voltage supply are in Figure 4.10, while the conditions for the transient analysis shown in Figure 4.11.

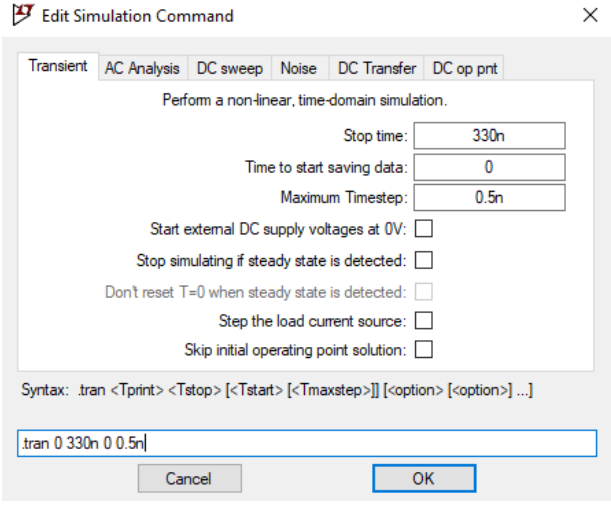


Figure 4.11: The conditions set for the transient analysis.

As a final note, the increase in resistance due to the unknown defect is provoked by the capacitors in the first inverter in each gate that have to charge the second inverter. Due to the values of the drain and source being included, a resistance is formed. The simulations allowed the fault model to recreate the effect of a resistive fault by demonstrating the increase in the delay as the resistance also increases.

4.2 ANN Training

The results from the learning phase of the artificial neural network initially were not near to the expected prediction. The expected result was a signal with a waveform similar to the waveform of the one of the output signals used for the healthy circuit database. There were some adjustments made to the code in order to get better results, as mentioned in the past chapter. There have been several changes from when the training began, some of which are as follows: the ETA, the number of epochs, and the number of input neurons.

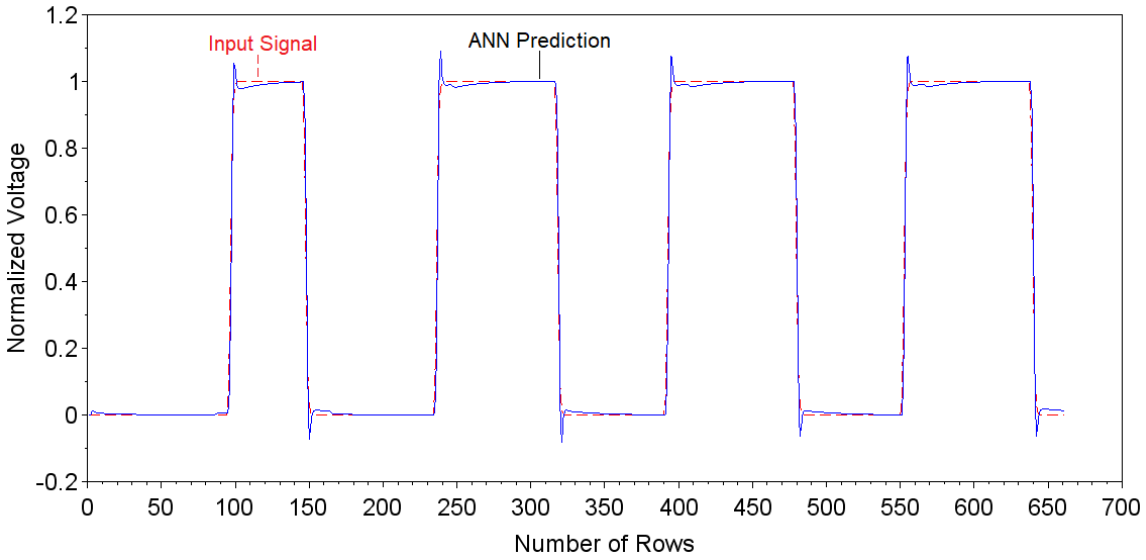


Figure 4.12: An approximation with 5 neurons in the hidden layer and 500 epochs.

In the Figure 4.12, the dashed line signal is the output signal for one of the healthy circuits that was used in LTspice. Training was conducted using a simulated circuit with an interconnection, which had resistance of 1 Ω. The solid line signal is the artificial neural network trying to predict the output signal after comparing it to its sample data of healthy circuits. It is shown clearly that the ANN still has to be trained more to accomplish getting a better approximation of the waveform of that output signal.

