AN FPGA-BASED RECONFIGURABLE DIGITAL CHIP TESTER

By

EEO AI TING

DISSERTATION

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

AN FPGA-BASED RECONFIGURABLE DIGITAL CHIP TESTER

by

Eeo Ai Ting

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

Approved:

_____

Dr. Fawnizu Azmadi Hussin
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2010

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.

_____

Eeo Ai Ting

ABSTRACT

This project presents the project work and results of an FPGA-based Reconfigurable Digital Chip Tester. The aim of this project is to achieve a reconfigurable, user-friendly and cost-effective digital chip tester for users to perform chip testing on the most commonly used digital ICs. The project adopts the software-defined approach, and is implemented on an FPGA which makes it versatile and reconfigurable. Functional testing is employed in the test approach to verify the functionality of the device under test. This project emulates a traditional digital chip tester, however with improvements made in terms of its versatility and cost-effectiveness. One advantage of this tester is it is able to test each output of a device individually which the traditional tester cannot perform. This will allow the user to still be able to use the functioning gates of a chip while avoiding the faulty ones, hence not putting the entire chip to waste. Due to its reconfigurability and expandability, it is proven to be a more cost-effective solution in the long run comparing to a traditional digital chip tester that is not reprogrammable.

ACKNOWLEDGEMENT

I would like to express my highest appreciation to those who has contributed greatly towards making this project a success. First and foremost, I would like to thank my project supervisor, Dr. Fawnizu Azmadi Hussin for his guidance and tutelage throughout the development of this project. Many thanks for his valuable suggestions and advices as well as for his effort in reviewing the reports written throughout the project.

I would also like to thank the Digital Design Lab technologist, Mr. Badrunizam, for his coordination in leasing of Altera's University Program 2 (UP2) development platform for the project development purpose, as well as for the leasing of the LEAPER-1 Digital IC Tester, for presentation demonstration purpose. I also wish to thank lab technologist Ms. Siti Hawa for her assistance in assembling proper connector tools for the prototype development.

I also wish to express my gratitude towards lecturers whose names are not mentioned here, yet they have contributed their opinions and valuable suggestions towards the improvement of this project. Thanks to the University for providing valuable resources which contributed immensely toward the completion of this project.

Last but not least, I would also like to thank my friends and family for their encouragement and support throughout this project.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

VLSI	Very-large-scale Integration
IC	Integrated Circuit
ATE	Automated Test Equipment
TTL	Transistor-transistor Logic
CMOS	Complementary Metal-oxide-semiconductor
FPGA	Field-programmable Gate Array
I/O	Input/Output
SoC	System on Chip
DIP	Dual-inline Package
PC	Personal Computer
HDL	Hardware Description Language
VHDL	VHSIC Hardware Description Language
VHSIC	Very-high-speed Integrated Circuit
DUT	Device Under Test
LED	Light Emitting Diode
ZIF	Zero Insertion Force
UP2	University Program 2
RTL	Register Transfer Level

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Since the introduction of very-large-scale integration VLSI devices in the early 1980s, the complexity and density of digital circuits continue to increase; a single chip today can consist of millions of transistors measuring in nanometers. As circuit size increases with steadily decreasing transistors dimension, referred to as feature size, more quality and reliability are required, making the validation of VLSI circuits more and more challenging.

In any manufacturing industries, manufacturing defects are unavoidable in its manufacturing process. Hence in the electronics industries, IC testing is vital to separate a good chip from the bad. In the industry, IC testing is often done using automated test equipments (ATE) which are large, complex and very costly machineries. Likewise at the consumer end, there are digital IC testers to perform testing on digital chips. These digital IC testers come in different specifications, normally defined by the range of ICs it can support. Depending on the functionalities of the tester, the cost of a traditional digital IC tester usually comes at a price too high for an individual to own.

In research centers and educational institutions, users often only deal with a common range of digital ICs, which mainly are basic logic gates from the 74 series TTL and 74 series CMOS. Since not all chips come in perfect condition, a digital IC tester would be useful to determine a good chip from the bad. This will save users a lot of precious working time from using a faulty chip. In this project, the aim is to design an FPGA based digital IC tester that will perform functional testing on digital ICs targeting the commonly used range of ICs in research and educational centers.

1.2 Problem Statement

To test and tell apart a faulty chip from the good ones is important to save users from wasting precious time working on a faulty chip. Users from research centers and educational institutions only deal with a small range of digital ICs. Available digital IC testers in the market are often too costly for individuals to own, furthermore these traditional testers generally support a wide range of digital ICs, and this will come redundant to users from this market end.

Traditional testers in the market are designed to support only a specific range of devices defined by the vendor and they are not reprogrammable. Hence users generally cannot extend or customize the functions of these testers to accommodate their needs. Having said that, a user might need to own a few types of testers in order to perform testing on various kinds of devices. This will come at a high cost, and will again be redundant since not all of the functions will be frequently used. Therefore, a cost effective, user friendly and reconfigurable digital IC tester would be ideal for users from this market end to test digital chips.

Cost effectiveness and I/O re-configurability being the major factors here, leads to performing a study on designing a low-cost reconfigurable FPGA-based digital chip tester in this project.

1.3 Objective and Scope of Study

1.3.1 Objective

The objective of this project is to design a cost effective, reconfigurable test instrument based on FPGA to test basic logic digital ICs. Adopting a software-defined approach, this project aims to develop a versatile and user friendly tester for basic digital chips, providing a cost effective test instrument for users to perform digital chip testing in research laboratories.

1.3.2 Scope of Study

This study will encompass, but not limited to, the field of designing a system-on-chip (SoC) on an FPGA. The design of this tester supports 14-pins DIP basic digital logic ICs from the 74 series TTL and CMOS family. The devices tested in this project are 74LS32, 74LS00, 74LS02, 74LS08, 74LS86 and 74LS386. The testing model used will be based on the functional fault.

CHAPTER 2
LITERATURE REVIEW

2.1 Theory

2.1.1 Digital IC Tester

Digital IC tester is a test instrument to test the digital ICs in order to verify faulty gates from the good gates. The primary purpose of this digital IC tester is that it can easily check the IC instantly and to show discrepancy results if there were any. The manual operation to test a digital chip is a time consuming and tedious process [1]. The procedure is such that each individual IC is tested on a prototype board by making necessary connections manually and verifying the outputs for each gate by the truth table.

A general digital IC tester available in the market for instance is the LEAPER-1 Handy Digital IC Tester developed by LEAP Electronic Co., Ltd, Taiwan [3]. This digital IC tester is capable of testing ICs with 14 to 24 pins, from the TTL74xxx, CMOS40xxx, CMOS45xxx, DRAM41xxx and DRAM44xxx series. To perform a test on a digital chip of the aforementioned series, the IC is inserted into the ZIF socket on the tester, and then the user can select the device number by pressing the UP and DOWN keys, or by pressing the AUTO key. The LP-1 Digital IC Tester will then compare the inserted component to the components in its database. The LP-1 will return the first component number from its database, which matches the inserted device. This is not necessarily the correct component. By pressing the AUTO key again, the LP-1 will search the remainder of its database.

When a component has been found, the display will show "[X] NNNNN FIND" where X stands for the IC type and NNNNN for the IC

number. If a component does not match any device in the database of the LP-1, the display will show "** NOT FOUND **". The IC is then tested by pressing the TEST key. If the device is functioning, the display shows "[X] NNNNN PASS". If the device test fails, the display will show "[X] NNNNN FAIL" (X stands for the IC type and NNNNN for the IC number). [3]

Figure 1: LEAPER-1 Handy Digital IC Tester.

The LEAPER-1 digital IC tester is categorized under the lower end of the digital IC testers available in the market due to the limited device family it can support. Furthermore, they are of the most basic digital logic chips. Also being considered as a traditional tester, it is fairly easy to operate which comes very handy in performing IC checking. However for a low-end tester like such, it comes at a price range of about USD 275 which is evidently too much for a student or a research fellow to own. Furthermore a student most probably will not be fully utilizing all the features that come with it. The LEAPER-1 digital IC tester is selected to be reviewed out of the vast range available because this model is used in the digital electronics laboratory of the author's university, Universiti Teknologi PETRONAS.

