A STUDY OF MRF CMOS CIRCUIT DESIGN IMPLEMENTATION

By

ATIEH RANJBAR KERMANY (7187)

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

A STUDY OF MRF CMOS CIRCUIT DESIGN IMPLEMENTATION

by

Atieh Ranjbar Kermany

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr. Nor Hisham B. Hamid
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

May 2008

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

atieh R

Atieh Ranjbar Kermany

ABSTRACT

Shrinking devices to the smaller scale and reducing of voltage levels down to the thermal limit, all conspire to produce faulty systems. One possible solution for this matter is to have a paradigm shift to a fault tolerant probabilistic framework. Probabilistic computing provides a new approach towards building fault-tolerant architectures and systems. The logic states are considered to be random variables. Under this framework, one no longer expects a correct logic signal at all nodes at all times, but only that the joint probability distribution of signal values has the highest likelihood for valid logic states. The probabilistic approach is based on the theory of Markov Random Fields (MRF), which is extensible to a large number of logic variables. This theory can be used to design the circuit with high noise immunity.

This report discusses about the inverter circuits, and comparison between the obtained results for both MRF and Standard inverters using Cadence tools and MATLAB in both noisy and ideal conditions. The results are in micro-regime, since the minimum dimensions of the software were in micro-ranges.

The project focused more on the analysis of noise for both inverters and the transistors inside each one of them. As a result of completing the above procedure, it was proved that MRF inverter is tolerant to noisy conditions where as the standard inverter is not.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor, Dr. Nor Hisham B. Hamid who found time in his very busy schedule to give me his full support, monitor my progress and answer my questions. I am also deeply grateful for his advice, encouragement and patience throughout the duration of the FYP. The support he has given me as an expatriate in a predominantly Malaysian environment has been greatly appreciated too.

Secondly, I would like to show my appreciation and gratitude to Dr. Kundan Nepal (Assistant Professor, Electrical Engineering Dept. Bucknell University) and Miss. Erin Taylor (University of Florida, Department of Electrical and Computer Engineering, Advanced Computing and Information Systems Lab), for their kindness in replying my emails and questions through out this project . It was a pleasure to me to get the respond from them.

Lastly, I would like to express my thanks to Examiners for taking the time to evaluate this report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

This report presents the details of my FYP experience on 'MRF CMOS Circuit Design & Noise Analysis' from July 2007 to May 2008. As background for your reading of this report, I have included: (1) the introduction, (2) design methodology and (3) the circuit design and simulation results.

1.1 Background

According to Moore's law (Intel co-founder Gordon Moore's prediction), the number of transistors will double every eighteen months in a given space. This prediction has been the reason for rapid improvement in semiconductor industry. In the past few decades, the transistor dimensions have increasingly downscaled. However, it was predicted by the International Technology Roadmap for Semiconductors (ITRS) that the continuous downscaling will stop, perhaps around 2020, due to unavoidable physical limits.

As transistor dimensions move into the smaller range, a number of errors/ faults arise from noise. So, error-free logic states can no longer be guaranteed. It will be necessary to consider error correction designs in to the circuit.

As a solution for the matter above, researchers proposed a paradigm shift to a fault-tolerant probabilistic framework based on Markov random fields (MRF). They proposed a new framework using Markov random networks under extremely noisy conditions [1].

1.2 Problem Statement

As Si CMOS devices are scaled down into the smaller scale regime, current approaches are reaching their practical limits. Looking to the future, the next major challenges to Si CMOS include further downscaling of devices with difficulties in manufacturing. The challenges divide into two interrelated areas. On the system side, there are the computer architecture issues arising from the problem of integrating billions of transistors at the lowest possible supply voltage. On the device integration front, there is hope for being able to combine CMOS FET-based circuits with any number of alternative devices all on the same chip. Here for my FYP, I focus on system side and architecture issues [2].

While it is not sure on how far and how fast CMOS will downscale, it is certain that future devices will have high manufacturing defect rates. Further, it is clear that the supply voltage, VDD, will be scaled down to reduce dynamic power dissipation. The reduction in noise margins will result in higher soft error rates.

1.3 Objective of the project

The objective is to create reliable CMOS inverter based on principles of Markov Random Fields that are tolerant to noise such that correct logic operation may be obtained even under extremely noisy conditions with superior noise immunity compared to standard CMOS inverter using Cadence tools.

CHAPTER 2

LITRETURE REVIEW

2.1 Markov Random Fields Theory

Markov random field is a model of the Probability Distribution of a set of random variables. Probabilistic computing provides a new approach towards building fault-tolerant circuits and systems. Here, the logic states are considered to be random variables whose values can vary over the range of the logic signal level between 0V and VDD. Under this framework, one no longer expects a correct logic signal at all nodes at all times, but only that the joint probability distribution of signal values has the highest likelihood for valid logic state.

Here is a brief overview of the Markov random field theory. Consider a set of random variables called sites, X = {x1, x2, ..., xk} where each variable, xi can take on various values called labels. The sites in X are related to one another via a neighborhood system (N) defined by a set of variables from $X - \{xi\}$. This collection of random variables is called a Markov Random Field (MRF) if:

$$P\,(x) > 0,\ \forall x \in X \quad \text{(Positivity)}$$

$$P\,(xi|\{X - xi\}) = P\,(xi|Ni) \ \text{(Markovianity)}$$

In other words, a set of random variables form an MRF if all sites have a finite positive probability and the probability of a particular site in the neighborhood depends only on its immediate neighbors to which it is connected by an edge. The edges in the neighborhood represent the conditional dependence between the connected variables in the neighborhood. The conditional probability of a given site in terms of its neighborhood can be formulated in terms of the associated clique of the graph structure. Figure 1 shows one such neighborhood with one 1st order clique and one 2nd order clique [3].

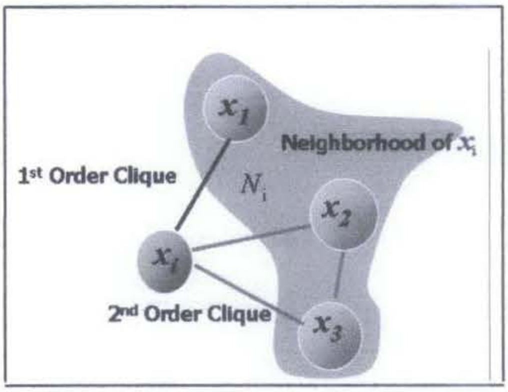

Figure 1: The MRF neighborhood system.

As mentioned earlier, in MRF theory the circuit networks are related to each other through neighborhoods, so the logic states and variables for each part are depended to other parts which can be represented in a dependence graph. Figure 2 shows a simple circuit and its corresponding dependence graph according to the MRF theory, which means that the nodes are random variables that can hold values ranging from 0V to VDD and the edges are the conditional dependencies between the variables [3].

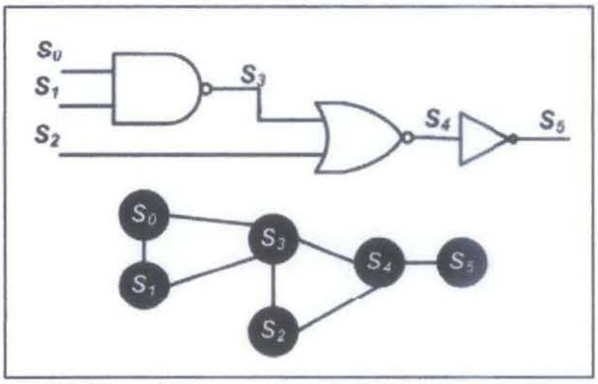

Figure 2: A simple MRF logic circuit and its dependence graph

All the logic variables, {s0, s1, s2, s3, s4, s5}, in Figure 2, are varying randomly from 0v to VDD. The correct logic states are those that maximize their joint probability. In Figure 2, three different sets of cliques [{s0, s1, s3}, {s2, s3, s4}, {s4, s5}] can be seen. Using the Hammersley Clifford theorem, the joint probability distribution can be written as;

$$p(S)=\frac{1}{Z}\prod_{c\in C}e^{\frac{-U(s_c)}{U_0}}$$

Where S is the set of all nodes in the dependence graph, C is the set of cliques, (sc) is the set of nodes in a clique c, U (sc) is the clique energy function (logic compatibility function), and U0 is an abstract term that defines the sharpness of the probability distribution. The term Z is called the partition function and is a constant required to normalize the probability function to [0, 1] [3].

The general algorithm for finding individual site labels that maximize the probability of the overall network is called belief propagation. The result of Figure 2 for the inverter circuit is shown by compatibility function f (s4, s5) in table 1 (s4 is the input and S5 is the output).

S4	S5	f
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Inverter logic compatibility function

All possible states are demonstrated in Table 1. Here, valid states are shown with f=1 and invalid states with f=0. The valid states should have a lower energy than invalid states. Thus, the clique energy will add a negative sign for valid states summation,

$$U(s_4,s_5)=-\sum_i f_i(s_4,s_5)=-(s_4s_5'+s_4's_5)$$

For the two valid states {01, 10}, the clique energy is -1 while for the two invalid states {00, 11} the clique energy is 0, since it is an inverter circuit. So, as long as the energy of the correct logic state is less than that of the invalid state, the inverter will operate correctly [3].

2.2 Building MRF Elements

The theory of Markov random fields is based on the fact that the probability of a node dependents only on its neighboring nodes. To implement this MRF concept in to the realistic devices, the bistable storage elements and feedback circuitry will be added to the circuit. This mapping contains the following two factors:

- Each logic state, s_i, should be represented as a bistable storage element, with same probability taking "0" or "1" . The probability for any other signal value should be low [3].
- Enforcing feedback from each clique to the appropriate storage elements, maximizing the probability of the correct value by implementing the logic compatibility functions.

Using simulation, I was able to show that MRF theory can be used to express circuit structures and that it has a high tolerance to noisy conditions.

2.3 MRF-Inverter

Figure 3 shows an MRF-Inverter circuit containing feedbacks, bistable elements and CMOS logic gates. The circuit consists of two storage nodes, one for s_4 and one for s_5.

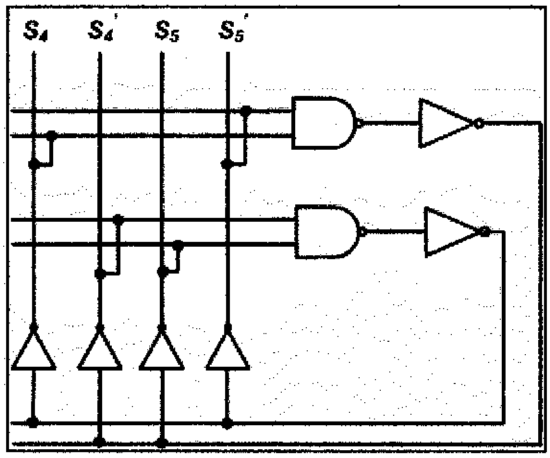

Figure 3: MRF-Inverter circuit

Considering the case that s_4=0 and s_5=1, the first NAND-Inverter gate is active and returns the logic state "1" to the inputs, thereby reinforcing the expected output value. The other NAND-Inverter gate feeds back the logic "0" state. As the result even if an error occur in any part of the circuit, it will be found and then corrected through feedback interconnects.

CHAPTER 3
METHODOLOGY

3.1 Procedure

I started my Final Year Project on 'MRF CMOS Circuit Design & Noise Analysis' from July 2007 to June 2008. Below is a flowchart of what I did in the last few months for my FYP project.

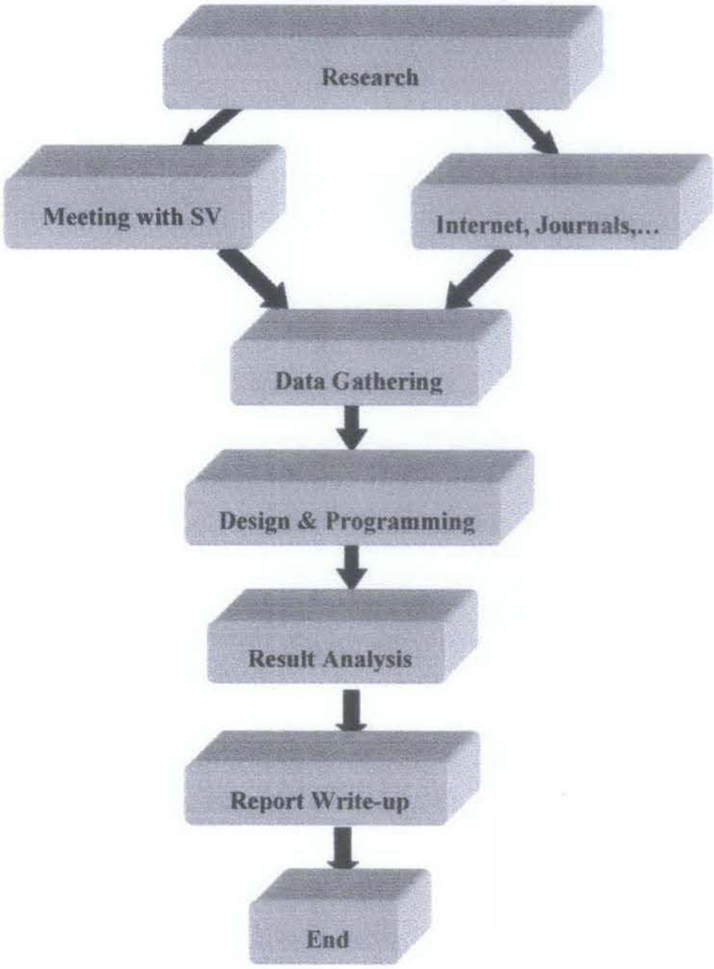

Figure 4: Flowchart for FYP

3.1.1 Research

The research involved in this study scope is on the concept of probability distribution, Building and Mapping MRF circuits, Designing, Simulation results and mainly noise and obtained result analysis.