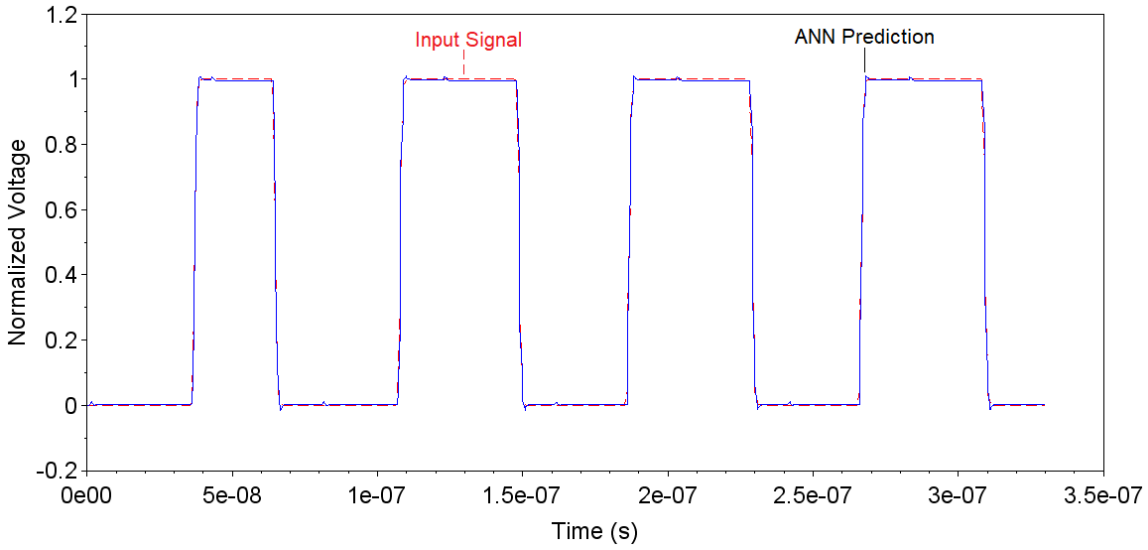


Figure 4.13: An approximation with 6 neurons to the hidden layer and training with 90,000 epochs.

Figure 4.13 only has a slight variation in its waveform when compared to the output signal of the model with 1 Ω. Once more, the dashed signal is the output signal for one of the healthy circuits and the solid line signal is the output predicted by the ANN. Considering the model used in simulations with two inverters connected by a resistor, the predicted signal of the ANN had to be almost exactly the same as the input signal of the circuit. The "healthy" circuits were chosen from 1 Ω to 2.2 kΩ because these models resulted in acceptable delayed rise and fall times, while already including a 17 ns delay from the chosen CMOS inverter. This result was decided to be superior to the past attempts and was chosen to be the conclusion to the learning phase for the ANN.

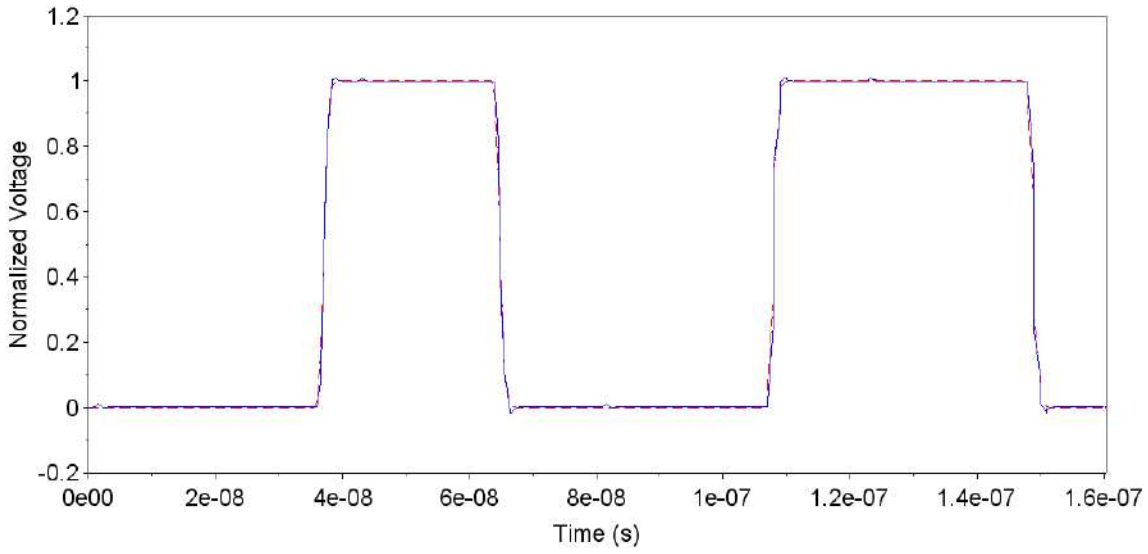


Figure 4.14: A closer look at the results of the previous graph.

Considering the proximity of the signals, Figure 4.14 is a zoomed in version of Figure 4.8. It is easier to spot a few more differences between the output signal of one of the simulated fault models and the ANN's predicted output signal. Yet, both signals overlap far too well for there to be an outstanding difference.

4.3 Diagnostic of Resistive Faults

After concluding the learning phase for the artificial neural network, the next step was to test the network. This was done using the calculated errors from the healthy circuits and compared to in a plot to faulty circuits.

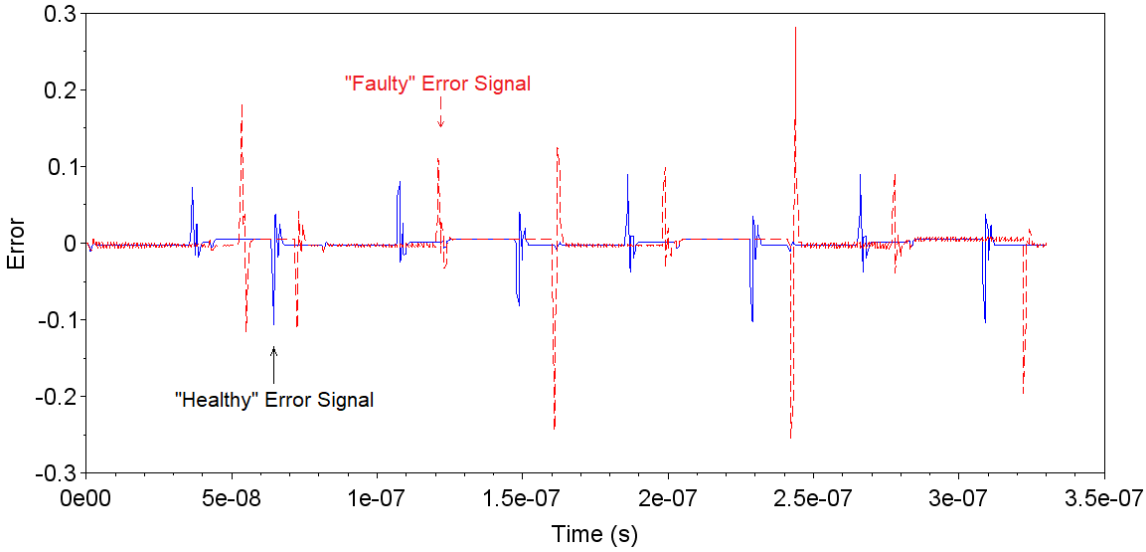


Figure 4.15: A graph of both errors taken from healthy circuits compared to faulty circuits.