2.1.2 Traditional versus Virtual Instruments

Instruments today are fundamentally based on two types of architecture, virtual and traditional. A traditional or also known as a stand alone instrument usually comes with a fixed user interface, specifically defined by the vendor. With the software processing and the user interface fixed in the instruments itself, the instrument can be updated only when and how the vendor desires to, for instance via a firmware update. Hence, it is impossible for the user to perform functionalities not included in the list of functions of a traditional instrument. [4]

On the other hand, a virtual instrument, which is based on a software-defined approach, makes the raw data from hardware available to users which then they can customize the usability according to their requirements. Virtual instruments, by virtue of being PC-based, take advantage of the latest technology incorporated into off-the-shelf PCs. Its hardware functionality is characterized through user-defined software running on a host multicore processor. In the engineering or research industry, a developer's need, application and requirements change rapidly, thus flexibility is essential to create their own solutions. By adapting a virtual instrument to the user's particular needs, the entire device does not have to be replaced because the application software is installed on the PC and the there are a wide range of hardware plug-in available. The flexibility of defining one's system in a virtual or modular manner frees the user from vendor-defined systems, which emerge as a cost effective solution in the long run. [5]

2.1.3 Reconfigurable Instruments using FPGA

Test systems are reconfigured for endless reasons – from adapting to new test requirements to accommodating instrument substitutions. Having mentioned that software-defined instrumentation is based on a modular architecture, it enables a high degree of reconfigurability. The software-defined architecture needs to be flexible enough to incorporate user-programmable hardware; in this case, an FPGA is often used. An architecture

like this creates the environment where data can be acted upon in real time on the FPGA and/or processed centrally by the host processor.

FPGAs are reprogrammable silicon chips at the highest level. By using prebuilt logic blocks and programming routing resources, these chips can be configured to implement custom hardware functionality. Having said that FPGAs are completely reconfigurable, they can instantly take on a new personality when recompiled with a different configuration of circuitry. Years back then, FPGA technology was only available to engineers who have deep understanding of digital hardware design software, for instance hardware description languages which use low-level syntax to describe hardware behavior, like Verilog or VHDL. However, with new high-level design tools emerging, the FPGA programming interface has changed to be more user-friendly or more object oriented based. Graphical diagrams or even C code can be converted into digital hardware circuitry. Users without extensive knowledge in digital hardware design software can now deploy the intelligence of the FPGA to develop more efficient solutions. [6]

2.1.4 Functional Testing

A functional test exercises the actual operation of the digital logic design through the various functional operations that it is intended to undertake. Functional testers apply a set of stimuli to input pins of a DUT and sample the response at the output pins after sufficient time has elapsed to permit signals to propagate and settle out. The tester will then compare the sampled response to the expected response which will determine whether the DUT responded correctly to the applied stimuli. [2] For complex digital circuits and systems, this can be extremely time consuming and hence costly. Since only basic logic gates will be tested in this project, functional test is a suitable approach.

CHAPTER 3
METHODOLOGY

3.1 Process Flow

This project would be conducted based on the process flow as illustrated by the following figure.

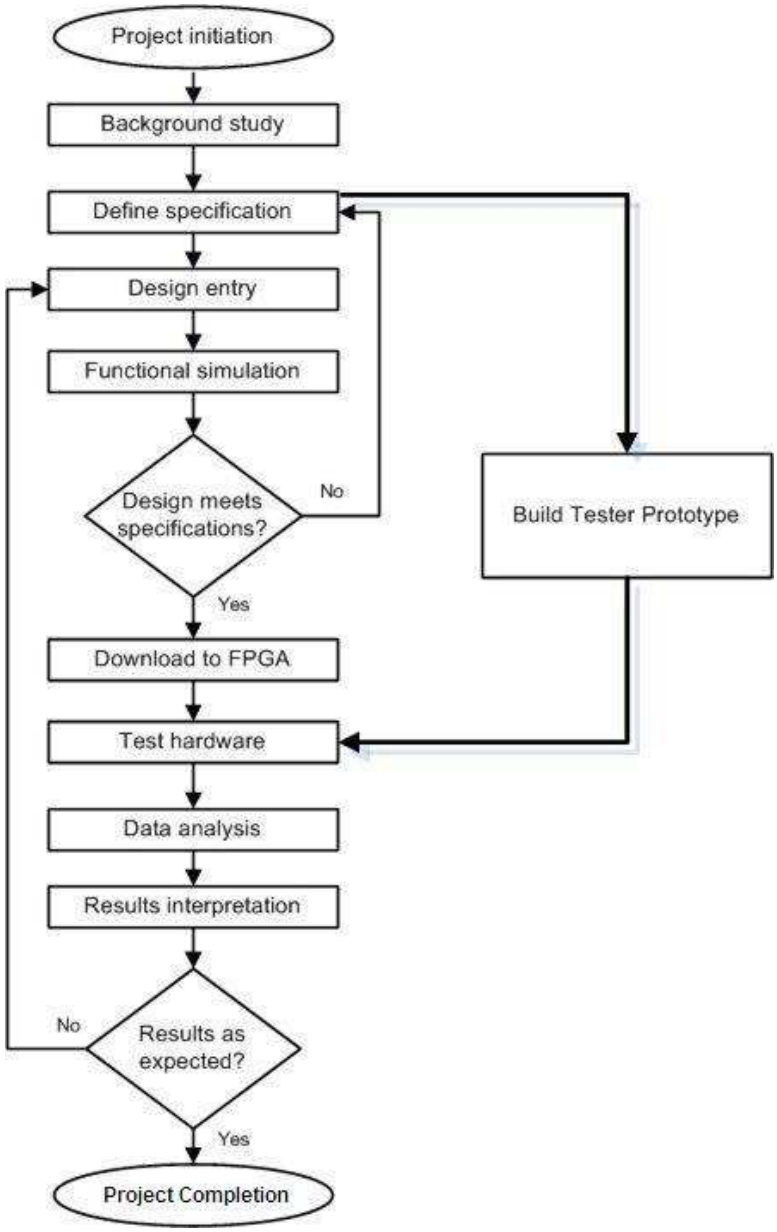


Figure 2: Process Flow Chart.

3.2 Tools and Equipments Used

The main tools used for this project are the Quartus II design software and the Altera University Program 2 Development Platform.

Quartus II software provides a platform for synthesis of HDL designs, to compile designs, perform timing analysis, examine RTL diagrams, simulate design's reaction to different stimuli and to configure the target device with the programmer. The FPGA used in this project will be the FLEX 10K device on the Altera UP2 development platform.

3.3 DUT Board Design Specifications

This section outlines the design of the device-under-test (DUT) board where the digital chip to be tested is placed to perform a test.

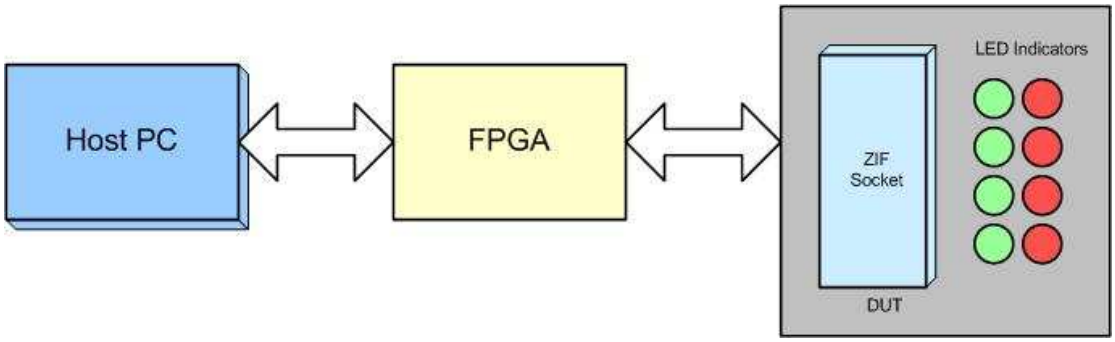


Figure 3: Design structure of the test system.