3.1.2 Data Gathering

All the required data must be gathered to develop the research. I started the research by gathering the required data from the Internet, journals, Thesis similar report, conference papers and some related books.

My supervisor was a great help to me even though he had a busy schedule most of the times. He helped me to follow the correct path.

Another helpful source for me was Mr. Nepal who had done a PHD project with the same topic and Also Miss. Taylor who was involved in a lot of projects regarding implementing noise in the circuit. Both of them kindly replied to all my questions through email.

3.2 Tool requirement

3.2.1 Cadence Software

This is the software for designing the schematic circuits for both standard and MRF invertors, and it used for simulation purposes. The circuits are sketched in this software, and the resulted waveforms are checked for the purpose of comparison.

3.2.2 MATLAB Software

This is for programming purposes. The noise random values were obtained by doing some simple programming in this software and then they were used in a voltage source parameter in the Cadence tools for both standard and MRF invertors.

CHAPTER 4
RESULTS AND DISCUSSION

As the objective of this project, I am supposed to build the MRF circuit in micro-scale regime. For this matter I started with designing of Standard-Inverter and Standard-NAND circuits, since they both are required in the MRF circuit. After that the MRF inverter was built. The design has been done in transistor level, so then the dimensions can be arranged to the desired values.

Both inverters are simulated in ideal and noisy conditions and the obtained simulation results are analyzed and discussed in detail for each individual inverter. The comparison is made then to find the inverter which has the higher immunity to noise and faulty conditions.

4.1 Ideal condition

In ideal state both inverters are considered in an error-free condition. As the result, the output for both circuits should be the same and without any fault.

4.1.1 Standard-Inverter Circuit

Following is a CMOS Standard-Inverter circuit.

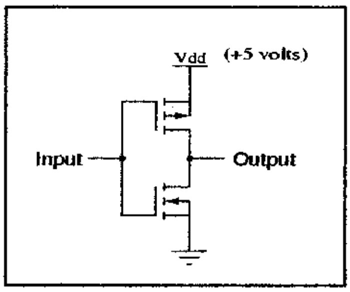

Figure 5: Standard-Inverter Circuit [5]

The "V_{dd}" label on the positive power supply terminal stands for the constant voltage applied to the drain of a field effect transistor, in reference to ground. Below is the inverter circuit with connection to power supply.

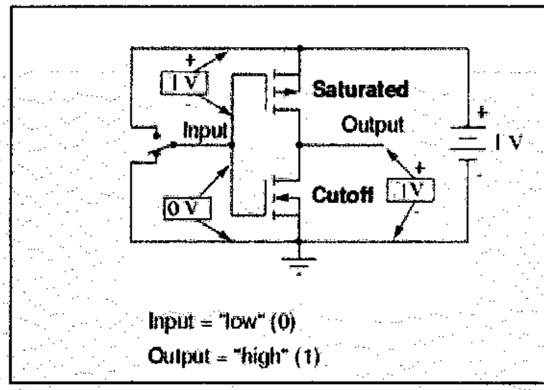

Figure 6: Standard-Inverter Circuit with power supply (substrate is more positive than gate) [5]

In Figure 6, when the channel (substrate) is made more positive than the gate, the channel is enhanced and current is allowed between source and drain. So, in the above illustration, the top transistor is turned on.

The lower transistor, having zero voltage between gate and substrate (source), is in its normal mode, off. Thus, the actions of these two transistors are such that the output terminal of the gate circuit has a solid connection to V_{dd} and a very high resistance connection to ground. This makes the output "high" (1) for the "low" (0) state of the input. Below, I move the input switch to its other position:

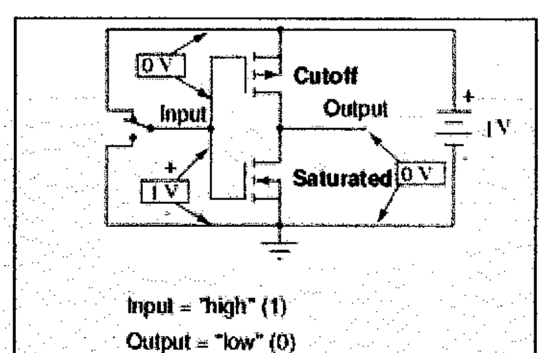

Figure 7: Standard-Inverter Circuit with power supply (gate is more positive than substrate) [5]

Here the lower transistor (N-channel) is saturated because it has sufficient voltage of the correct polarity applied between gate and substrate (channel) to turn it on. The upper transistor, having zero voltage applied between its gate and substrate, is in its normal mode, off. Thus, the output of this gate circuit is now "low" (0). Clearly, this circuit exhibits the behavior of an inverter, or NOT gate. Below is the table of an Inverter logic conditions.

Input	Q1 (PMOS)	Q2 (NMOS)	Output
HI	OFF	ON	LOW
LOW	ON	OFF	HI

Table 2: The truth table for Standard-Inverter Circuit

4.1.1.1 Simulation Results

Above were all the Theories of designing a Standard-Inverter CMOS Circuit. Below is the circuit design and obtained simulation results (figure 8 & 9). As it can be seen from the schematic circuit, I used two transistors (one PMOS and one NMOS) with width of 3 and 1.5 um respectively, and length of 600nm (these are the minimum width and length that I could use in Cadence, for having a smaller scale I required to have specific library which is not available in UTP). In simulation environment, and in Stimuli window, the values for input voltage and Vdd are given (V1=1v, V2=0v, pulse width= 50us, period=100us, Vdd=1v).

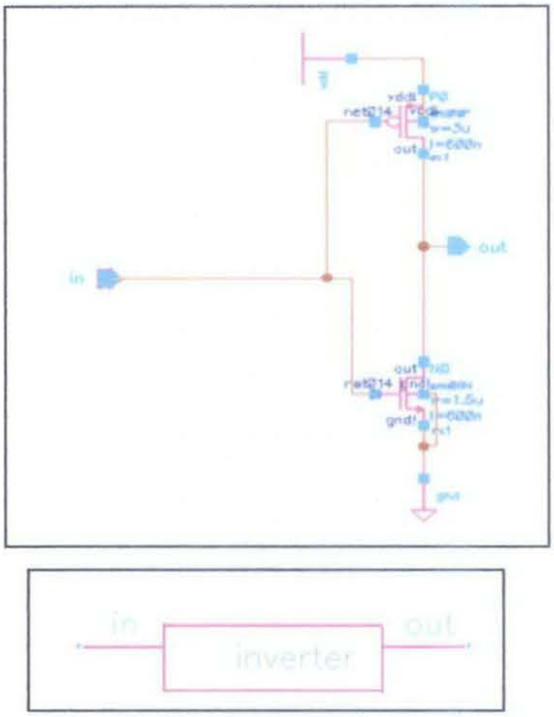

Figure 8: Schematic and Symbol of Standard-Inverter CMOS circuit

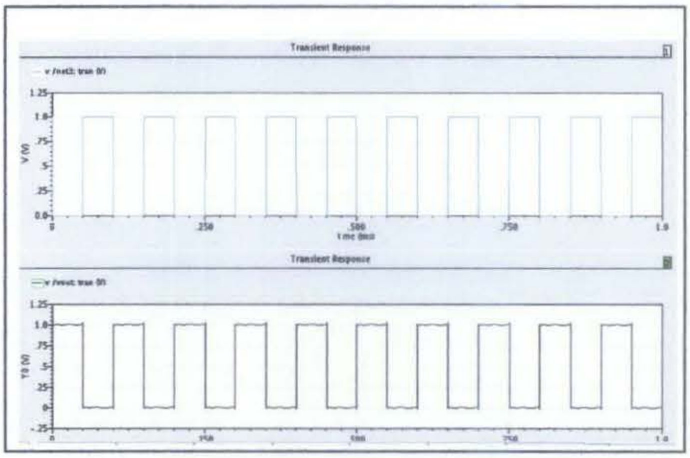

Figure 9: Simulation results for Standard-Inverter Circuit

The blue color waveform is the input; it is zero for a pulse width of 0.05ms and then it goes to one for again 0.05ms. The black color waveform is the output; it follows the reverse sequence of input waveform that means that it started with one for a period of 0.05ms and then it goes to zero.

As the conclusion for Standard-Inverter circuit design, it can be seen from the resulted simulation (Figure 9) that the circuit is working properly and it is inverting the respective input.

4.1.2 MRF-Inverter circuit

The MRF circuit and the obtained simulation waveforms are shown in Figure 10, in comparison with Standard-Inverter CMOS circuit. As it can be seen from the waveforms the MRF has a much higher noise tolerance comparing to Standard-Inverter circuit. This is the result for a nano-regime circuit design that I found from internet and I used it here for the more understanding of the concept.

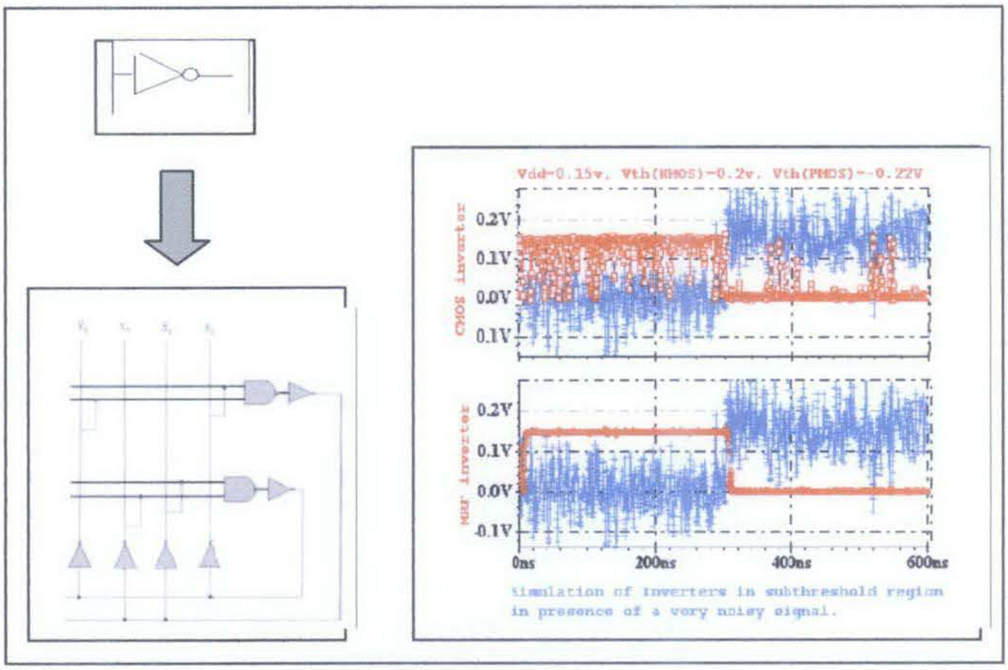

Figure 10: Schematic & Simulation Results of Noise-Tolerant MRF-Inverter and Standard-Inverter [4]

As it can be seen from the figure above, the MRF circuit consists of two NAND circuit and six standard inverter circuits. So, before designing the MRF inverter, it is required to design the NAND circuit. Below is the discussion on the NAND circuit design procedures.

4.1.2.1 Standard-NAND Circuit

Following is a CMOS Standard-NAND circuit.

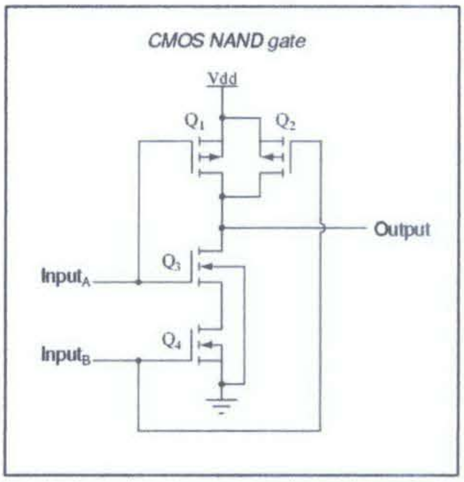

Figure 11: Standard-NAND Circuit [5]

In figure above (figure 11), transistors Q_1 and Q_3 resemble the series-connected complementary pair from the inverter circuit. Both are controlled by the same input signal (input A), the upper transistor turning off and the lower transistor turning on when the input is high, and vice versa. At the same time, transistors Q_2 and Q_4 are similarly controlled by the same input signal (input B), and they will also exhibit the same on/off behavior for the same input logic levels. The upper transistors of both pairs (Q_1 and Q_2) have their source and drain terminals paralleled, while the lower transistors (Q_3 and Q_4) are series-connected. This means that the output will go high if either top transistor saturates, and will go low only if both lower transistors saturate. The following sequence of illustrations (Figure 12) and truth table (Table 3) show the behavior of this NAND Gate for all four possibilities of input logic levels (00, 01, 10, and 11):