The graph in Figure 4.15 has two signals. The signal with the dashed line is the error signal for one of the faulty models, while the solid line signal is for the error signal of one of the healthy models. The faulty error signal also happens to be delayed, which is an indication of the resistive fault being present. There are larger spikes for the faulty circuit data indicating an even larger difference between the faulty model's output values and those of the ANN's predicted output. While the spikes in the error signal for the healthy data are smaller, ranging from an error of approximately -0.12 to 0.1. There is still a difference between the real output values of the "healthy" model and the ANN's predicted output values, yet this difference is the limit of the error that is acceptable.

The RMSE was calculated using the error values taken from the network. This information was then used to calculate the Root Mean Square Error (RMSE) in Excel. By following the formula, it was simple to do using a few commands for Excel. There was a great amount of data, but it was necessary to have the actual value of the RMSE, which resulted in 0.083749.

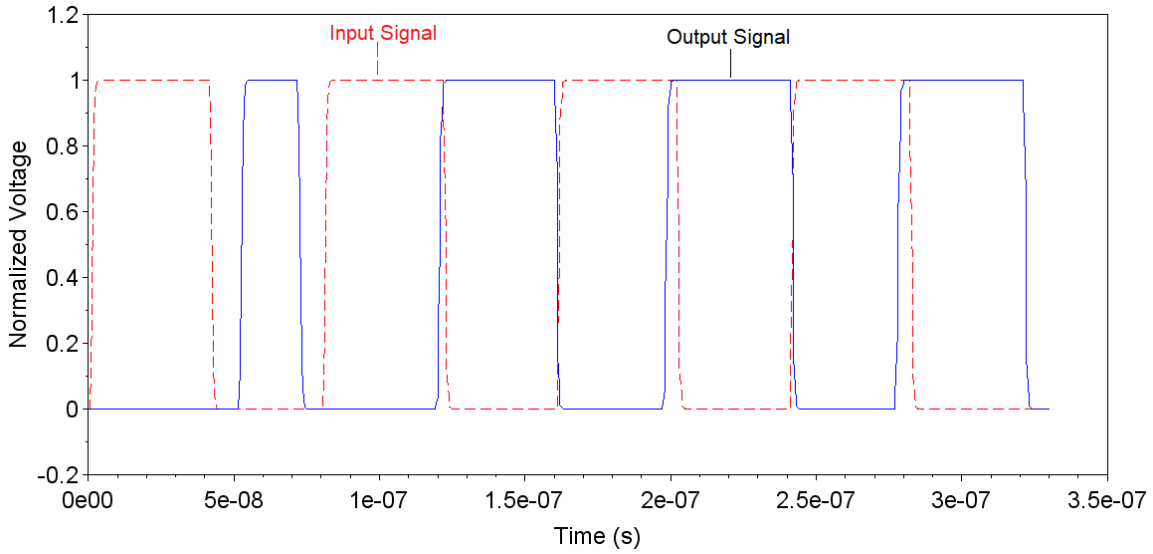


Figure 4.16: A graph of the input signal and the output signal for a faulty circuit with 2.3 kΩ.

In Figure 4.16, the dashed signal represents the input signal for the resistive-open fault model. The solid line signal is the output signal of a fault model considered to be a circuit with a defective interconnection. The fault model circuit is simulated by using a resistor with 2.3 kΩ. The delay the output signal shows when compared to the input signal is large enough for the output signal to look like the inverted version of the input signal. Yet, it is clear the pulse width is being affected as well.