Figure 3 illustrates the structure of the test system. The test system consists of three main components, namely the host computer, the DUT board and the FPGA which interfaces the DUT board with the host computer. The DUT board comprised of a ZIF (zero-insertion-force) socket which holds the digital chip to undergo testing. The test results are indicated by a set of green and red LEDs; of which the green LEDs indicate the test is passed while the red LEDs denote failed results.

The FPGA functions as the kernel to process the test system. After it is programmed with the test design of the targeted digital chip, a same set of input stimuli will take place in the test design and be loaded into the input pins of the chip.

The output data from the DUT will then be retrieved and compared against the expected results defined by the test design. Subsequently, the pass or fail results can be observed on the LEDs.

The DUT board in this project is made to have a set of green and red LEDs. The number of LEDs to be placed on the board can be varied, depending on the number of outputs required to be tested. For this project there will be 4 of each green and red LEDs. These LEDs indicate the results of each output pin of the DUT according to the sequence. In other words, this tester prototype is capable of testing each output. This will come handy for the targeted devices in this project as they consist of a few gates on one chip.

3.4 Design Entry Methodology

This section describes the methodology to create the test design of each device, based on one example, which is the 74LS32 quad 2-input OR gate. The same procedures are employed to create the test designs for all the targeted devices described in this project.

3.4.1 Describing the Functional Behavior

The functional behavior of the logic gate in the targeted device, in this case an OR gate, is described in Verilog. This defines the expected output which is then used to compare the output results from the DUT with. The functional behavior is simply described by the truth table of the OR function as displayed in Figure 4:

```
1
2  module truthtable(A,B,C);
3
4      input   A,B;
5      output  C;
6      reg     C;
7
8      always@(A,B)
9
10         if (A==0 && B==0)    C<=0;
11         else if (A==0 && B==1)   C<=1;
12         else if (A==1 && B==0)   C<=1;
13         else C<=1;
14
15 endmodule
```

Figure 4: Functional behavior of OR gate.

3.4.2 Describing the "Compare" Function

This compare function is basically used to compare the results from the output of the DUT with the expected results from the truth table mentioned in section 3.4.1. When the results match, a logic HIGH will be sent to 'pass' pin hence turning on the green LED while the red LED remain off. When the results do not match, likewise, a logic HIGH will be sent to 'fail' pin, thus turning on the red LED while the green LED will be off. The compare function is shown in Figure 5 as follow:

```
1  module compare(C,op,pass,fail);
2
3      input   C,op;
4      output  pass,fail;
5      reg     pass,fail;
6
7
8      truthtable  truthtable(C);
9
10     always@(op,C)
11
12         if (op==C)
13         begin
14             pass<=1; fail<=0;
15         end
16
17         else
18         begin
19             pass<=0 ; fail<=1;
20         end
21
22 endmodule
```

Figure 5: Compare block.

3.4.3 Top Level Module

Next, the top level module is written as shown in Figure 6, to describe the overall test function by instantiating the earlier two modules. The number of 'compare' modules instantiated depends on the number of gates on the chip under test. The function of each gate is tested individually to obtain separate results. *A* and *B* represents the input vector from the truth table module stated in section 3.4.1. Inputs *op1…4* are the individual outputs of each gate of the DUT, and they are taken in as inputs to be compared with *C*, which is the desired results. The compared results are indicated by *pass1…4* and *fail1…4* respectively.

```
1  module  test74LS32_OR   (A,B,op1,op2,op3,op4,
2                          pass1, pass2, pass3, pass4,
3                          fail1, fail2, fail3, fail4);
4
5      input   A,B,op1,op2,op3,op4;
6      output  pass1,pass2, pass3, pass4,
7              fail1, fail2, fail3, fail4;
8
9      wire C;
10
11     truthtable  truthtable1 (A,B,C);
12     compare     compare1(C,op1,pass1,fail1);
13     compare     compare2(C,op2,pass2,fail2);
14     compare     compare3(C,op3,pass3,fail3);
15     compare     compare4(C,op4,pass4,fail4);
16
17 endmodule
```

Figure 6: Top level module.

3.4.4 Functional Simulation

Functional simulation is performed on each module to verify its functionality. This will be further discussed under the "Results" section.

3.4.5 Creating Symbol Block

The top level module is then converted into a symbol block to illustrate the input and output nodes as shown in Figure 7. This is done to ease the upcoming pin assignment step.

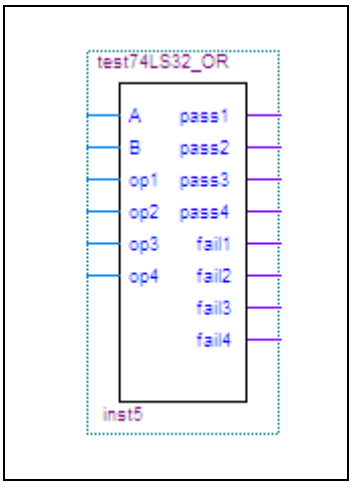


Figure 7: Symbol block diagram to test 74LS32.

3.4.6 Input Stimuli and Pin Assignment

The input test vector is obtained from a 2-bit counter, and pin assignment is assigned accordingly based on the schematic layout of the targeted device. Figure 8 displays the schematic block diagram of test system with each pin out assigned to their relevant locations.