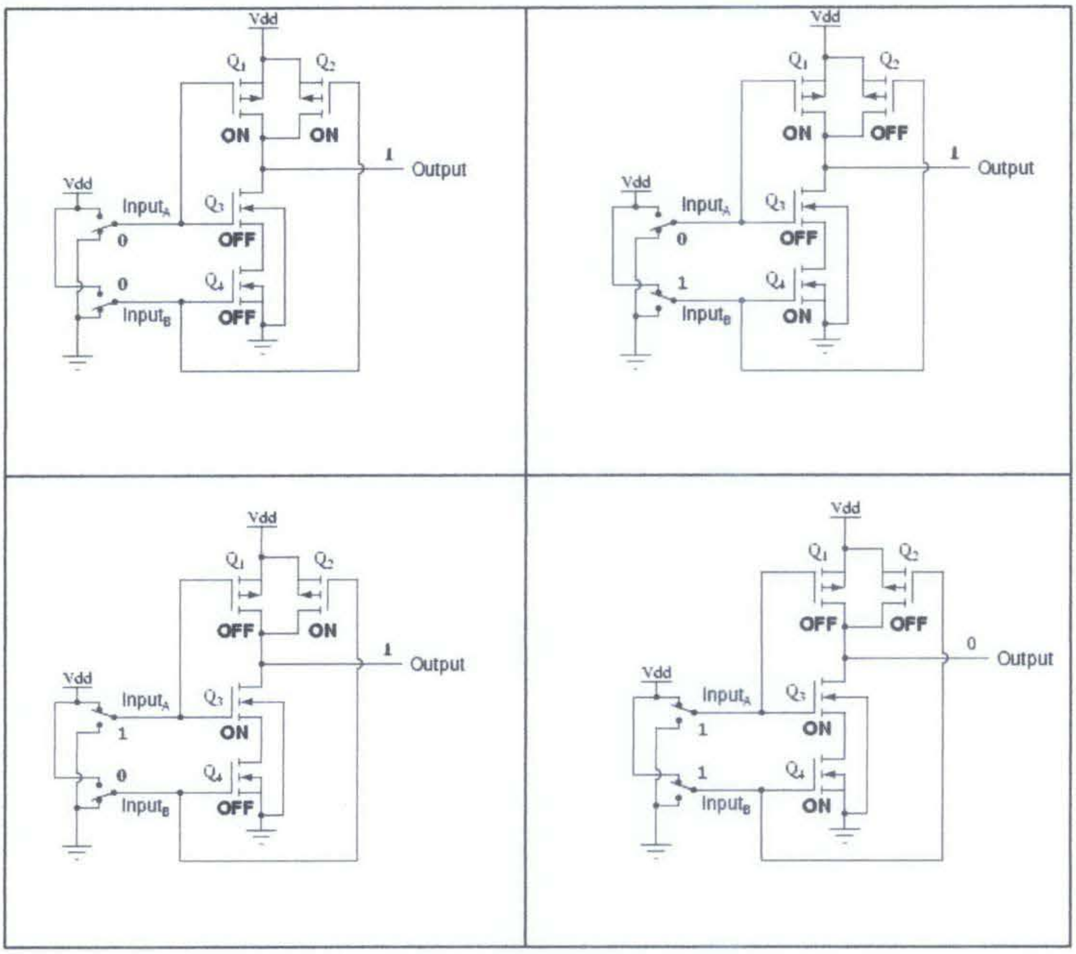

Figure 12: The behavior of NAND Gate for all four possibilities of input logic levels (00, 01, 10, and 11) [5]

Input		Transistors				Output
A	B	Q1	Q2	Q3	Q4	Output
LOW	LOW	ON	ON	OFF	OFF	HI
LOW	HI	ON	OFF	OFF	ON	HI
HI	LOW	OFF	ON	ON	OFF	HI
HI	HI	OFF	OFF	ON	ON	LOW

Table 3: Truth table for Standard NAND Circuit

4.1.2.1.1 Simulation result

Below are my Schematic circuit and simulation results for Standard-NAND circuit (Figure 13 & 14). It can be seen from the schematic diagram that I used four transistors in my circuit (2 PMOS, and 2 NMOS) with width of 1.5um and length of 600nm for each transistor. Beside that there are two input voltages. In simulation environment, and in Stimuli window, the values for input voltages and Vdd are (for vin1: [V1=0v, V2=1v, pulse width= 50us, period=100us], for vin2: [V1=1v, V2=0v, pulse width= 20us, period=40us], Vdd=1v). This is due to providing all logic conditions for the inputs (00, 01, 10, and 11).

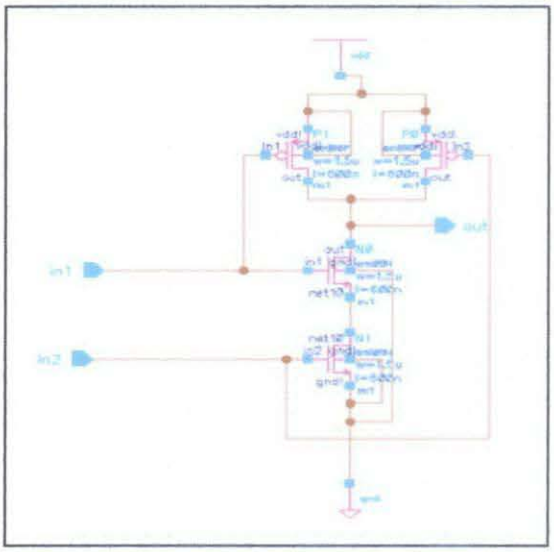

Figure 13: The schematic circuit for Standard-NAND circuit

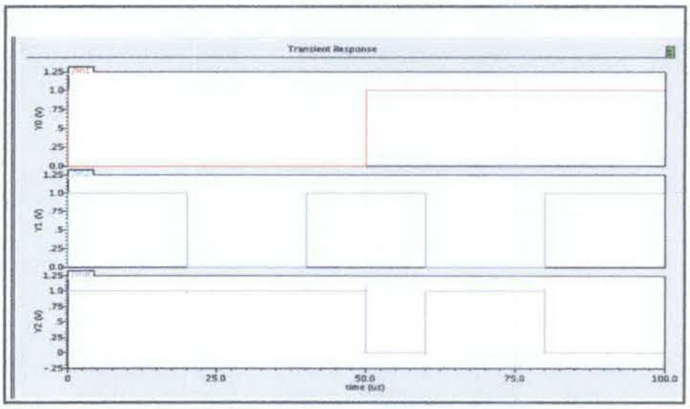

Figure 14: Simulation results for Standard-NAND circuit.

The simulation result shows that the NAND circuit is working correctly according to its logic sequence and it follows its respective truth table (Table 2). The output is LOW only when both inputs are high.

4.1.2.2 MRF-Inverter Simulation Result

Figure 15, shows the schematic design for an MRF-Inverter circuit. As it was shown previously, it consists of two Standard-NAND circuits and six Standard-Inverter circuits. The order of pins are S5', S4, S4' and S5 respectively (S4 is the input and S5 is the output). Each input point is connected to a voltage supply with period of 0.1ms and pulse width of 0.05ms. To have all the four logic conditions, the second supply has a delay of 0.025ms comparing to first voltage supply. The transistors' width and length are 1.5um and 600nm respectively.

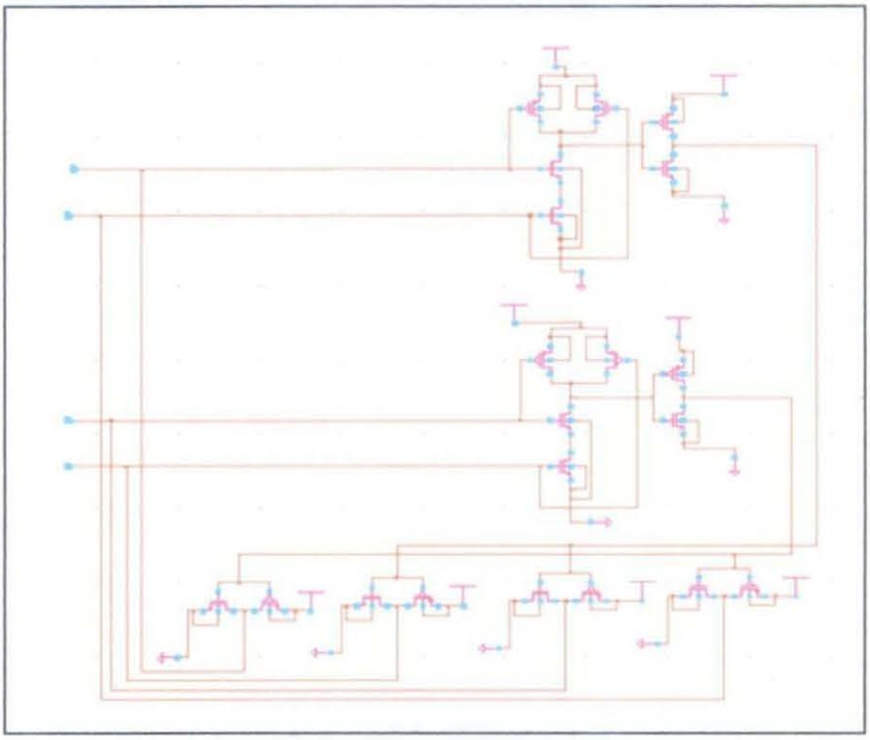

Figure 15: Schematic design for MRF-Inverter circuit

Figure 16, is the symbolic schematic of the MRF-Inverter circuit of Figure 15. A block has replaced each Standard-Inverter and NAND circuits.

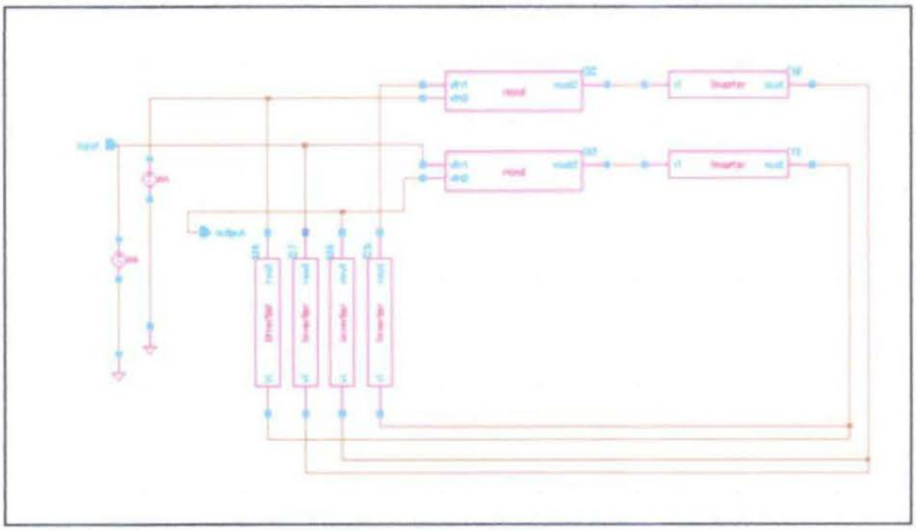

Figure 16: schematic design for MRF-Inverter circuit using symbols for each standard-NAND and standard-Inverter.

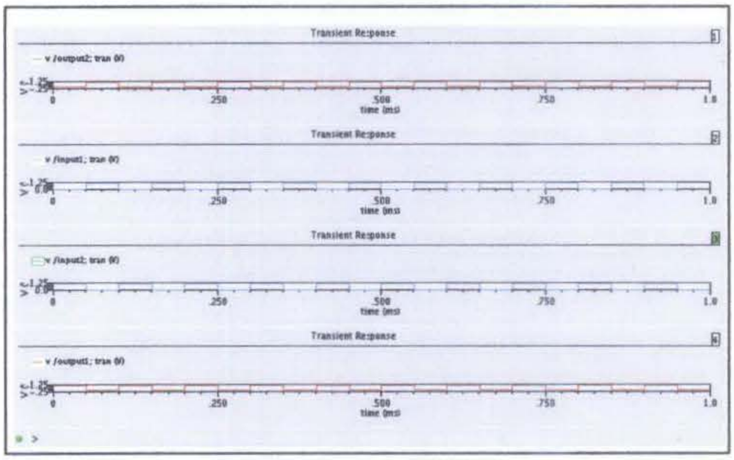

Figure 17: Simulation results for MRF-Inverter circuit (Red- vout2, Blue- vin1, Blue- vin2, Red- vout1 & vin2= vin1', vout2= vout1')

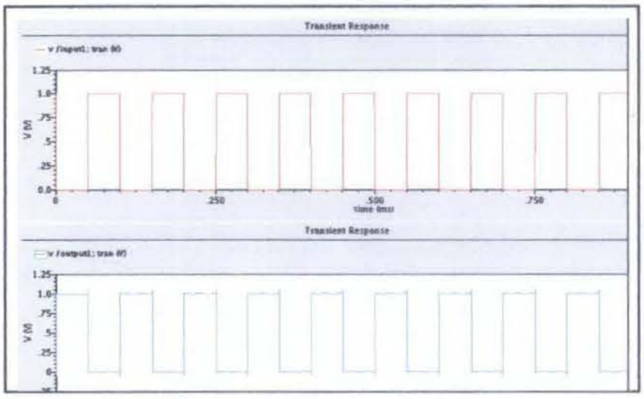

Figure 18: Simulation result for MRF-Inverter circuit (Red- Input, Blue- Output).

As for the conclusion, the simulation results are correct and the MRF circuit is inverting the input properly (as shown in Figure 17 & 18).

To be mentioned, before using cadence software, I tried to make use of the Pspice software but I could not get the correct output since the transistors' length and width were not in nano or micro ranges and as the result I had either some shift in voltage value or the circuit was just working as a buffer instead of inverter.

As it could be observed both inverters have the same output results in ideal conditions.

4.2 Noisy condition:

As mentioned earlier, shrinking devices to smaller regime will result in fault and noisy systems due to the thermal limits. To prove the noise tolerance of MRF circuit, an external noise (Gaussian noise) is applied to both inverter circuits. Using this method will help to compare the simulation result of both inverter circuits.

Noise can be implemented in Cadence software using different methods. First is to use the VPWLF Parameter by implementing the gathered data from a noise program in MATLAB into this parameter. This is the easiest way to implement the noise. In this method, firstly the noise code is generated in MATLAB and the produced data is saved in a text file like notepad (the number of the obtained data should match with the simulation time and step size). After that, a vpwlf device from AnalogLib is added to each transistor or input voltage source in the Cadence schematic. For each of these devices, a file name should be specified to be used for the device. Without specifying an explicit path to the file, the file is assumed to reside in the netlist directory. It is usually a good idea to use full pathnames. So, the VPWLF components are used to open a file in the schematic.

Another method is to use the verilag-A programming to program the noise data. In this project the first method is implemented which I using the vpwlf parameters.