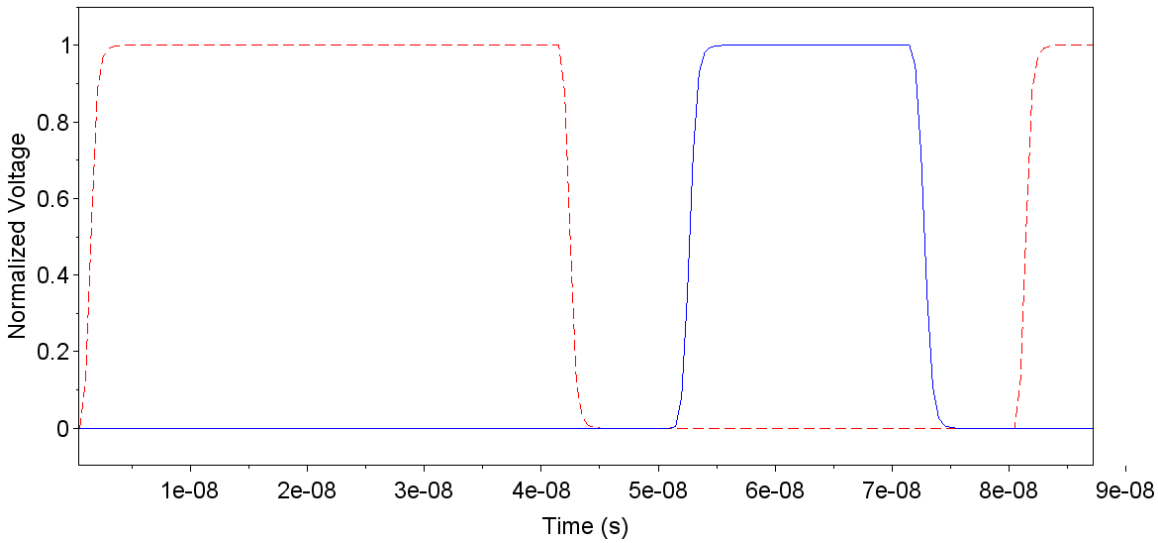


Figure 4.17: A zoomed in version of Figure 4.16.

By zooming in to the first pulse for both the input and output signals, as is done in Figure 4.17, the delay is more observable. The inverter CD74HC04 has a delay that is about 17 ns. The output signal for the faulty circuit of 2.3 kΩ is rising at around 52 ns, which means this signal has a delay of 35 ns.

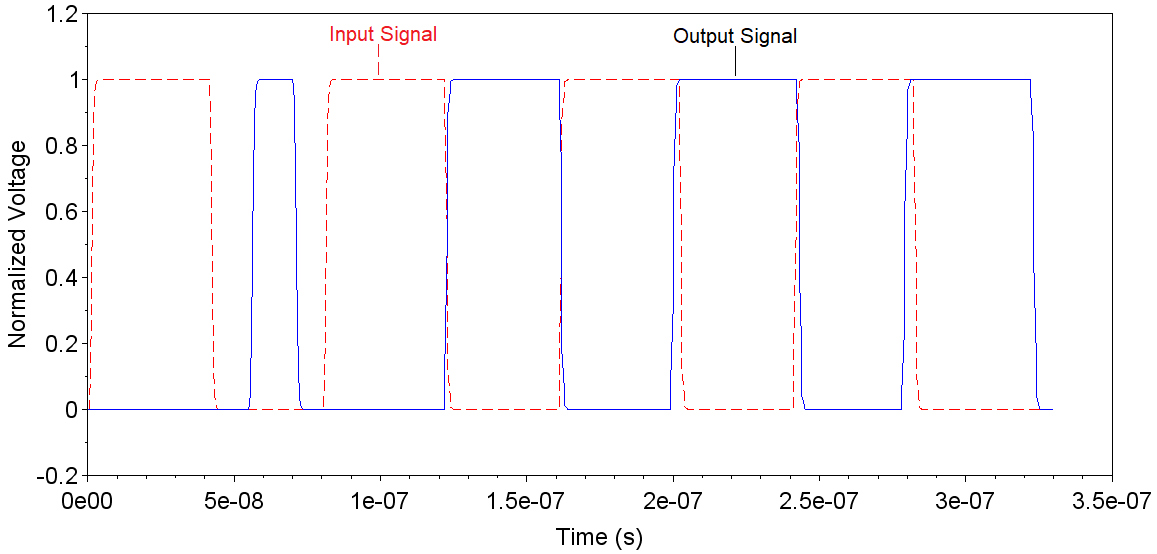


Figure 4.18: A graph of the input signal and the output signal for a faulty circuit with 2.8 kΩ.

The input signal is represented by the dashed line and the solid line represents the output signal for the fault model of 2.8 kΩ. When comparing Figure 4.18 to Figure 4.16, there are a few points to be addressed. The first pulse has a width that is getting thinner as the resistance grows, Figure 4.18 having the thinner pulse width out of both.

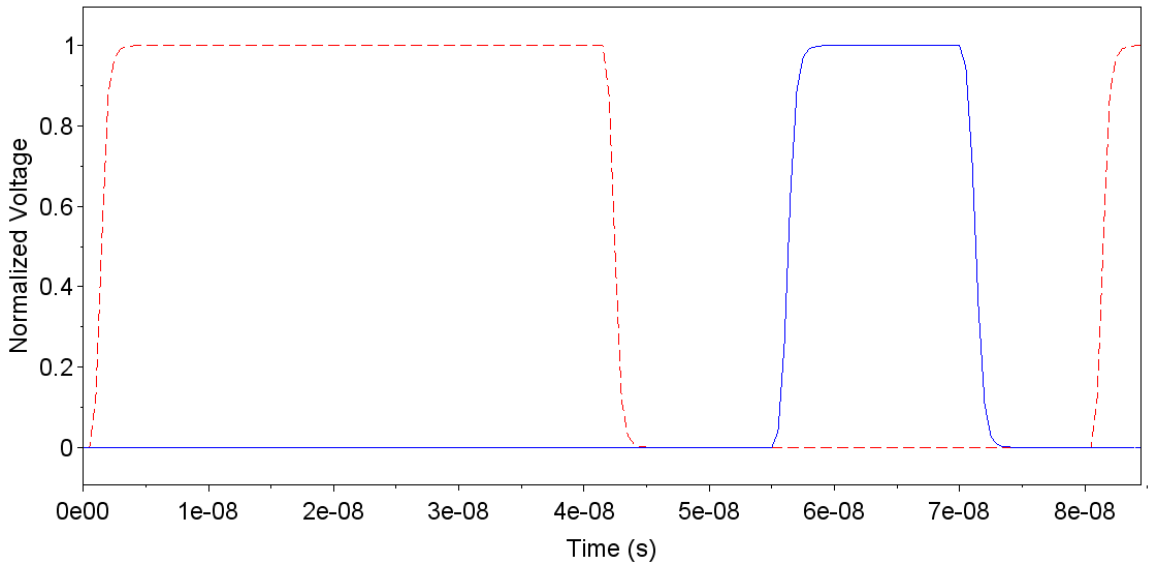


Figure 4.19: A zoomed in version of Figure 4.18.