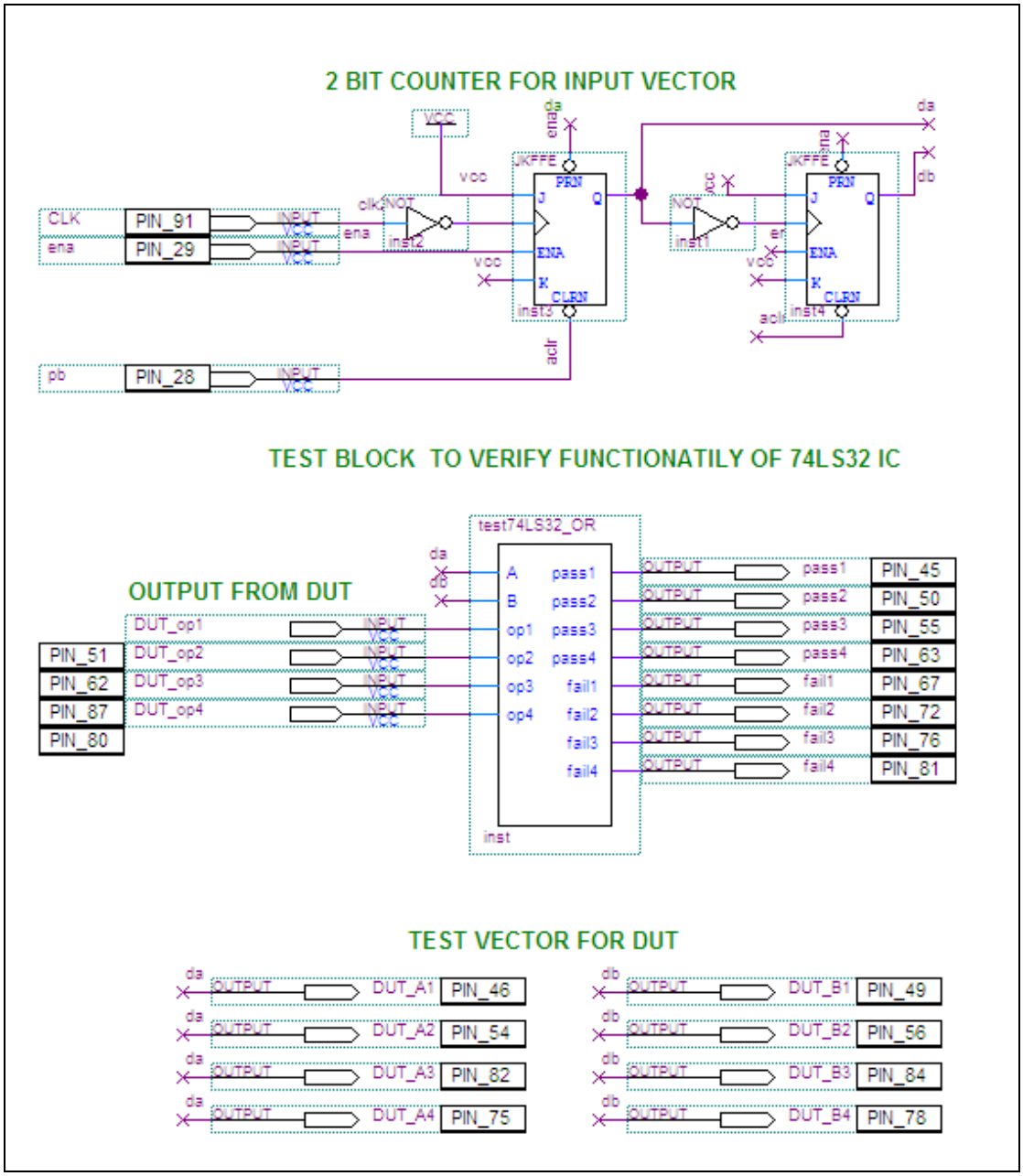


Figure 8: Schematic block diagram of test system with pin assignment.

3.4.7 Programming the FPGA

After performing pin assignment, the design is compiled and the configuration file is downloaded to the FPGA via the ByteBlasterII cable to program it. The device can now be tested on the DUT board. More will be discussed in the "Results & Discussion" section.

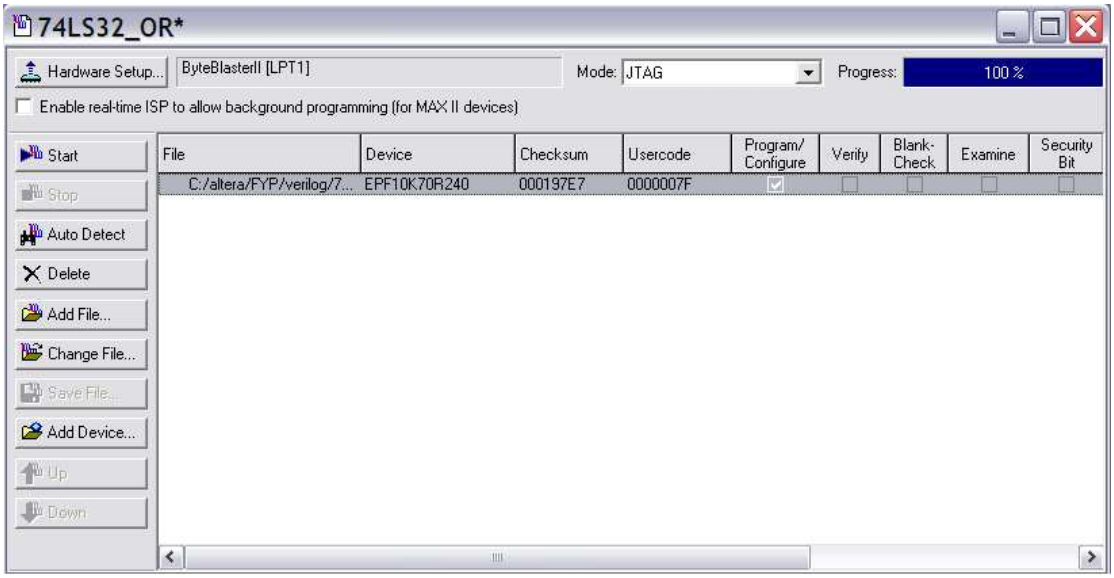


Figure 9: Programming the FPGA.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Results

4.1.1 Functional Simulation Results

Functional simulation is performed to verify the functionality of the module written in Verilog. This module is intended to yield the expected results of an OR gate function, whereby the output results from the DUT would be compared against with. Figure 10 shows the simulation wave form of the module earlier described in section *3.4.1 (Figure 4)*, it is verified to be true comparing to the truth table displayed in Table 1.

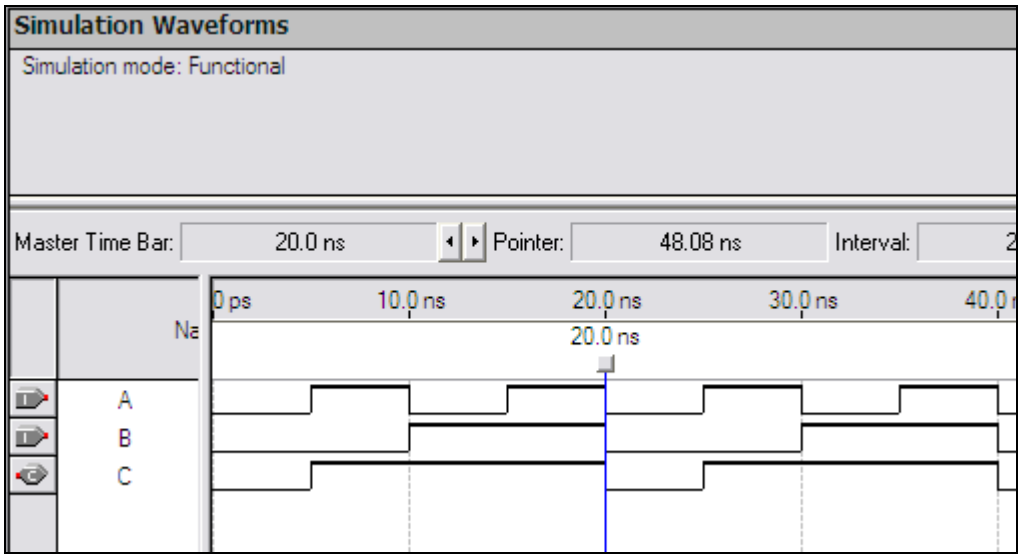


Figure 10: Functional simulation of OR function.

Table 1: Truth table for OR function.

Truth Table (OR)				
A	0	1	1	1
B	0	0	0	1
C	**0**	**1**	**1**	**1**

Next, the top level module describing the overall test function is simulated, again to verify its intended function. Figure 11 displays the simulation waveform on this module and its results is tabulated in Table 2.