4.2.1 NOISE Program in MATLAB

To implement noise using MATLAB, I used two functions, and I compare them in this report. Below is the first function which being used in MATLAB to generate about 400 data for both time and voltage.

```
%SimulationTime and TimeIncrement are in microseconds
%Mean and StDev of Guassian distribution is in V
%Nominal voltage is in V
function out = GenerateGaussianInputNoise(SimulationStartTime,
SimulationEndTime, TimeIncrement, Mean, StDev, NominalVoltage)
fid = fopen('pwlFile.in','w');
for i=SimulationStartTime:TimeIncrement:SimulationEndTime
 time=i;     x=randn(1)*StDev + Mean;     x=NominalVoltage + x;
 fprintf(fid,'%6.5fe-6 ',time);     fprintf(fid,'%4.5f\n',x);     end
fclose(fid);     out=1;
```

The next step after writing the program in MATLAB M-file page is to call this function in the MATLAB command window by using the command below;

```
GenerateGaussianInputNoise(0, 20,0.5, 0.497, 0.2885,0.1);
```

NOTE: The transition time of the circuit is 20 us and the time used in the function is in microseconds. So, the *SimulationEndTime* is 20 us.

Using 0.5 for *TimeIncrement* will result in generating 400 data.

The *Mean* and *StDev* are defined terms in MATLAB and they can be calculated using the formulas as below;

```
X = rand (1000, 1);     n = length (x);     Mean = sum (x)/ n;
StDev = sqrt (sum ((x-Mean).^2/ n))
```

The generated data of the function will be saved in "PwlFile.in" file and can be used in the VPWIF parameter by employing its path.

The second function is by writing the command below in MATLAB. This function will result in having 1000 noise data. Here the simulation time is 100us, which is obtained by (i/10) command.

```
format long
for i=1:1000  Time(i,1)=i/10*1e-6;
end
voltage=randn(1000,1);
 x=[Time voltage]
```

Then the next step is to copy and paste the variables in to a notepad document with a name like "noisefile.in".

Note: it is required to add the suffix "in" to the end of the file name.

For this project I used both options for the matter of comparison

4.2.2 First MALAB Function

For the first method, I generated the noise for 400 values using the function as stated in the above section (section 4.2.1). The results are discussed for both invertors.

The schematic and achieved waveforms are shown in sections below, as can be observed from the simulation result, the output of the circuit is not the same as the input and it is noisy, the attenuation of the noise can vary by changing the value of nominal voltage in the Noise function (The VDD is 2v in here).

4.2.2.1 Standard inverter

In standard inverter, there are only two transistors. The noise source is given firstly, to one of the transistors (Figure 19), and then to both transistors (Figure 20). The output is noisy for both cases accordingly and the simulation result proves it.

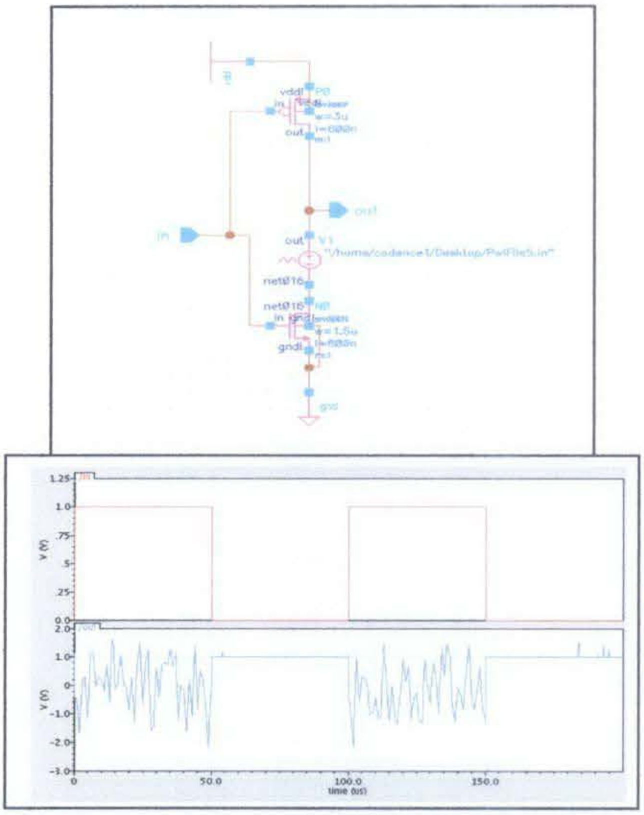

Figure 19: standard inverter schematic & simulation result for only one noisy transistor.

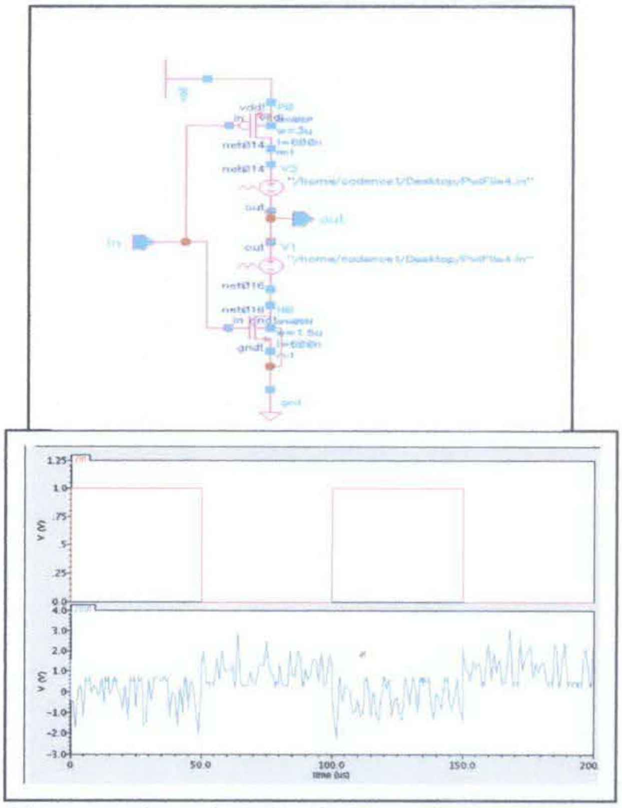

Figure 20: standard inverter schematic & simulation result when both transistors are noisy.

4.2.2.2 MRF inverter

The MRF inverter has totally 20 transistors (12 transistors for standard inverter and 8 transistors for NAND circuits). The noise sources are firstly given to all transistors (Figure 21), then to some randomly chosen transistors (Figure 22 & 23). The results are shown in the simulation figures. The VDD is 2v for the first figure, so the output voltage attenuation is up to 2v and it is 1v in the other two figures.

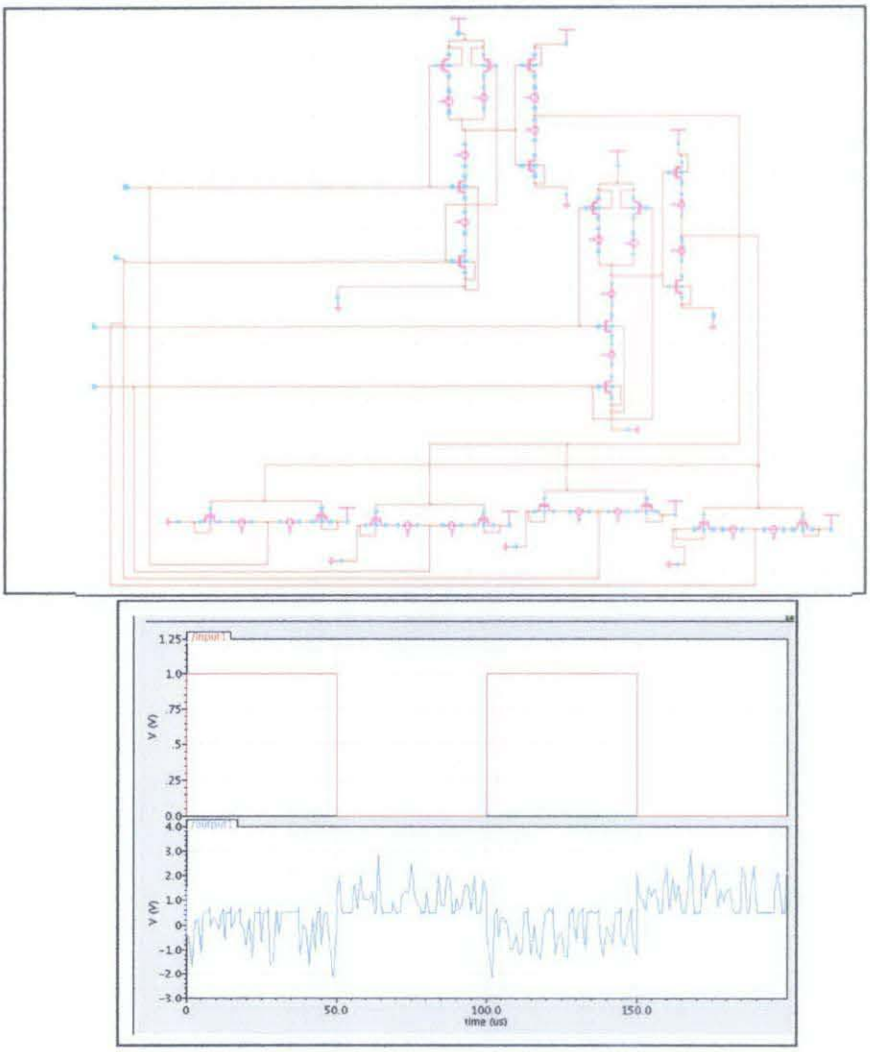

Figure 21: MRF inverter schematic & simulation result with having noise in all transistors (20 noise sources).

As it can be observed from the simulation result the output voltage is noisy and it is the same case as the standard inverter. If an error occurs in input voltage it will affect the out put voltage value as well.

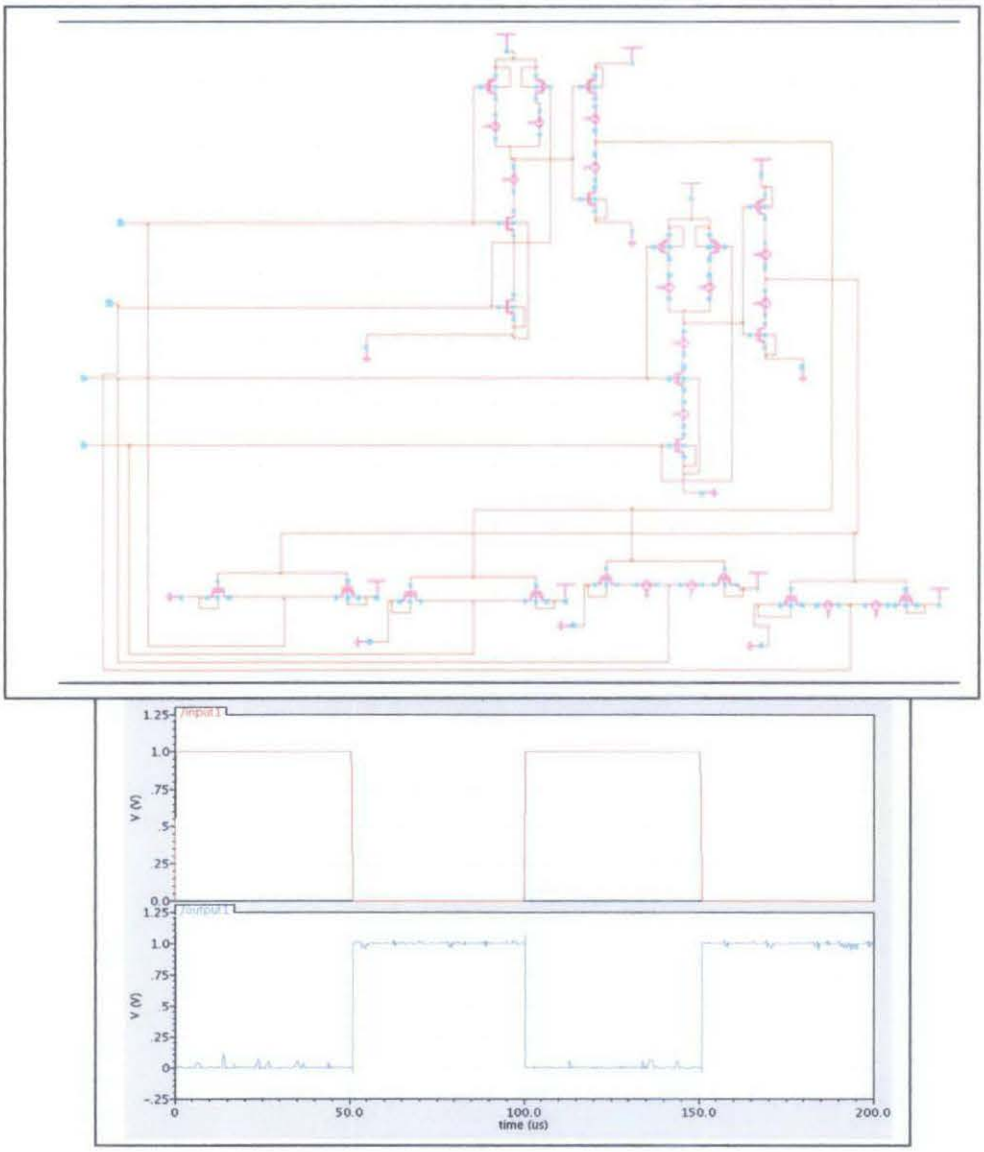

Figure 22: MRF inverter schematic & simulation result with 15 noisy transistors (15 noise sources).

As it can be seen from Figure 22, the MRF inverter is tolerant to noise when the feedback inverters which are going to output voltages are excluded from having the noise source. This means that if an error happens inside the feedbacks going to the output source, the output will be noisy, but this is not true for the other transistors. This is due to the fact that those feedbacks are connected directly to the output pins, which means that if an error occurs in any transistors inside them, it will directly affect the output (here the VDD is 1v).