If it is looked at more closely, once more the delay and the differences between the input and the output signals may be appreciated better in Figure 4.19. The delay is also present in Figure 4.18, only this time the pulse rises at about 55 ns. After subtracting the 17 ns delay of the inverter, the signal has a delay of 38 ns.

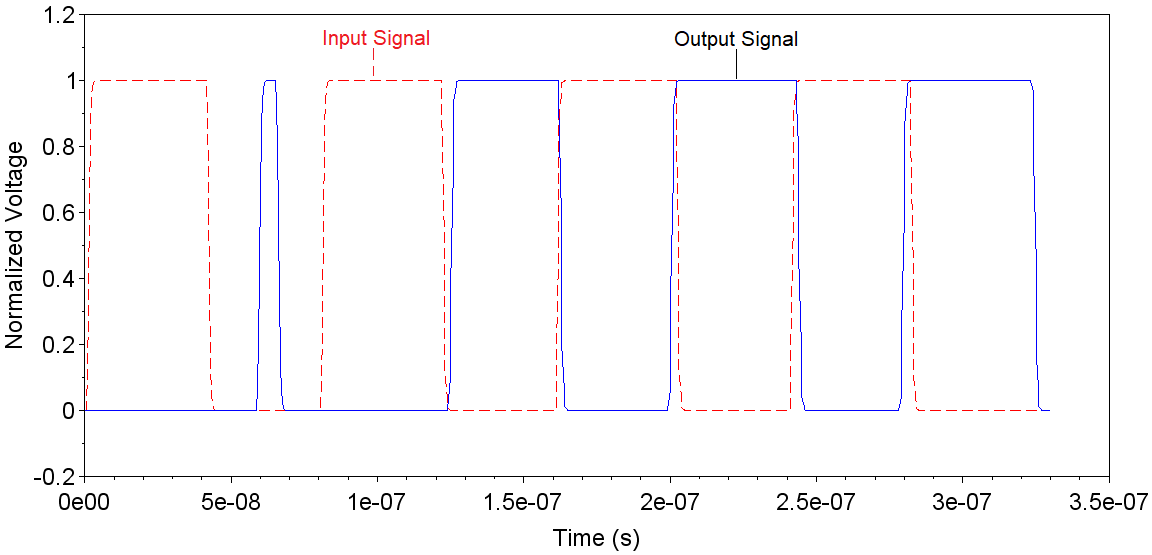


Figure 4.20: A graph of the input signal and the output signal for a faulty circuit with 3.3 kΩ.

As stated previously, the dashed line is the input signal and the solid line is the output

signal, this will be true for Figure 4.18 until Figure 4.25. The pulse width for the output signal of a faulty circuit with 3.3 kΩ, is the narrowest that has been observed.

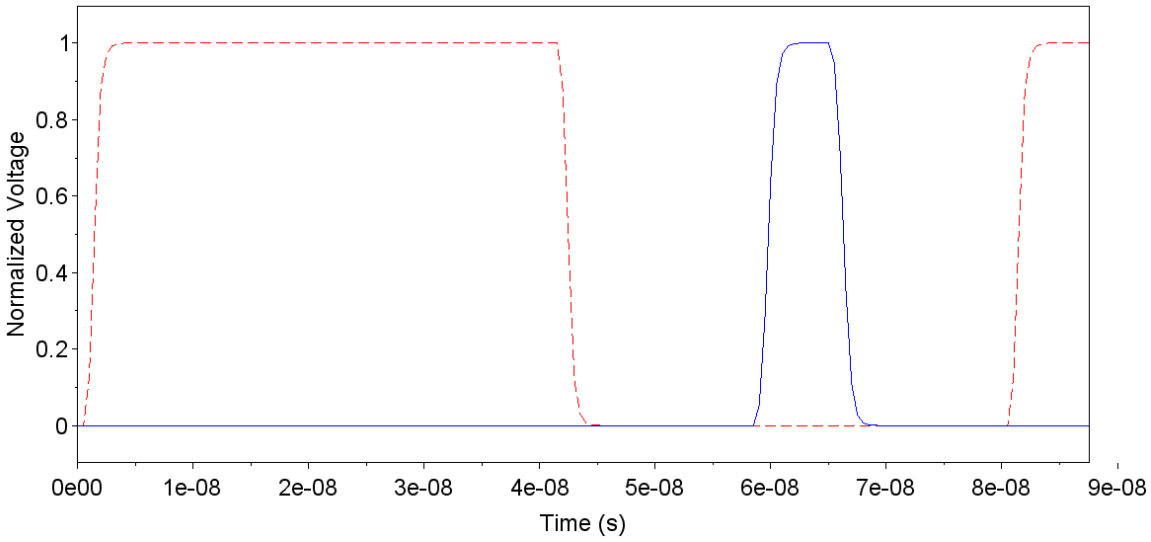


Figure 4.21: A zoomed in version of Figure 4.20.

In Figure 4.21, it can be estimated that the output signal for the fault model with 3.3 kΩ rises at 58 ns. This output signal is delayed by 41 ns, after removing the circuit's 17 ns delay. Once more it is clear that as the resistance increases there is a larger delay and the waveform is also affected more.