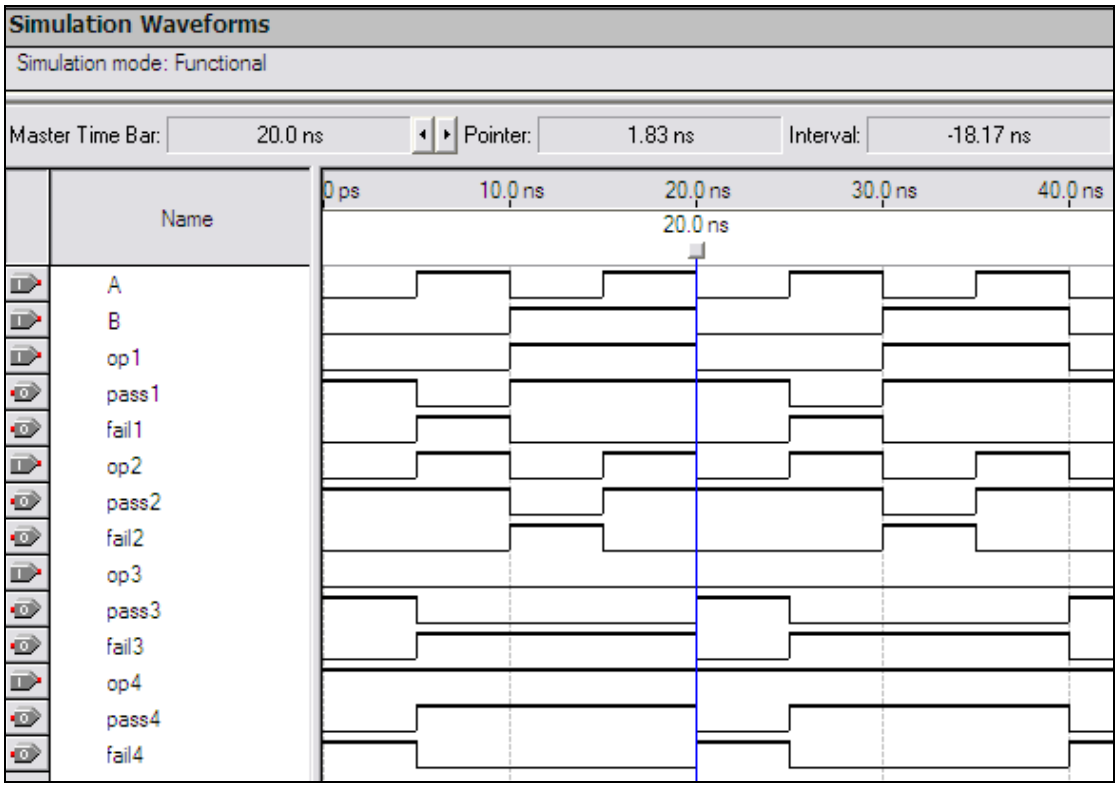


Figure 11: Functional simulation of the overall test function.

The input vector is denoted by *A* and *B* respectively. Pin name *op1*, *op2*, *op3* and *op4* (highlighted in grey) represent the output pin from the DUT, whereby they would be compared with the expected results of output *C*, as shown in the row highlighted in orange. Since the outputs from DUT cannot be acquired to be simulated, they are assigned with random value for functional verification purposes. As noted, when results from *op1…4* match *C*, *pass* will be 1 while *fail* will be 0, vice versa for the case where they do not match.

Table 2: Simulation results tabulated.

Pin Names	Simulation Results			
A	0	1	1	1
B	0	0	0	1
Expected	**0**	**1**	**1**	**1**
op1	0	0	1	1
pass1	1	0	1	1
fail1	0	1	0	0
op2	0	1	0	1
pass2	1	1	0	1
fail2	0	0	1	0
op3	0	0	0	0
pass3	1	0	0	0
fail3	0	1	1	1
op4	1	1	1	1
pass4	0	1	1	1
fail4	1	0	0	0

4.1.2 Results on DUT Board

The test results of chip 74LS32 performed on the DUT board is demonstrated in Figure 12. The left figure displays the results of a functioning chip, by which it is denoted by the lighting of all green LEDs. The right figure displays the results of testing a faulty chip. It is observed that the 4th output or the 4th gate of this chip is not functioning, indicated by the red LED, whereas the rest of the gates are in good function, indicated by the green LEDs. All the chips under test in this project have also been tested manually to validate the results obtained from the tester prototype.

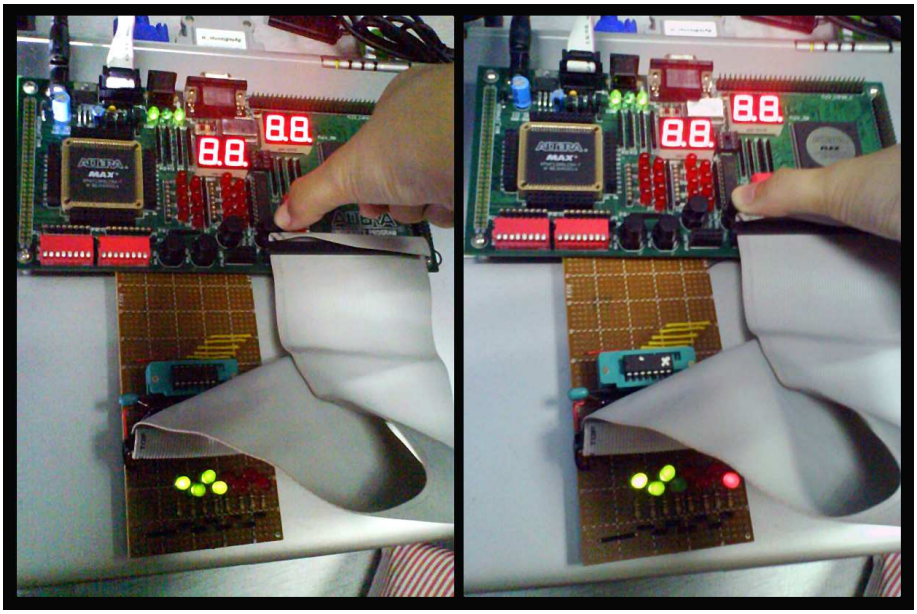


Figure 12: Results observed on the DUT board for device 74LS32.

4.2 Discussion

The working concept of the FPGA-based reconfigurable digital chip tester is demonstrated in Figure 13. First the same set of test stimuli are downloaded simultaneously into the test design and the DUT, the output from DUT will then be compared with the expected results from the test design. Subsequently, a pass or fail results will be indicated through LEDs of green and red respectively.

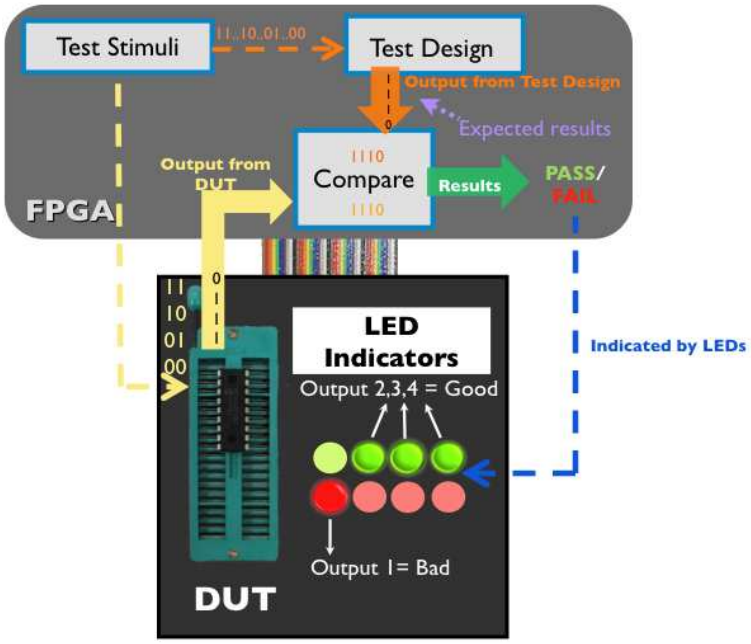


Figure 13: The working concept of FPGA-based digital chip tester.

All configurations such as pin assignment and any possible changes with the design entry are defined in the software. The hardware part only consists of the ZIF socket to load or unload a device, and LED indicators.

One main advantage of this tester compared to the traditional tester is, each gate on the device can be tested individually while the traditional tester only yields a "PASS" or "FAIL" indication. This will allow the user to still be able to use the functioning gates of a chip while avoiding the faulty ones, hence not putting the entire chip to waste. The user would most likely discard of a chip tested "FAIL" on the traditional tester because more time and trouble would be required if the gates were to be tested individually and manually.

Figure 14 and Figure 15 illustrates the results for the case when all the all outputs are functioning and when all the outputs are not functioning respectively.