There is another transistor which will result in having noise in the output. The figure below (Figure 23) shows the result.

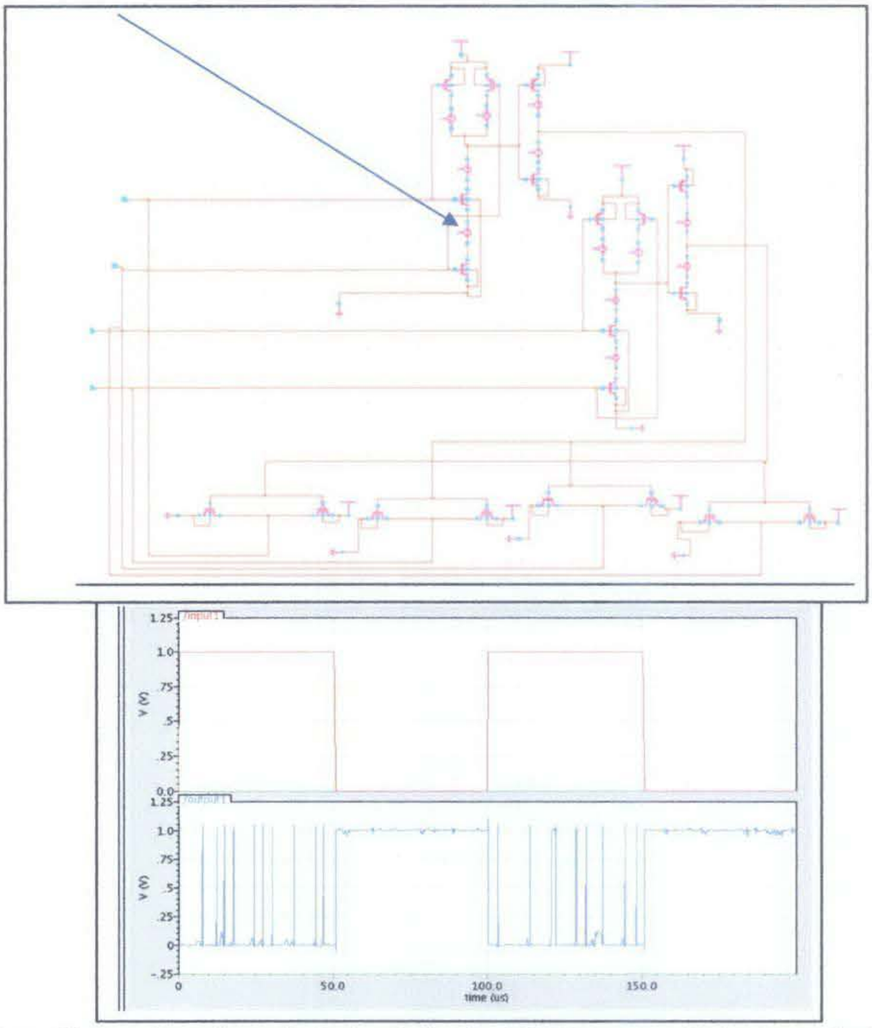

Figure 23: Another schematic & simulation for MRF inverter with having noise in some transistors (12 noise sources).

In Figure 23, there are some noise in the output, the reason is that there are some noise random values greater than vdd=1v. As the result the inverter mistakenly take those random variables noise data instead of the input data when input is one volt, since they have a higher value than the input. So for the output instead of having a steady value of zero, there are some noise data equal to 1volt. This may happen to any other transistor in the circuit as well. It can be concluded that if an error occur in a transistor it may affect the output, but it will only result for the random values grater than 1v when the input is high, which means that by reducing the nominal voltage this error can be solved.

As the conclusion, the standard circuit output is noisy even if one of the transistors is noisy, but in MRF circuit, the output will be noisy only if one (or more) of the feedbacks transistors going to the output, is noisy.

4.2.3 Second MATLAB Function

In this method I generated 1000 noise data random variables using the function stated in section 4.2.1. Using command [voltage= randn(1000,1)], the random values voltage ranges are from -2v to 2v, which is greater than vdd value (vdd is 1v here), as the result, the output will be noisy every where for standard inverter and it is also noisy for MRF inverter when the input is high. After checking this result, I decided to reduce the random values range of voltage to below one volt. So the command will change to below;

```
format long
for i=1:1000  Time(i,1)=i/10*1e-6; end
voltage= sqrt(0.1)* randn(1000,1); x=[Time voltage]
```

After that I used the command "plot" to plot the random values voltage vs. time. Figure 24 shows that all the random values are from -1v to 1v.

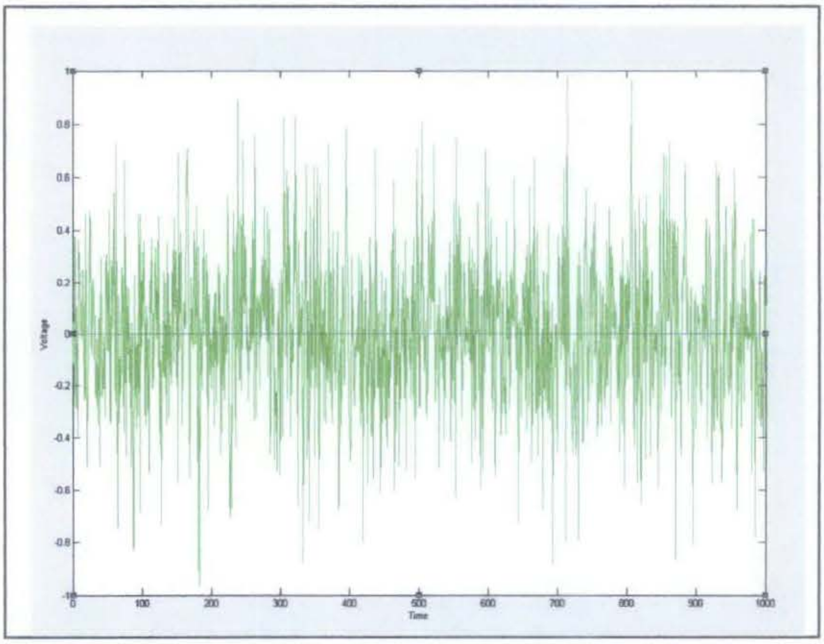

Figure 24: Noise values (1000 values) plot for voltage vs. time.

This method has been used for both inverters and the results are obtained for both commands (voltage = randn(1000,1); & voltage= sqrt(0.1)* randn(1000,1);) accordingly. For the matter of comparison the simulation result has been checked for less than 1000 variables in some cases as well.

4.2.3.1 Standard inverter

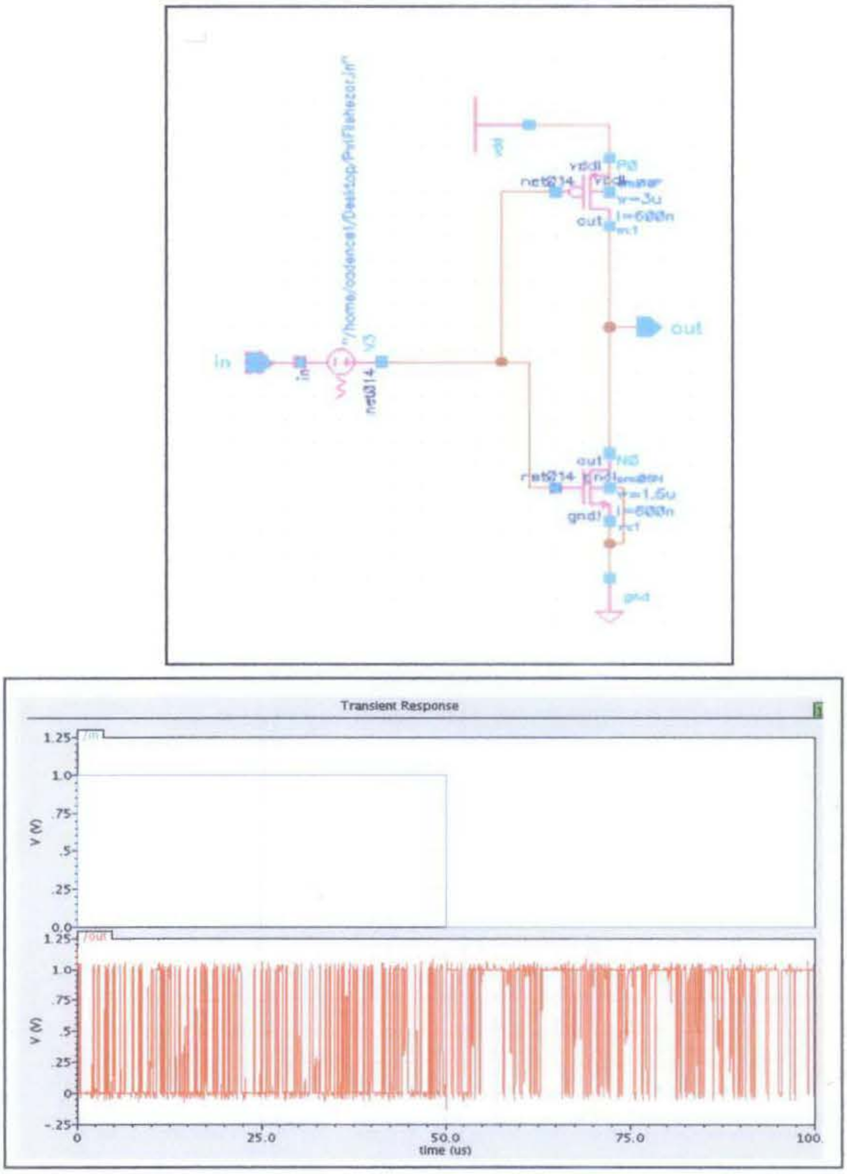

Figure 25: Standard Inverter circuit and simulation result with noise source connected to input voltage source when the voltage attenuation is from (-2 to 2) v.

Figure 25 shows that the output of the standard inverter is noisy when there is an error in the input voltage source (the noise source is connected to the input voltage source here) for the random voltage values ranging from -2v to 2v according to the first command (voltage = randn(1000,1)).

Below (Figure 26, 27, and 28) are the simulation results when the number of random values reduced from 1000 values to respectively 400, 200, and 100 values. By changing the command to this;

format long

*for i=1:400 (200 or 100) Time(i,1)=i/4 (2 or 1)*1e-6; end*

voltage= randn(400 (200 or 100),1); x=[Time voltage]

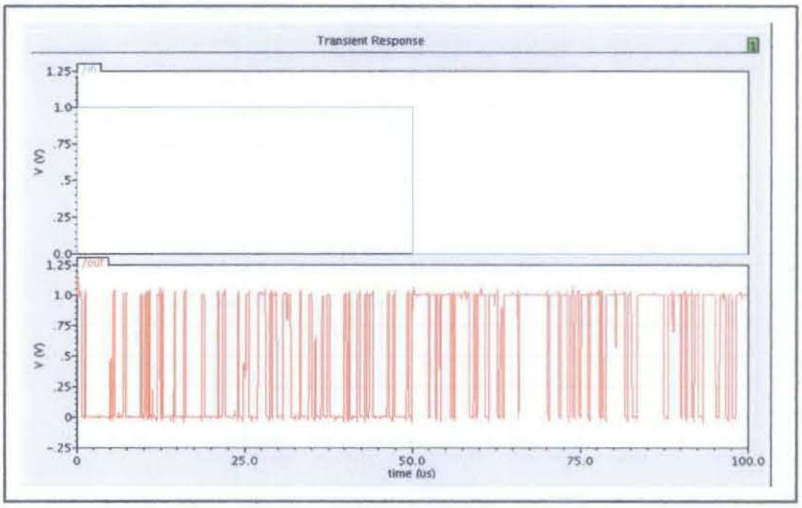

Figure 26: Simulation result for standard inverter with 400 random values.

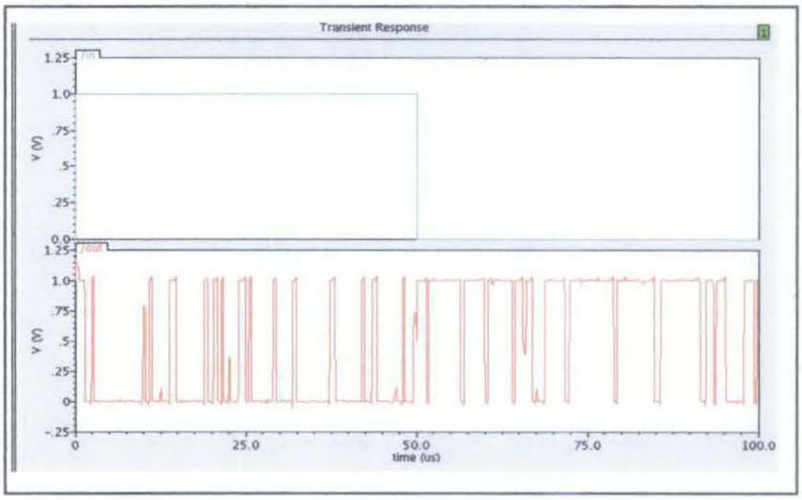

Figure 27: Simulation result for standard inverter with 200 random values.