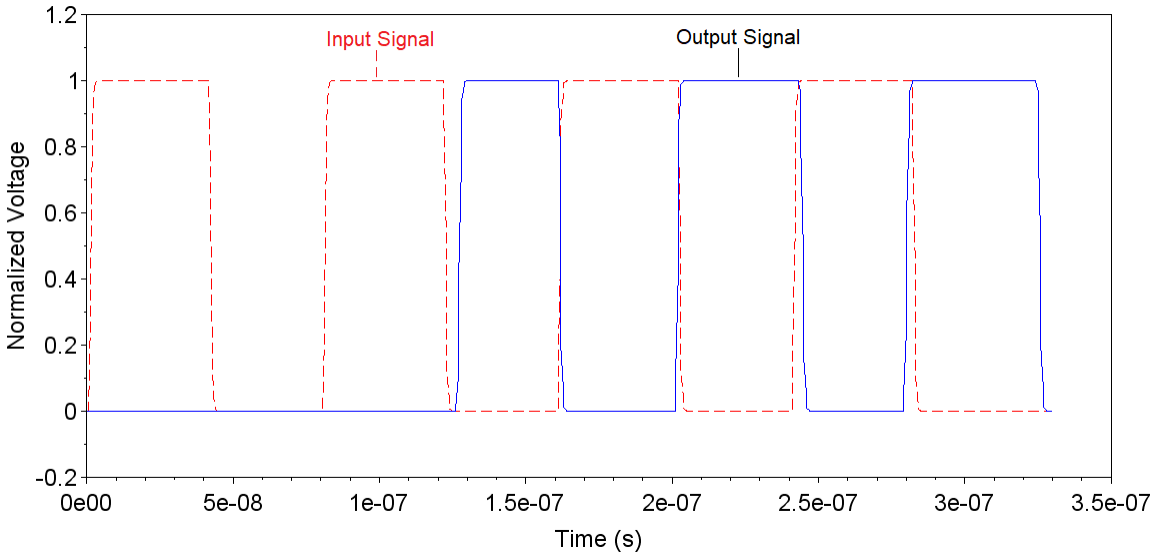


Figure 4.22: A graph of the input signal and the output signal for a faulty circuit with 3.8 kΩ.

Figure 4.22 demonstrates the output signal of a fault model with 3.8 kΩ, as being out of phase as the initial pulse seen in the past graphs (Figures 4.16 to 4.21) is gone. The narrowing first pulse can no longer meet the threshold voltage for the second inverter to result in a Boolean value of 1, otherwise known as a "high" state, and this results in the the output signal being out of phase with the input signal completely.

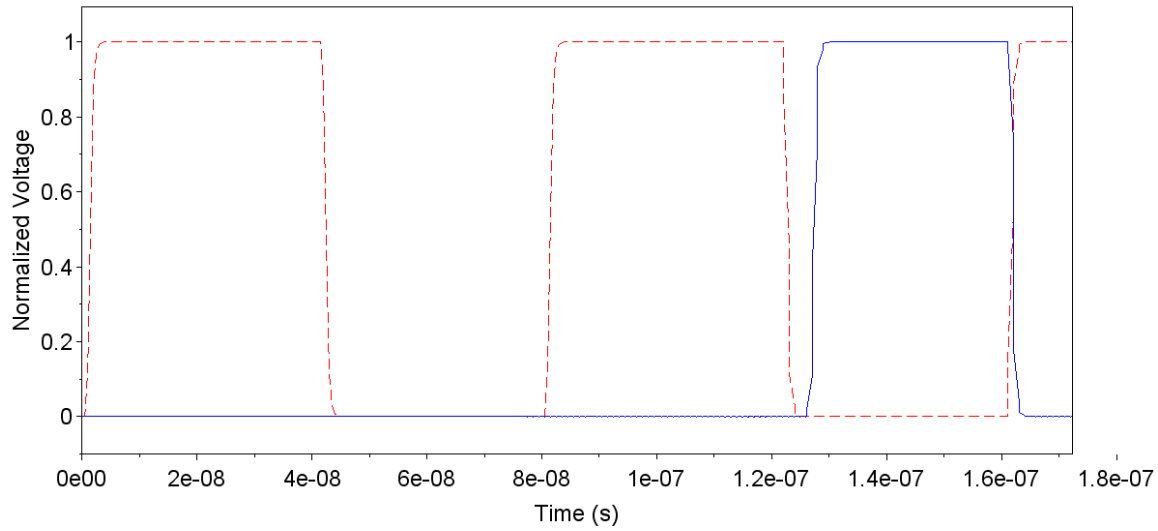


Figure 4.23: A zoomed in version of Figure 4.22.

In Figure 4.23, the output signal rises nearly at 125 ns, later resulting in a delay of 108 ns once the 17 ns delay of the inverter have been subtracted. The delay is so large due to the increase of the resistance to 3.8 kΩ, proving that the increase in the resistance results in a greater delay and being a clear example of a resistive-open fault.