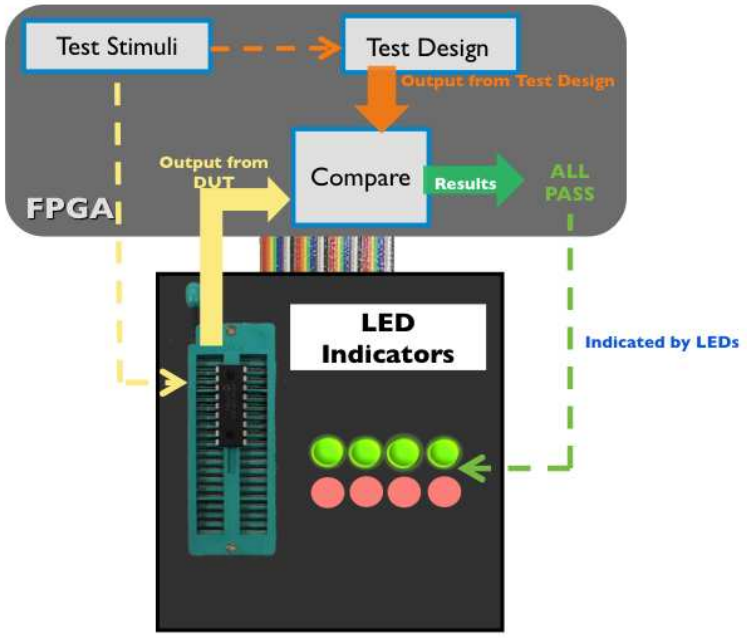


Figure 14: When all output pass.

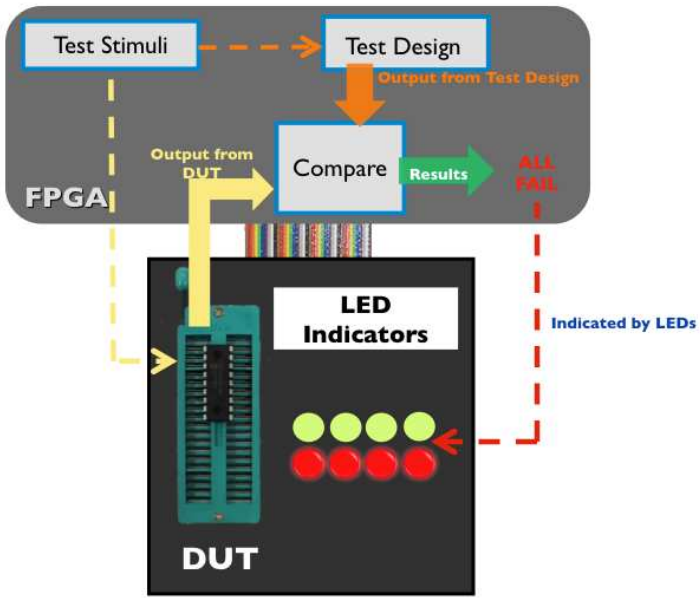


Figure 15: When all output fails.

4.3 Cost Comparison

The cost of the FPGA-based tester is considered by taking into account the cost of the market price of the Altera UP2 board which is quoted at USD 99 from the source referenced at [8], approximately equivalent to MYR 317 after currency conversion. The cost of the DUT board which consists of a few basic electronics components is estimated to be approximately MYR 6, which comes to a total of MYR 323 for the overall. Cost comparison of both instruments is tabulated in Table 3.

Table 3: Cost comparison between FPGA-based and traditional digital chip tester.

FPGA-Based Digital IC Tester		LEAPER-1 Digital IC Tester	
Altera UP2 Board (FPGA)	(USD 99) MYR 317	Unit Price	(USD 275) MYR 881
DUT Board	MYR 6		
Total	**MYR 323**	**Total**	**MYR 881**

The cost of the LEAPER-1 digital chip tester is quoted at USD 275 based on source referenced at [9], which approximately equals to MYR 881 after conversion.

Considering if a user is selecting in between these two devices to perform the same function which is to do chip testing, if the user were to design his or her own tester based on this project, the price the user would have to pay is MYR 323 as compared to getting the LEAPER-1 digital IC tester at MYR 881. Evidently, the FPGA-based tester is a more cost-effective solution, as it is at least 63% lower in cost as compared to the LEAPER-1 tester in this case.

$$\frac{881 - 323}{881} \times 100\% = 63.34\%$$

Moreover, it should be noted that the FPGA is multipurpose and can be utilized for many other applications, not solely for this application only. While the function of LEAPER-1 is fixed, that is to only perform digital chip testing. This clearly shows that the FPGA-based instrument is a lot more cost-effective in the long run.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In today's research and development, we constantly strive to achieve improvements in cost effectiveness and time efficiency. In this project, a reconfigurable, cost-effective and versatile test instrument has been designed and a basic prototype was developed. This project provides a practical approach to address the problem of wasting unnecessary time working on faulty chips.

It has the advantage of being able to test each output of a chip individually which the traditional tester cannot perform. This will allow the user to still be able to use the remaining functioning gates of a chip while avoiding the faulty ones, hence not putting the entire chip to waste. Due to its reconfigurability and expandability, it is proven to be a more cost-effective solution in the long run comparing to a traditional digital chip tester that is not reprogrammable.

This project hopes to bring forth a notable resource for future development in the field of software-defined test approach.

5.2 Recommendations

This project has the capability to expand to test any available digital IC of the TTL or CMOS family up to 32 pins. It could also be expanded to test more parameters besides functional testing. The project could also be improved so that it can perform the test on multiple devices by only having to download the test file once, a control button can be added onto the DUT board to select the targeted device to perform the test.

A graphical user interface (GUI) could be designed to further enhance as well as to automate the user. The purpose is to provide a user friendly, easy-to-use

experience by simplifying the test operation. Besides, the software-defined test approach could be adopted in testing new kinds of devices, especially for applications that require point-to-point I/O testing.

REFERENCES

[1] Kiran Kumar Kolli, Implementation of IC Tester, Gayatri Vidhya Parishad College of Engineering, JNTU, Visakhapatnam..

[2] Alexander Miczo. “Basic Tester Architectures,” in Digital Logic Testing and Simulation, 2nd ed., New Jersey: John Wiley & Sons, Inc., 2003, pp. 284.

[3] LEAP Electronic Co., Ltd, “LEAPER-1 Handy Digital IC Tester User Manual”, 2006, http://www.leap.com.tw/english/pdf/LEAPER-1.pdf.

[4] National Instruments, “Understanding a Modular Instrument System for Automated Test” 29 January 2010, White paper, Retrieved 23 April 2010 from URL: http://zone.ni.com/devzone/cda/tut/p/id/4426#toc0.

[5] Chandra Nair, “Virtual Instrumentation for User-defined Apps” 16 March 2005, Retrieved 23 April 2010 from URL: http://www.eetindia.co.ni/ART_8800405592_1800003_TA_1f7bcf23.HTM

[6] National Instruments, “The Automated Test Outlook 2010 - In the area of I/O: Reconfigurable Instruments” 12 April 2010, Retrieved 26 April 2010 from URL: http://zone.ni.com/devzone/cda/tut/p/id/11287.

[7] Altera Corporation, “University Program UP2 Education Kit User Guide”, December 2004, version 3.1. URL: www.altera.com/literature/univ/upds.pdf