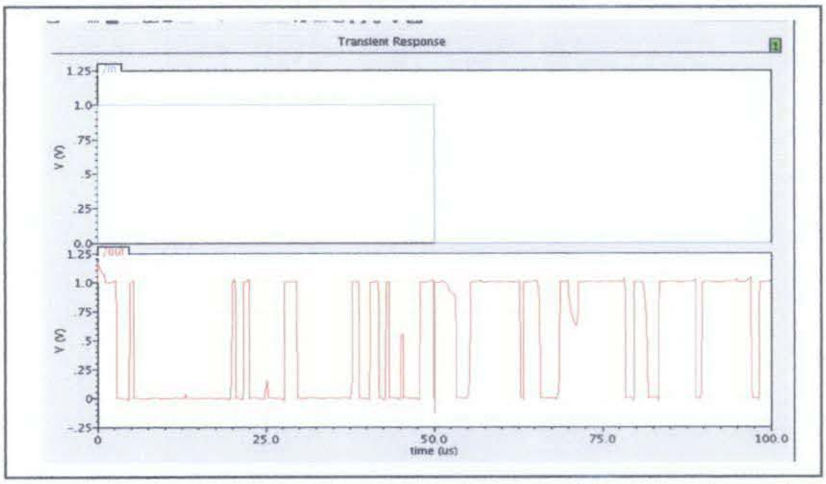

Figure 28: Simulation result for standard inverter with 100 random values.

Figure 29 shows the standard input with the random variables voltage attenuation from -1v to 1 v, using the command [voltage= sqrt(0.1)*randn(1000,1)].

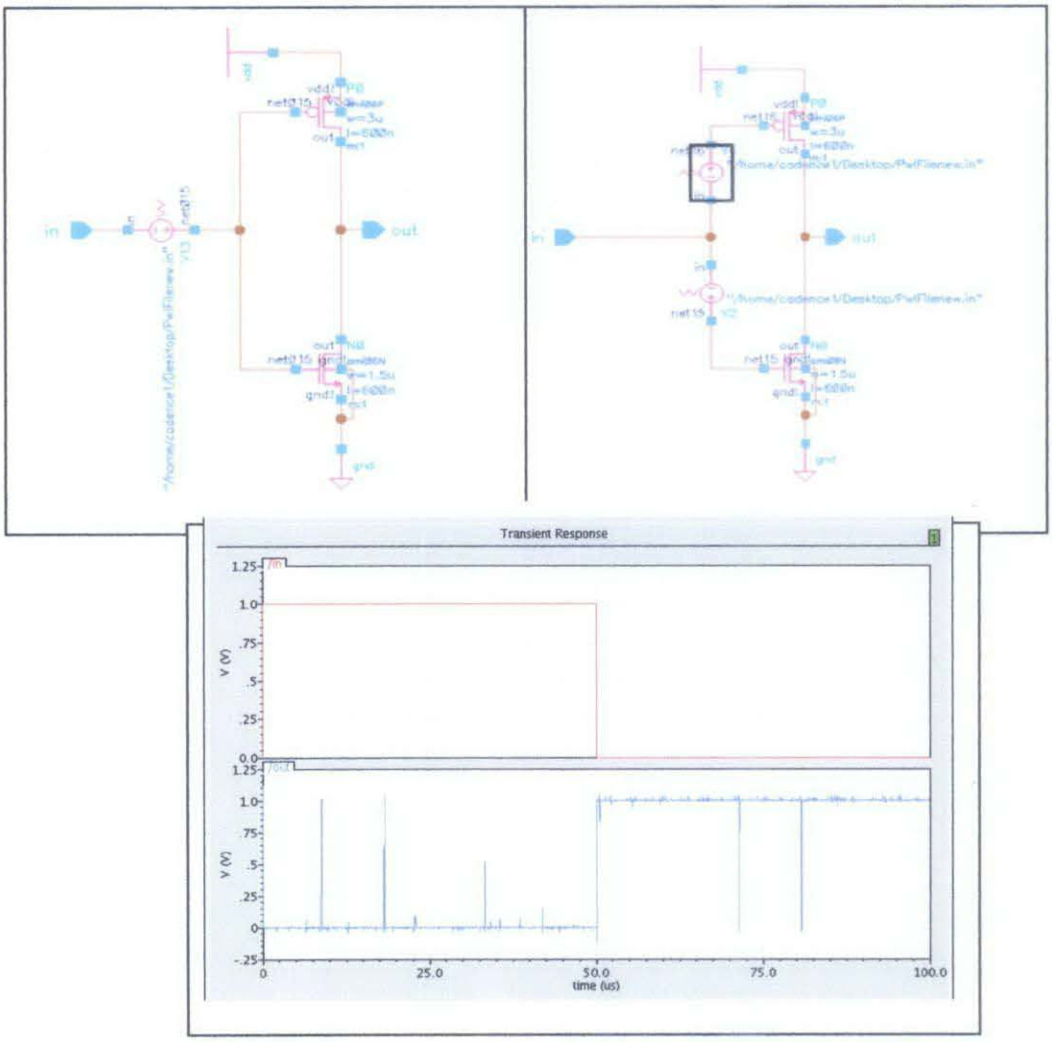

Figure 29: Standard inverter schematic and simulation with noisy input voltage source when the voltage attenuation is from -1v to 1v.

Figure 29 shows that the standard inverter is noisy, but the tolerance is much better comparing to the last case. So, it can be concluded that reduction in the noise attenuation will increase the noise immunity of the circuit when an error occur in the input voltage source.

4.2.3.2 MRF Inverter

The same matter is tested using MRF circuit. Primarily, the circuit is examined using the first command [(voltage= randn(1000,1)) and some analysis has been done using this command (Figures 30 & 31). Then, the second command [voltage= sqrt(0.1)* randn(1000,1)] has been analyzed (Figures 32, 33, 34, 35, 36, 37, and 38).

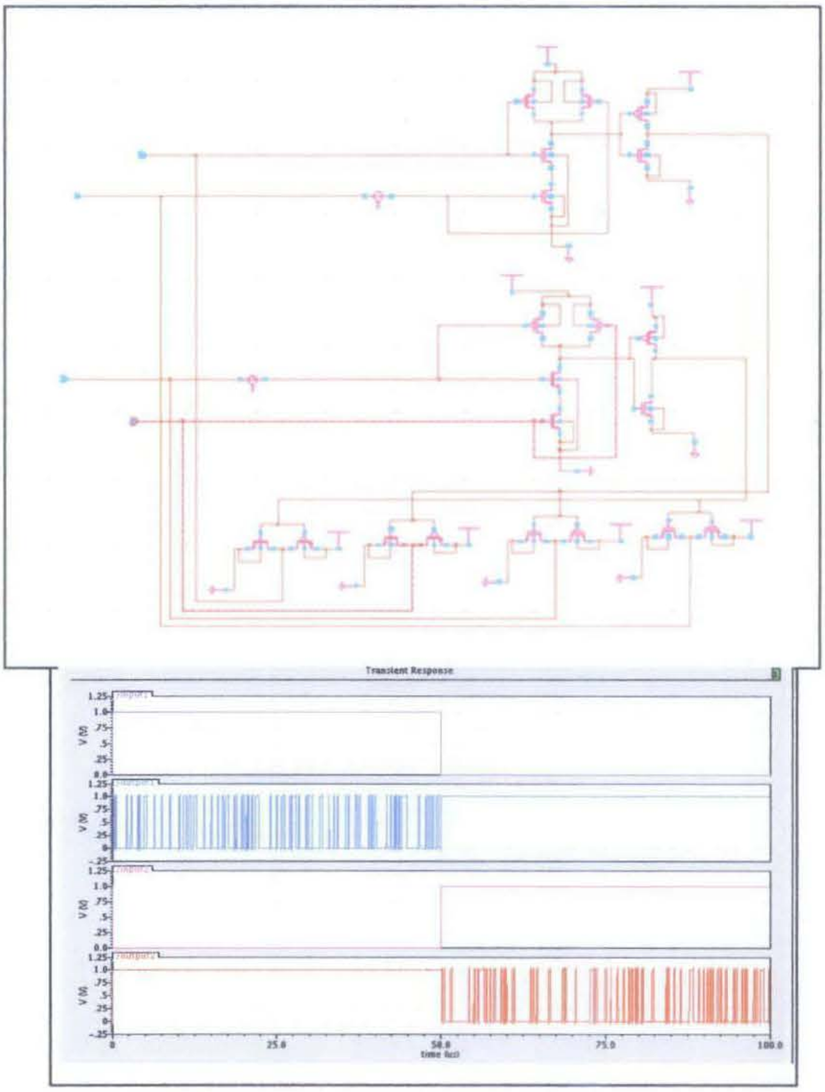

Figure 30: MRF circuit and simulation result when the noise source is connected to the input voltage source with voltage attenuation from (-2 to 2) v.

As it can be observed, the output is noisy when the input is one (or the output is zero). This is due to the fact that there are some noise voltage values greater than vdd=1v. But since the inputs and outputs are complementary pairs (input s_4 & s_4', output s_5 & s_5'), this will help to eliminate the noise to some extant. The reason that this noise is not eliminated through the feedback path is that according to MRF theory the valid state should always have a lower energy than invalid state which is not true in figure above. For the values of input equal to one (output equal to zero), the valid state has higher energy than the invalid state (noise random variables values greater than one volt). As the result the output follows the noise random variables values. But, when the input is zero, the output will be one, since the valid state has always a lower energy than any of those noise data invalid states.

Below is the result of connecting the noise source to some transistors randomly instead of the input voltage source. As it can be seen, the output voltage is not noisy.

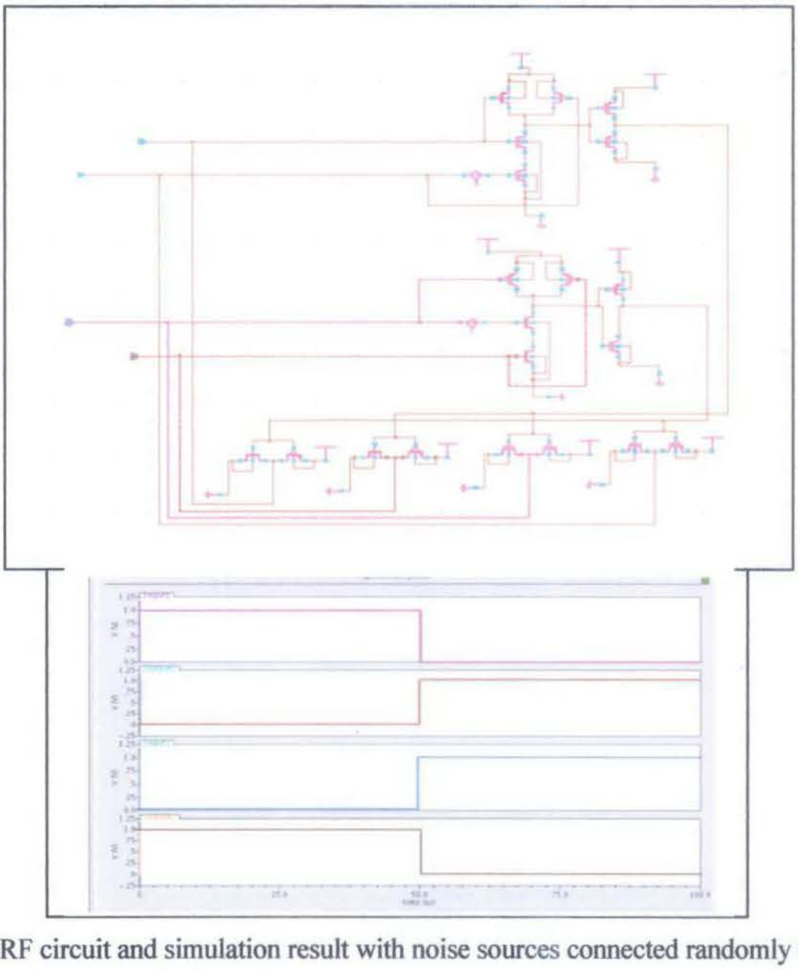

Figure 31: MRF circuit and simulation result with noise sources connected randomly to some transistors.

Below are the results obtained from using the random variables with reduction in voltage attenuation to (-1 to 1)v inside the noise source for NAND circuit (Figure 32) and finally MRF inverter (Figures 33, 34, 35, 36, 37, and 38).

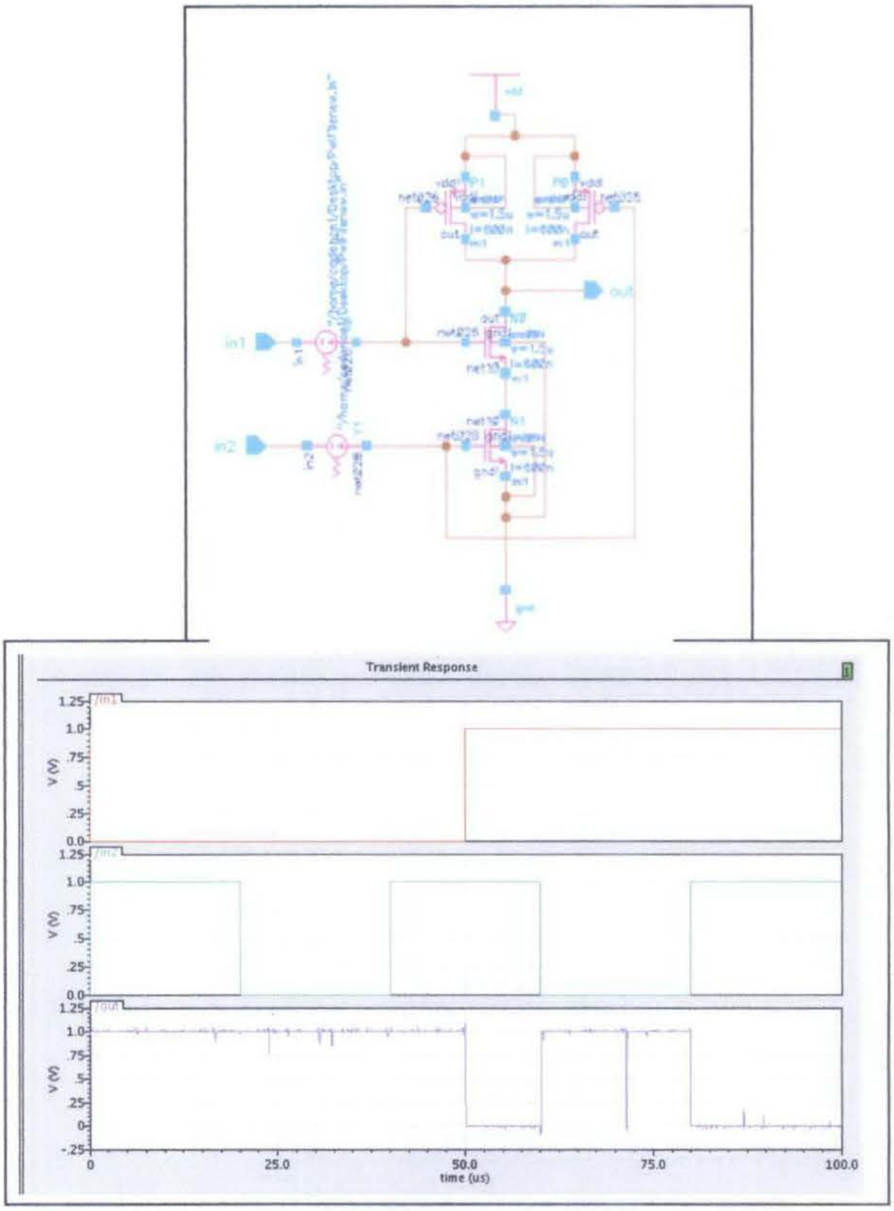

Figure 32: NAND schematic and simulation with noisy input voltage sources with voltage attenuation from (-1 to 1) v.