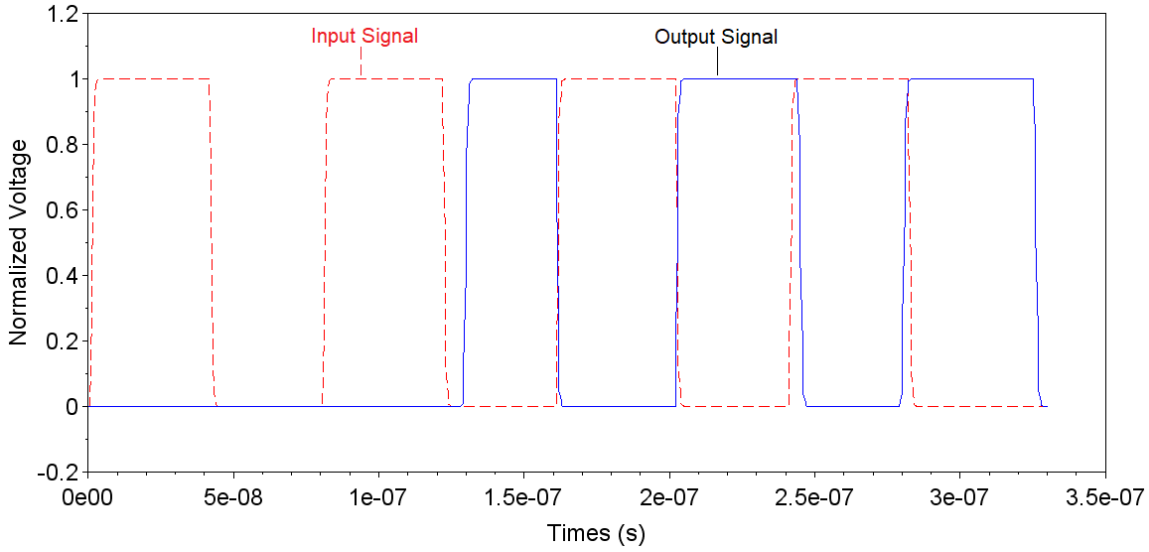


Figure 4.24: A graph of the input signal and the output signal for a faulty circuit with 4.3 kΩ.

When comparing the output signals in Figure 4.22 and Figure 4.24, the pulse width is almost the same with little difference. Again, these output signals are out of phase when compared to the input signal. Using a resistor of 4.3 kΩ in the fault model, Figure 4.19 has the output signal that confirms the continuation of an increase in the delay, which is best observed in Figure 4.25.

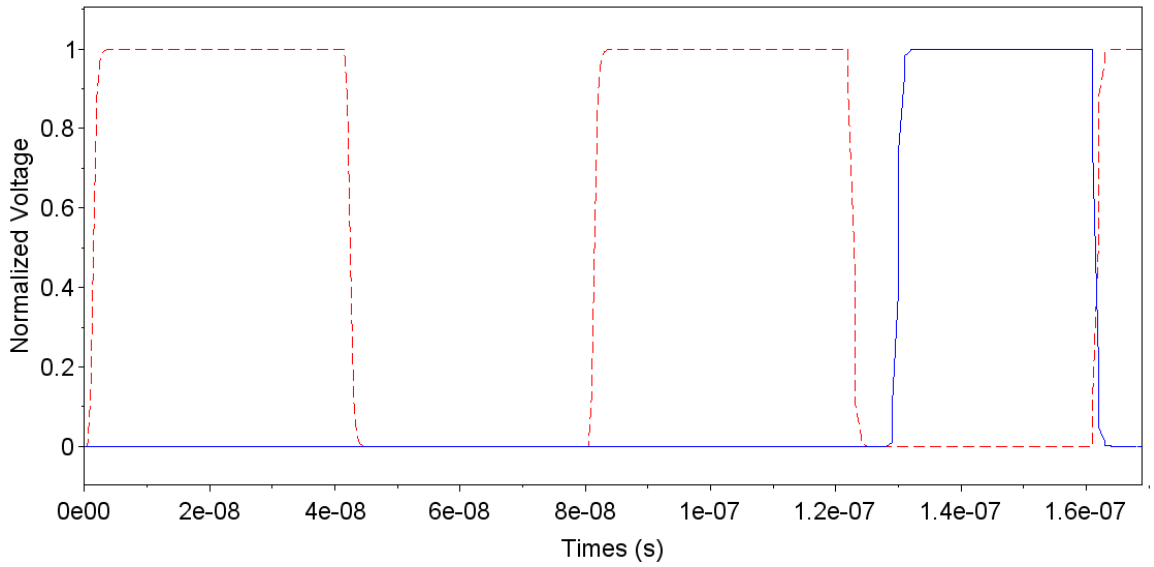


Figure 4.25: A zoomed in version of Figure 4.24.

In Figure 4.25, the rise of the output signal is visible around 129 ns. If the 17 ns delay is removed, the final delay is 112 ns for the output signal. Once more if the resistance is raised, the delay will also have an increment.

Chapter 5

Conclusions

A method for the diagnostic of resistive faults applying neural networks was applied on an interconnection between two inverter gates. Research was conducted in order to approximate realistic results. Deciding on a process technology and consequently an inverter from the same technology, made the simulations more realistic. It was very important to understand the inverter chosen and its data sheet information. The data sheet was the main indicator that the inverter CD74HC04 already had a delay that was to be expected when used.

The gathering of the adequate data for the ANN took up the most amount of time, since it involved a lot of simulations and processing the resulting data. Considering that there was a time step issue with Ltpsice, it may be better to try a more advanced simulator. This could result in a quicker collection of data for a future project and more accurate data to train the ANN.

The importance of a fault model is to copy the effect of a defect that may be found in an integrated circuit, it is a representation that accurately defines the defect. In this case, the resistive-open fault model is used for defects that could affect an interconnection line. Since these defects are not avoidable, it is best to be able to find them quickly. Using an ANN such as this, it could predict whether there is a design flaw before it appears during fabrication. For testers, it may help improve on methods that are not absolute and reduce time with the ANN's response time.