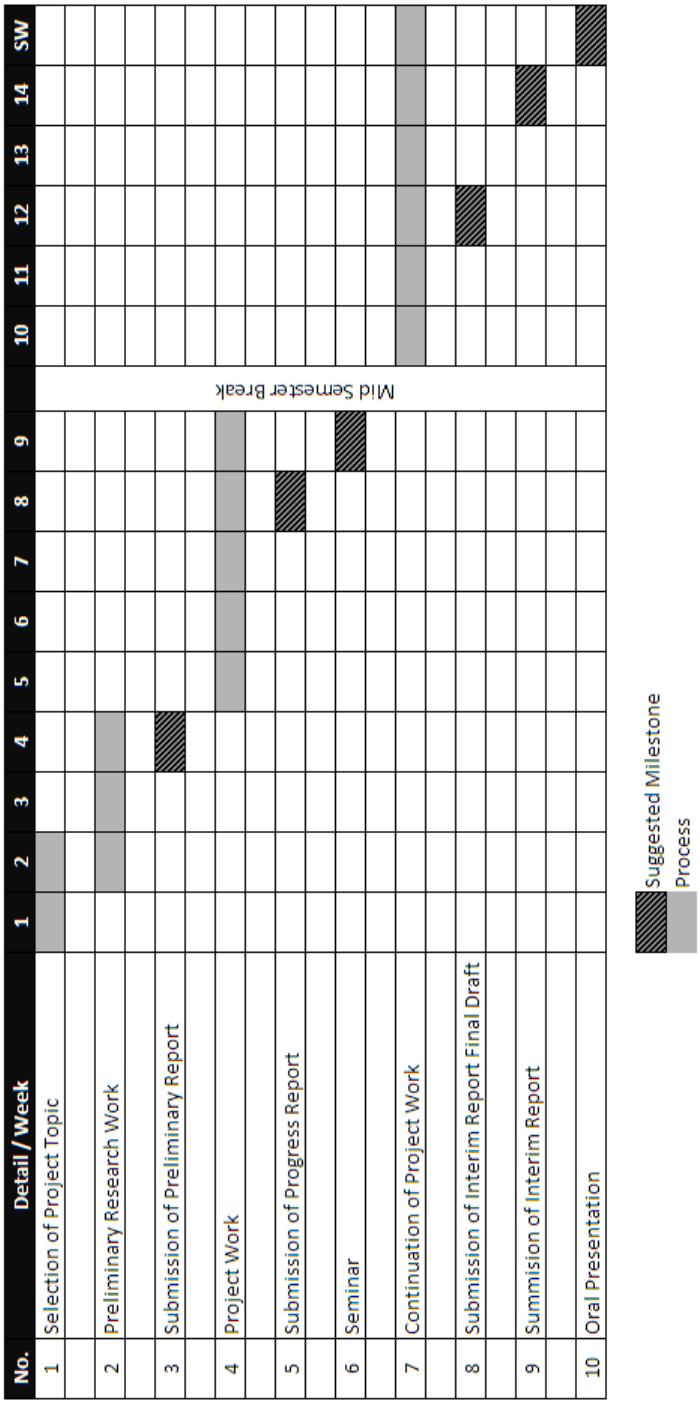
[8] Quoted price of Altera UP2 development board, Cited 26 April 2010 from URL: http://www.altera.com/education/univ/materials/boards/unv-up2-board.html

[9] Quoted price of LEAPER-1 digital IC tester, Cited 26 April 2010 from URL: http://www.capetronics.com/leaper1_ictester.htm

APPENDICES

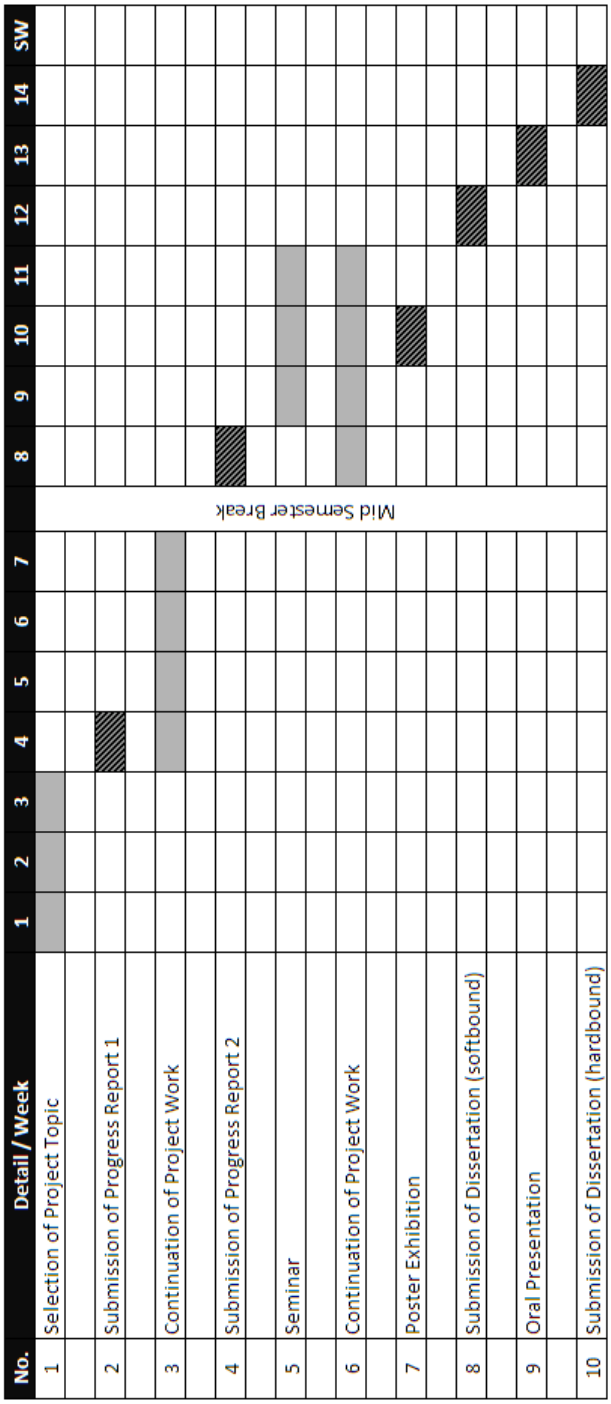
APPENDIX A

GANTT CHART OF FINAL YEAR PROJECT 1



APPENDIX B

GANTT CHART OF FINAL YEAR PROJECT 2



APPENDIX C
ALTERA UP2 EDUCATION KIT USER GUIDE

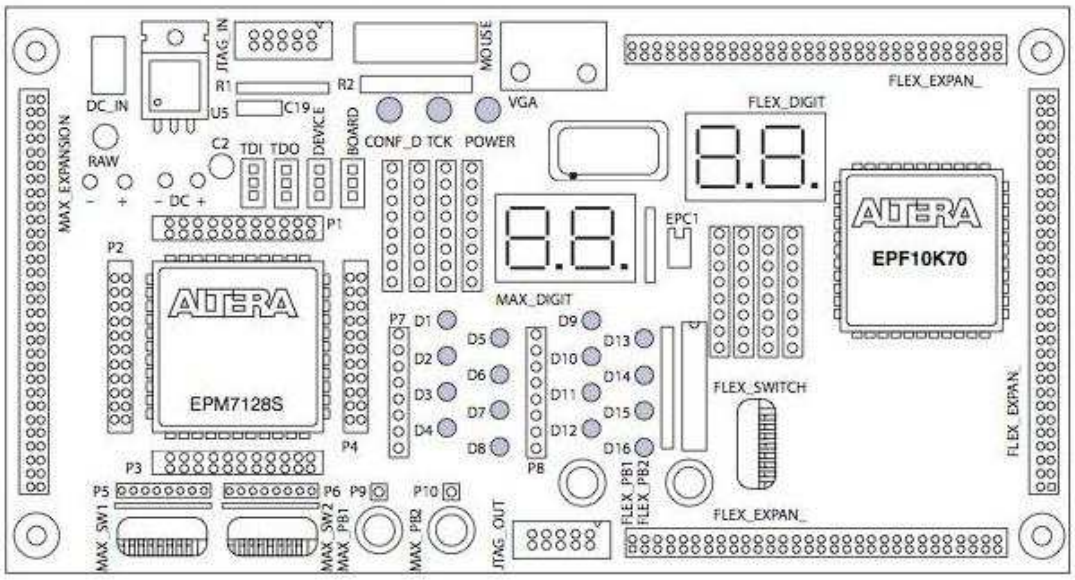


Figure 16: Altera UP2 Developemnt Platform Component Layout, from page 3 of **[7]**, "University Program UP2 Education Kit User Guide", December 2004, version 3.1. URL: www.altera.com/literature/univ/upds.pdf

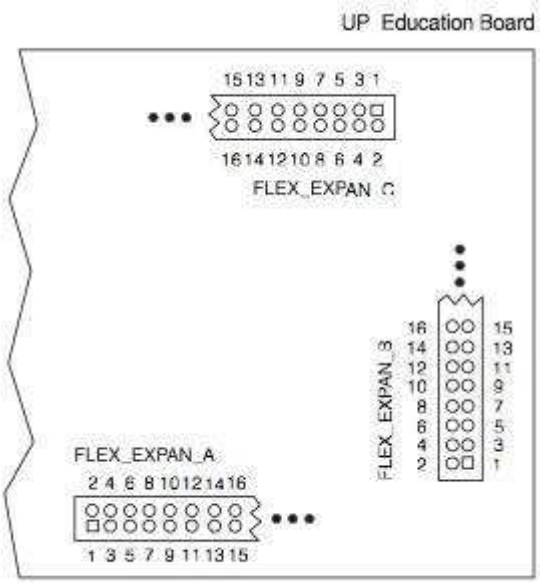


Figure 17: Numbering convention for FLEX_EXPAN_A, FLEX_EXPAN_B & FLEX_EXPAN_C, from page 14 of [7].