The output from the NAND circuit is noisy, but with a higher tolerance to noise due to the reduction of voltage attenuation.

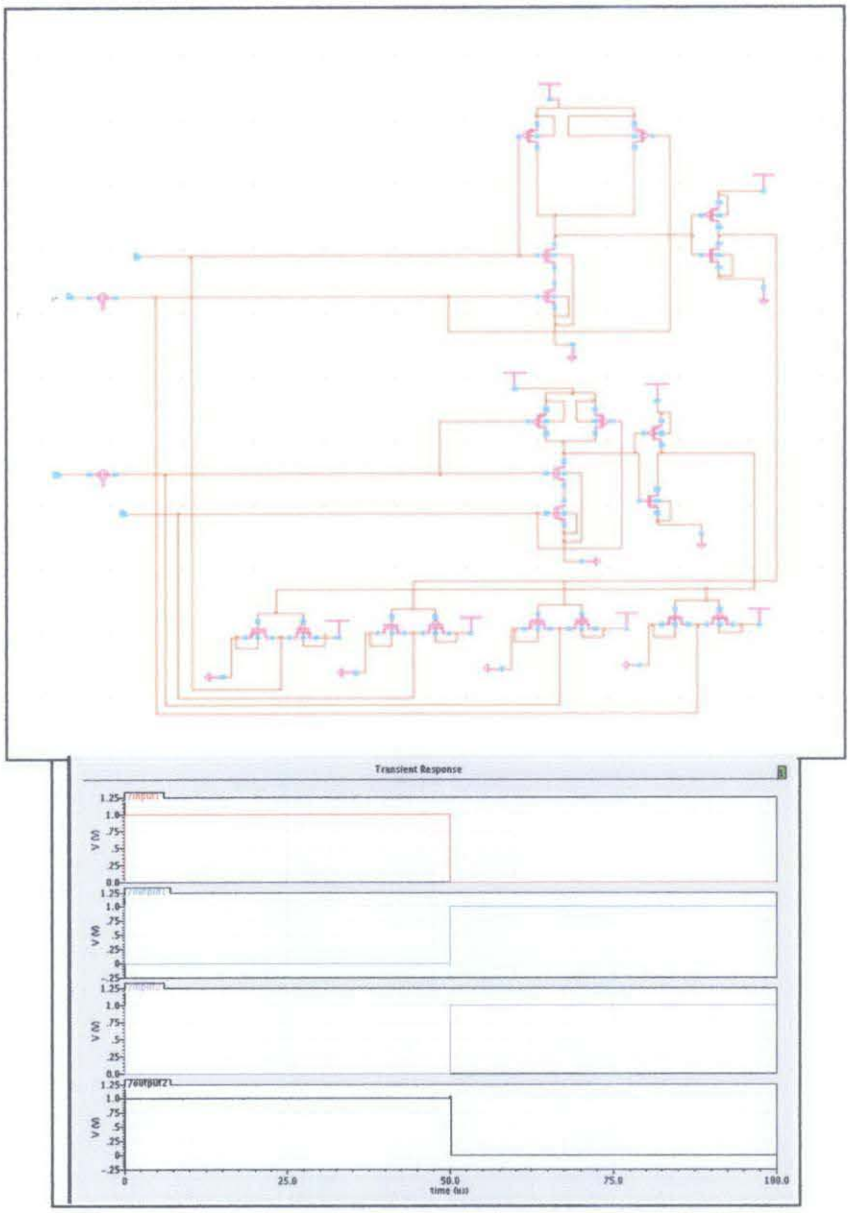

Figure 33: MRF inverter schematic and simulation with noisy input voltage sources with voltage attenuation from (-1 to 1) v.

Figure 33 shows that the MRF inverter has a high tolerance and immunity to noise when the noise source connected to the input voltage sources. In comparison with the standard inverter, the immunity is much better. Below (Figures 34, 35, 36, 37, and 38) are the results of applying the noise source to some random transistors.

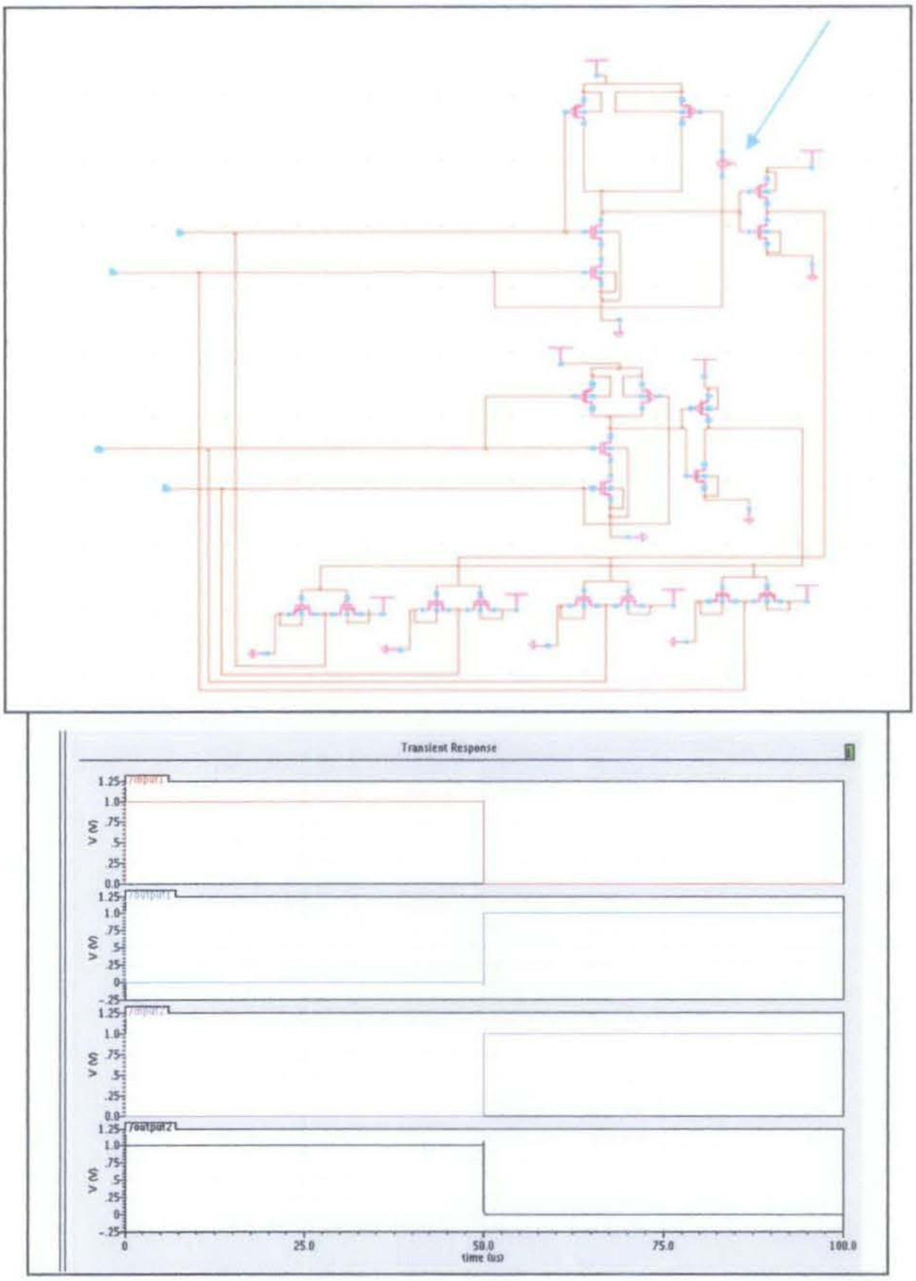

Figure 34: MRF Inverter schematic and simulation with one noisy transistor.

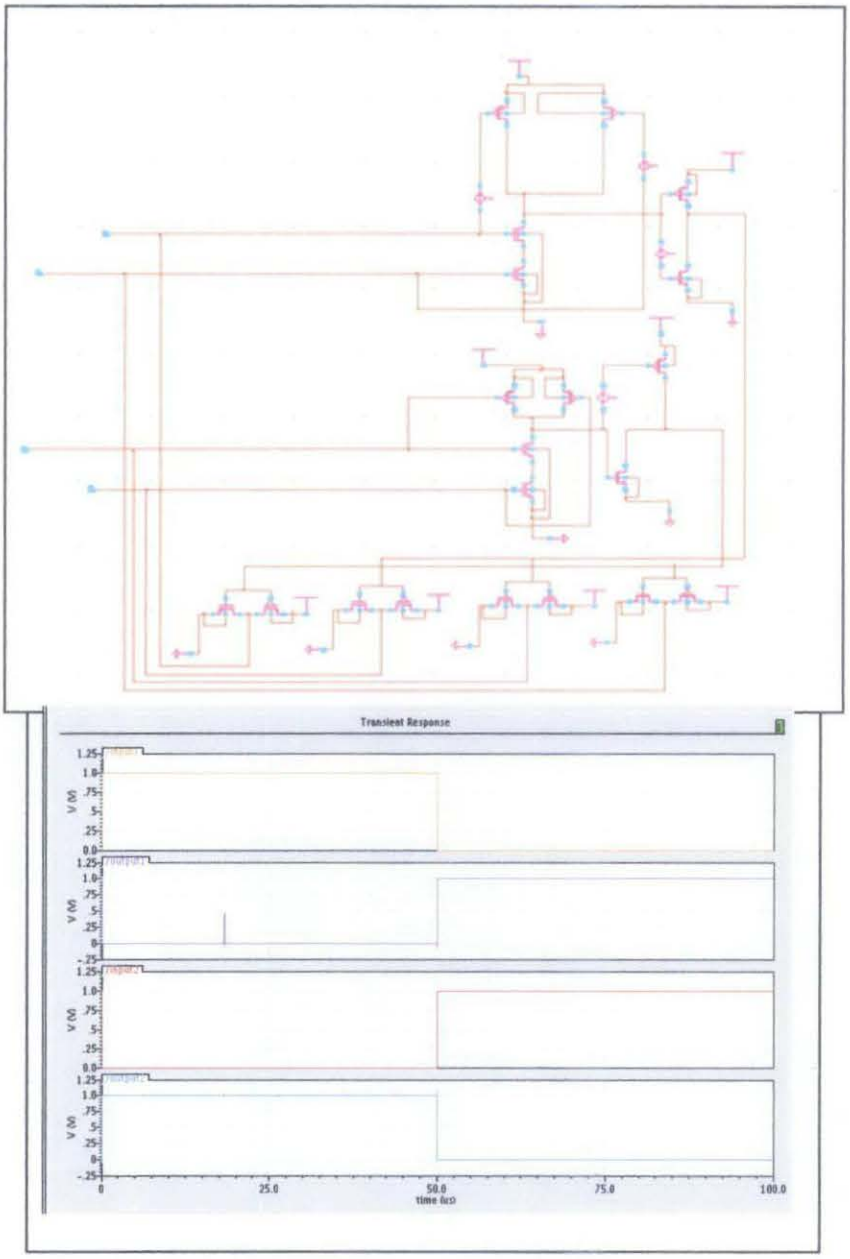

Figure 35: MRF Inverter schematic and simulation with some noisy transistors.