The consequences of having a resistive-open defect may affect the integrity of the signal, as shown in the past figures for faulty circuits of varying resistance values in Chapter 4. The main goal in manufacturing new integrated circuits is to process information faster, but if a resistive-open defect is present it will be an impediment. Having a neural network that can indicate the presence of a resistive-open would be beneficial to those involved in manufacturing or testing integrated circuits.

Considering a neural network can be designed to handle larger amounts of data, it may be trained to diagnose resistive faults for other process technologies besides the chosen 0.35 μm technology. The technology used in this project is older, but it served to test the

concept of this method. In future work, the models simulated may include capacitance or inductance to best represent a faulty interconnection line. This diagnostic may also expand to test for more than one fault in a single ANN. These options for future projects would prove to be a more complicated projects, but would be possible with more time.

This method for the diagnostic of resistive-opens would benefit from having a proper interface. The results currently must be analyzed from the graphs given by the ANN directly and it would be better if the ANN could also state if there was a resistive-open defect found or if the circuit was healthy. A few other characteristics could be added if this diagnosis was further advanced as previously suggested. One of these characteristics being the process technology stated after the ANN has been trained with the proper data. If the diagnostic were to be combined with a diagnostic for another fault, the ANN stating which defects were found present would be an important characteristic to show as well. The list for additional details may continue as the diagnostic is enhanced.

Due to the pandemic, access to a laboratory was quite limited and shipping delays resulted in not testing the diagnostic on standard testing circuits as initially planned. In future investigations, the ANN should be tested with these standard testing circuits to have more accurate results. It is recommended to continuously investigate process technologies and any new articles that may bring information on advancements on methods to diagnose resistive-opens. While this project may be improved and expanded on, despite the pandemic limiting the advancement of the project, it was still successful.

Bibliography

[1] M. L. Bushnell and V. D. Agrawal, *VLSI Testing Process and Test Equipment*. Springer, 2002.

[2] AnySilicon, "My golden rule for chip production testing." accessed 2021-09-29.

[3] F. A. F. Ramirez, *Development of New Fault Models and Test Methodologies in Nanometer Semiconductor Technologies*. PhD thesis, INAOE, 2020.

[4] J. P. Uyemura, *CMOS Logic Circuit Design*. Kluwer Academic Publishers, 2002.

[5] L. I. R. Gómez, *Machine Learning Support for Logic Diagnosis*. PhD thesis, Institut für Technische Informatik der Universität Stuttgart, Stuttgart, Germany, 2017.

[6] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall,, 2009.

[7] R. J. Baker, "Cmos circuit, design, layout and simulation./r. jacob baker. stuart k. tewksbury and joe e. brewer, series editors," *3rd. Edition.–IEEE Press Series on Microelectronic Systems*, 2010.

[8] M. J. Howes and D. V. Morgan, "Reliability and degradation: semiconductor devices and circuits," *Chichester*, 1981.

[9] D. Arumi, R. Rodriguez-Montanes, and J. Figueras, "Experimental characterization of cmos interconnect open defects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 123–136, 2007.

[10] P. Maqueda and J. Rius, "Analysis of the extra delay on interconnects caused by resistive opens and shorts," in *2009 15th IEEE International On-Line Testing Symposium*, pp. 208–209, IEEE, 2009.

[11] C. Yu, G. Liu, and L. Lai, "Diagnosis of resistive-open defects using iddt in digital cmos circuits," *WSEAS Transactions on Circuits and Systems*, vol. 13, pp. 296–300, 2014.

[12] J. L. G. Gervacio, *An Aware Methodology to Evaluate Circuit Testability for Small Delay Defects*. PhD thesis, INAOE, 2009.

[13] K. Baker, G. Gronthoud, M. Lousberg, I. Schanstra, and C. Hawkins, "Defect-based delay testing of resistive vias-contacts a critical evaluation," in *International Test Conference 1999. Proceedings (IEEE Cat. No. 99CH37034)*, pp. 467–476, IEEE, 1999.

[14] D. Gizopoulos, *Advances in electronic testing: challenges and methodologies*, vol. 27. Springer, 2006.

[15] R. R. Montañés, J. P. De Gyvez, and P. Volf, "Resistance characterization for weak open defects," *IEEE Design & Test of Computers*, vol. 19, no. 5, pp. 18–26, 2002.

[16] M. McDermott, "Lecture 13: Interconnects in cmos technology," 2018. accessed 2020-10-15.

[17] I. C. Education, "Machine learning." accessed 2020-09-06.

[18] Educaba, "Machine learning vs neural network." accessed 2020-09-08.

[19] B. Klein, "Neural networks." accessed 2020-06-03.

[20] A. Chauhan, "Perceptron: A basic neural network model for deep learning," 2020. accessed 2021-06-24.

[21] J. P. Mueller and L. Massaron, *Machine learning for dummies*. John Wiley & Sons, 2021.

[22] Technopedia, "Hidden layer." accessed 2020-11-24.

[23] M. Imran, "A look into eta problem using regression in python – machine learning."

[24] J. Delua, "Supervised vs. unsupervised learning: What's the difference?." accessed 2021-03-24.

[25] T. Instruments, "Design tools and simulation. cd74hc04 behavioral spice model." accessed 2020-09-20.

[26] B. University of California, "Nominal model for 0.35µm cmos." accessed 2020-10-01.

[27] J. Wernehag, "Electrical parameters." accessed 2020-10-01.

[28] U. of California (Berkley). MOSIS., "Mosis parametric test results." accessed 2020-10-01.

[29] E. Bruun, *CMOS Integrated Circuit Simulation with LTspice*. Bookboon.com, 2017.