Table 4: FLEX_EXPAN_A Signal Names and Device Connections from page 15 of **[7].**

Hole Number	Signal/Pin	Hole Number	Signal/Pin
1	RAW	2	GND
3	VCC	4	GND
5	VCC	6	GND
7	No Connect	8	DI1/90
9	DI2/92	10	DI3/210
11	DI4/212	12	DEV_CLR/209
13	DEV_OE/213	14	DEV_CLK2/211
15	45	16	46
17	48	18	49
19	50	20	51
21	53	22	54
23	55	24	56
25	61	26	62
27	63	28	64
29	65	30	66
31	67	32	68
33	70	34	71
35	72	36	73
37	74	38	75
39	76	40	78
41	79	42	80
43	81	44	82
45	83	46	84
47	86	48	87
49	88	50	94
51	95	52	97
53	98	54	99
55	100	56	101
57	VCC	58	GND
59	VCC	60	GND

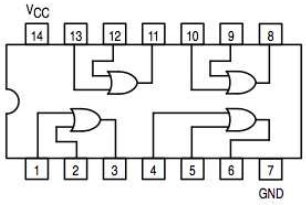
APPENDIX D

DEVICE DATA SHEET

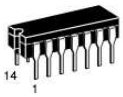
MOTOROLA

QUAD 2-INPUT OR GATE

SN54/74LS32



QUAD 2-INPUT OR GATE
LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08

N SUFFIX
PLASTIC
CASE 646-06

D SUFFIX
SOIC
CASE 751A-02

FAIRCHILD SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS00
Quad 2-Input NAND Gate

DM74LS00 Quad 2-Input NAND Gate

General Description

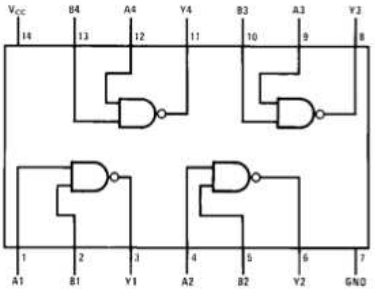
This device contains four independent gates each of which performs the logic NAND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS00M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS00SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS00N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = \overline{AB}$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

FAIRCHILD SEMICONDUCTOR™

May 1986
Revised March 2000

DM74LS02
Quad 2-Input NOR Gate

General Description

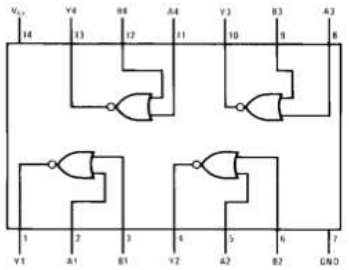
This device contains four independent gates each of which performs the logic NOR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS02M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS02SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS02N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = \overline{A + B}$

Inputs		Output
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

FAIRCHILD SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS08
Quad 2-Input AND Gates

General Description

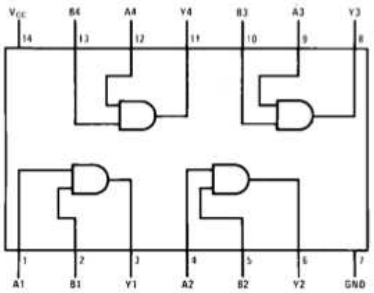
This device contains four independent gates each of which performs the logic AND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS08M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS08SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS08N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

FAIRCHILD SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS86 Quad 2-Input Exclusive-OR Gate

DM74LS86
Quad 2-Input Exclusive-OR Gate

General Description

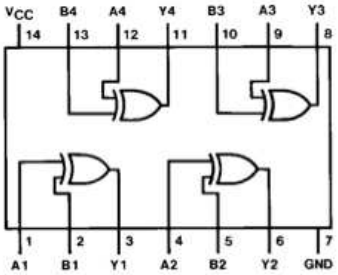
This device contains four independent gates each of which performs the logic exclusive-OR function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS86M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS86SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS86N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = A \oplus B = \overline{A}B + A\overline{B}$

Inputs		Output
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

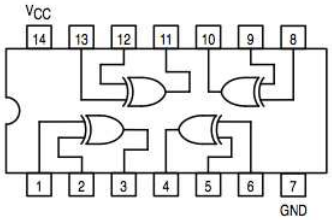
MOTOROLA

QUAD 2-INPUT EXCLUSIVE-OR GATE

SN54/74LS386

QUAD 2-INPUT EXCLUSIVE-OR GATE

LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08

N SUFFIX
PLASTIC
CASE 646-06

APPENDIX E
SOURCE CODE

E-1 Functional Behavior of Device

E-1.1 OR Gate

```
module truthtable(A,B,C);

      input       A,B;
      output      C;
      reg   C;

      always@(A,B)

            if (A==0 && B==0) C<=0;
            else if (A==0 && B==1)  C<=1;
            else if (A==1 && B==0)  C<=1;
            else C<=1;

endmodule
```

E-1.1 NAND Gate

```
module truthtable(A,B,C);

      input       A,B;
      output      C;
      reg   C;

      always@(A,B)

            if (A==0 && B==0) C<=1;
            else if (A==0 && B==1)  C<=1;
            else if (A==1 && B==0)  C<=1;
            else C<=0;

endmodule
```

E-1.1 NOR Gate

```
module truthtable(A,B,C);

      input       A,B;
      output      C;
      reg   C;

      always@(A,B)

            if (A==0 && B==0) C<=1;
            else if (A==0 && B==1)  C<=0;
            else if (A==1 && B==0)  C<=0;
            else C<=0;

endmodule
```

E-1.1 AND Gate

```
module truthtable(A,B,C);

      input       A,B;
      output      C;
      reg   C;

      always@(A,B)

            if (A==0 && B==0) C<=0;
            else if (A==0 && B==1)  C<=0;
            else if (A==1 && B==0)  C<=0;
            else C<=1;

endmodule
```

E-1.1 XOR Gate

```
module truthtable(A,B,C);

      input       A,B;
      output      C;
      reg   C;

      always@(A,B)

            if (A==0 && B==0) C<=0;
            else if (A==0 && B==1)  C<=1;
            else if (A==1 && B==0)  C<=1;
            else C<=0;

endmodule
```

E-2 Compare Block

```
module compare(C,op,pass,fail);

      input       C,op;
      output      pass;
      output      fail;
      reg         pass;
      reg         fail;

      truthtable  truthtable(C);

      always@(op,C)

            if (op==C)
            begin
                  pass<=1; fail<=0;
            end

            else
            begin
                  pass<=0 ; fail<=1;
            end

endmodule
```

E-2 Top Level Module

```
module      test  (A,B,op1,op2,op3,op4,
                              pass1, pass2, pass3, pass4,
                              fail1, fail2, fail3, fail4);

      input       A,B,op1,op2,op3,op4;
      output      pass1,pass2, pass3, pass4,
                  fail1, fail2, fail3, fail4;
      wire        C;

      truthtable        truthtable1 (A,B,C);
      compare           compare1(C,op1,pass1,fail1);
      compare           compare2(C,op2,pass2,fail2);
      compare           compare3(C,op3,pass3,fail3);
      compare           compare4(C,op4,pass4,fail4);

endmodule
```