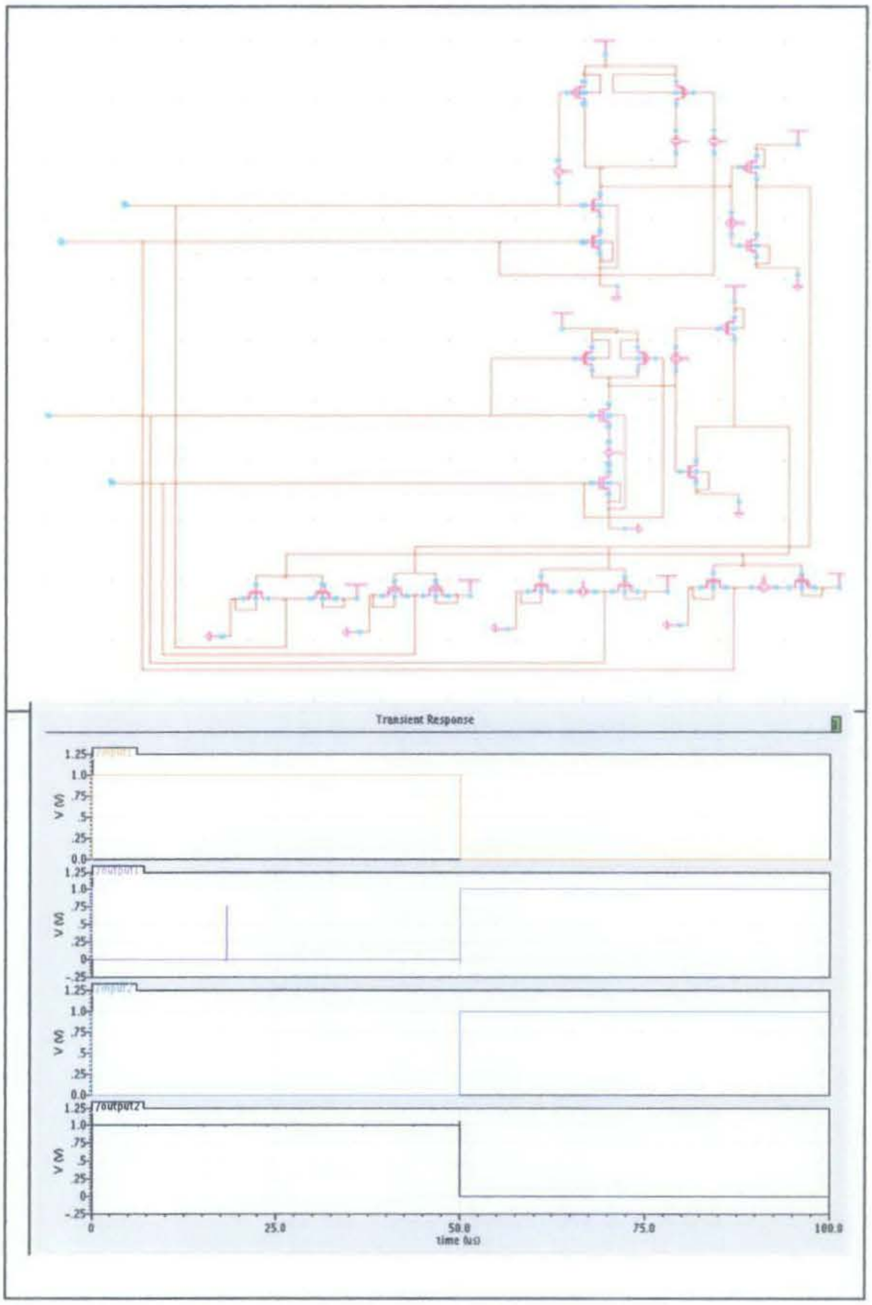

Figure 36: MRF Inverter schematic and simulation with some more noisy transistors.

In all the figures above (34, 35, and 36), the output is not noisy no matter where the noise source is connected. But as I mentioned previously for first MATLAB function, if the noise source connects to one of the output feedback transistors, it will result in a noisy output voltage. The result is shown in Figure 37.

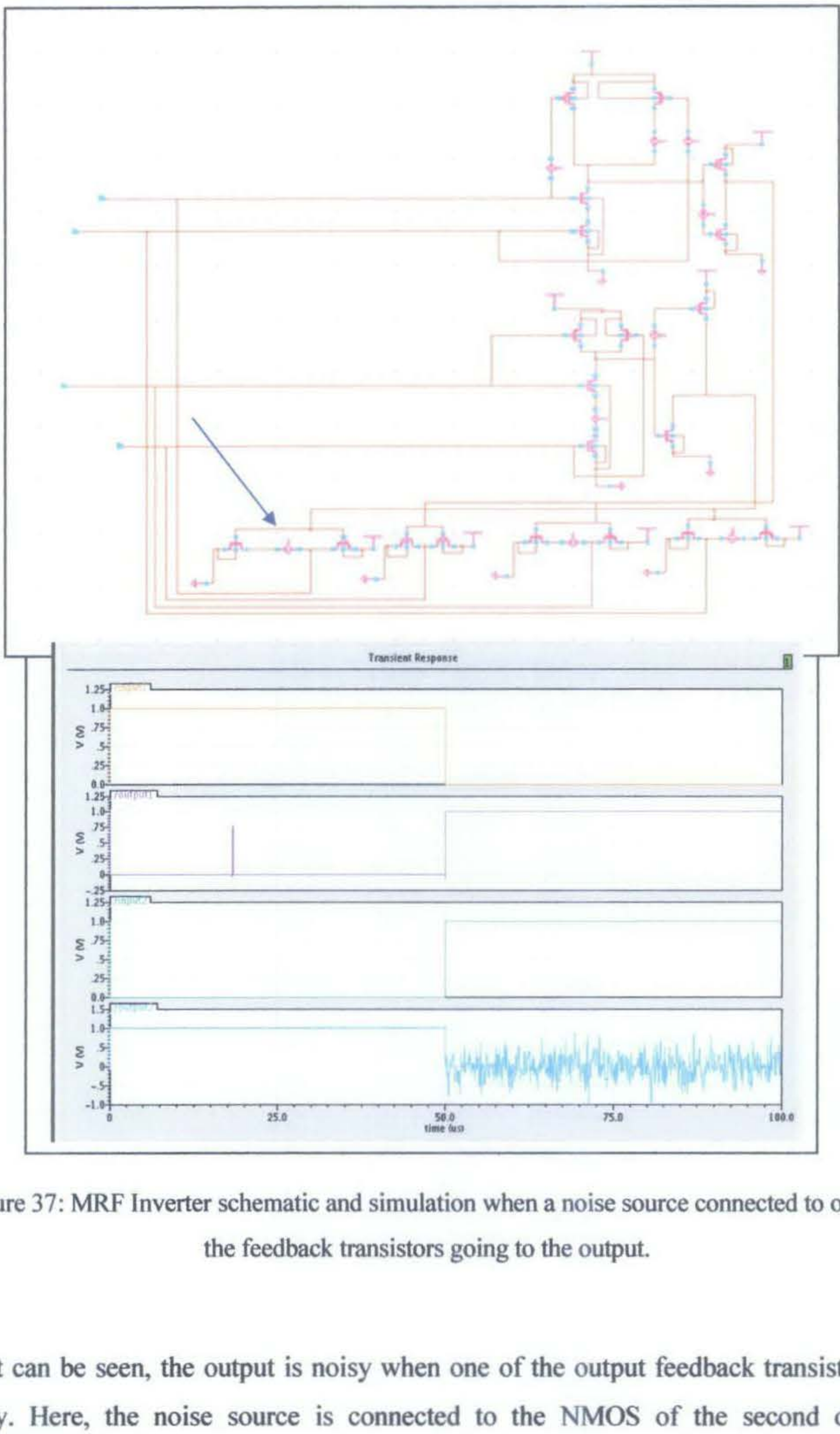

Figure 37: MRF Inverter schematic and simulation when a noise source connected to one of the feedback transistors going to the output.

As it can be seen, the output is noisy when one of the output feedback transistors is noisy. Here, the noise source is connected to the NMOS of the second output feedback transistor. As we know the NMOS is on when the input is high, so the part of the output related to vin2=1v (s_4'=1) is noisy, which is when the vout2=0 (s_5'=1). The reason is the same as discussed previously, if an error occurs in any of the feedbacks connected to any of the output paths, it will directly affect the output.

Figure 38 is when the noise source is connected to the output voltage; it is the same as connecting the noise source to each one of the feedback transistors going to that output source.

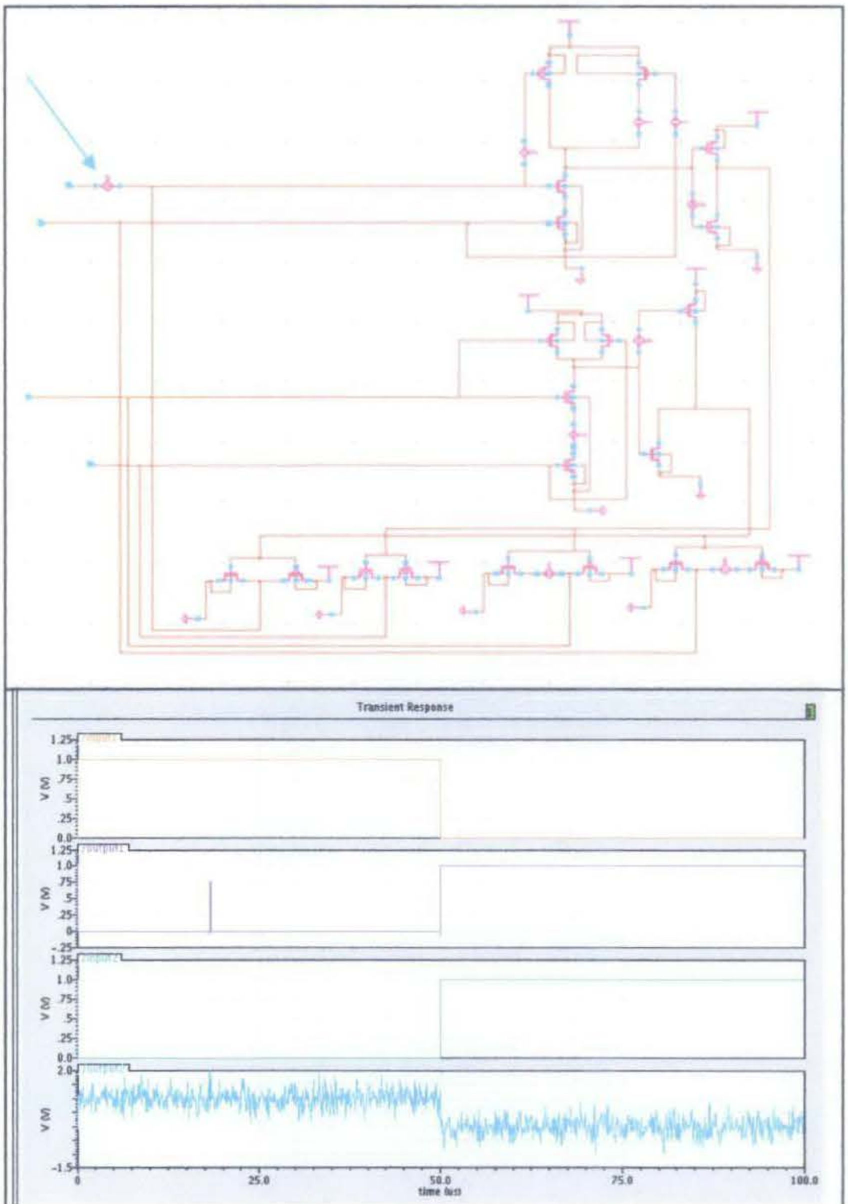

Figure 38: MRF Inverter schematic and simulation when the noise source is connected to the output (v2)

As it can be observed, the respective output is noisy all the way, when the noise source is connected to the output voltage.

As for the conclusion, it can be stated that both inverters have the same result in ideal conditions (error-free conditions) and they both invert the input without any fault or mistake.

In situation of noisy conditions, it was proved that the MRF circuit has much higher noise immunity in comparison with the standard inverter. Both MATLAB functions had the same results, in both methods the MRF circuit was tolerance to noise except the time when an error occurred in the output feedback transistors. But, in the case of the standard transistor, it is always noisy when there is an error in any part of the circuit.

For both inverters, the noise tolerance can be improved by reduction of the random variables voltage range and the obtained results will be less faulty.

Considering the point that MRF inverter contains much more transistors compared with standard inverter, it may come across the mind that, the size of the circuit is bigger and so it is not reliable. But, we have to consider that even though the number of transistors is much more, but the size of each transistor is reduced to micro, and so nano ranges, which in general result in a smaller size of circuit. This means that the MRF circuit is more reliable in comparison with standard inverter even though it has higher number of transistors due to the fact that the noise is eliminated and the size is reduced in the overall view.

At the same time, it should be considered that the inverter circuit is a small circuit, when we are dealing with big and complex circuits, the MRF theory may not be a good option since it will result in a more complex and complicated circuit which is not reliable any more.

CHAPTER 5
CONCLUSION

In this report, the comparison between standard and MRF inverter for ideal condition (error-free condition) and noisy condition has been done. The MRF probabilistic model provides a framework for designing CMOS circuits that can operate effectively under situations of extreme noise conditions.

The simulation results for both Standard-Inverter CMOS circuit and MRF-Inverter CMOS circuit was shown using Cadence tools and it was proven that the MRF inverter has much higher noise immunity in comparing with the standard inverter.

Finally, FYP has met all the objectives and it has strengthened my theoretical & technical base and enabled me to integrate theory with engineering practice.

REFERENCES

(1) Kundan Nepal PHD Thesis "Designing reliable nanoscale circuits using principles of Markov Random Fields". B.S., Trinity College, 2002 M.S.E.E., University of Southern California, 2003.

(2) K. Nepal, R. I. Bahar, J. Mundy, W. R. Patterson, and A. Zaslavsky. Designing logic circuits for probabilistic computation in the presence of noise. In Proceedings of Design Automation Conference, June 2005.

(3) K. Nepal, R. I. Bahar, J. Mundy,W. R. Patterson, A. Zaslavsky "Designing nanoscale logic circuits based on Markov random fields". Brown University, Division of Engineering, Providence, RI 02912.

(4) NAM-JIN OH, HYUNG-CHUL CHOI, NGUYEN TRUNG KIEN, JIN-YOUNG CHOI, "0.25 um CMOS Transistor and Inductor tet pattern layout for RFIC applications" SANG-GUG LEE School of Engineering, Information and Communications University, P.O. Box 77, Yuseong, Daejeon, 305-732, Korea.

(5) http://www.allaboutcircuits.com/vol_4/chpt_3/7.html

(6) R. I. Bahar, J. Mundy, and J. Chen, "A Probabilistic-based Design Methodology for Nanoscale Computation", *International Conference on CAD*, Nov. 2003.

(7) file:///C:/Documents%20and%20Settings/ark/Desktop/fypsem2/matlabresults/Adding%20noise%20to%20signals%20in%20SPICE%20%C2%AB%20Erin%20Taylor.htm

APPENDICES