LOGIC INTEGRATED CIRCUIT(IC)
FUNCTIONAL TESTER

GOH SIANG WEI

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

MAY 2011

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ___________________________

Name : ____GOH SIANG WEI______

ID No. : ______07UEB06544________

Date : _______6 / 5 / 2011_________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"LOGIC INTEGRATED CIRCUIT(IC) FUNCTIONAL TESTER"** was prepared by **GOH SIANG WEI** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _______________________

Supervisor : Mr. NG CHOON BOON

Date : _______________________

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to successful the completion of this project. The special thank goes to my helpful supervisor, Mr. Ng Choon Boon whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the project. The supervision and support that he gave truly help the progression and smoothness of the project. The co-operation is much indeed appreciated. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

My grateful thanks also go to my partner Miss Law Siew Li. A big contribution and co-operation from her during the development is very great indeed. All projects during the program would be nothing without the enthusiasm and imagination from her. The whole project really brought us together to appreciate the true value of friendship and respect of each other. Not forget, great appreciation go to the rest of Lab Assistant that help me from time to time during the project.

I also wanted to thank my family who inspired, encouraged and fully supported me for every trial that come my way .In giving me not just financial, but morally and spiritually. I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

Finally, special thanks to UTAR and Faculty of Engineering and Science for providing the numerous tools, programming systems, microcontroller programmers and facilities at the Lab that have encouraged, support and help me in completing this program successfully.

LOGIC INTEGRATED CIRCUIT FUNCTIONAL TESTER

ABSTRACT

In the present world, everything is going to be more developed, attractive and user friendly. Hence, the objective of this project is to develop a low cost, computer independent and user friendly logic Integrated Circuit (IC) tester. The logic IC tester will be able to test the function of basic 74 series TTL Logic gates (AND, OR, NOR, NAND, XOR, XNOR, Inverter (NOT)) and flip-flop ICs (D flip-flop, JK flip-flop). The logic IC functional tester can be operated in Personal Computer (PC) mode or Portable mode. This tester uses the flexible programmable features of PIC16F877A microcontroller for many applications. Visual Basic was used to develop the user interface to transmit the instruction from computer to the PIC16F877A microcontroller through Universal Serial Bus (USB) interface for PC mode. No matter 14 or 16pins of ICs, the user-friendly Graphical User Interface (GUI) allows users to select the available logic IC model in the Preset list. Moreover, those ICs are not available in the Preset list can be tested in Customized Testing. GUI will display the result when PIC16F877A send signal back to computer and the results can be stored in PC (hard drive). In Portable mode, PIC16F877A acts as a central processing unit which connected with 1 keypad, 16x2 LCD displays, ZIF socket and PIC18F2620. For PIC18F2620 is connected with memory card holder. Users just need to input the logic IC model by using the keypad, the microcontroller will perform the test on the selected logic IC and the results will be displayed on LCD. The result can be saved in memory card. The logic IC functional tester is successfully built and well functional without any errors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Aim

The primary purpose of the logic IC functional tester project is to construct a simple and inexpensive system which is suitable for testing the function of small scale chip. The system incorporates a friendly Graphical User Interface (GUI) which permits users with no programming expertise to generate tests and operate the system quickly and efficiently. The system also can be used in standalone mode without interface with Personal Computer (PC). The system also will provide data storage which can store the result as a reference for users. The tester system can be applied to standard TTL basic gate and common flip-flop Integrated Circuit (IC).

1.2 Background

The dramatic increase in the use of digital integrated circuits (ICs) has created a need for a fast accurate means of testing such ICs. An IC Tester is to be economically implemented for small or medium-scale users of such IC's (for example, in the lab) and provides a quick but thorough check of its functions with minimal operator action. The IC Tester can be used to test different ICs. The purpose of IC tester is to ensure IC components are in good condition for use. For testing an IC, different hardware circuits for different ICs are needed. This is the main trouble and disadvantage. Hence, it is needed to construct an IC tester to surmount this problem.

1.3 Motivation

The IC testers available in the market today are too costly for individuals to own. Therefore we decided to construct an IC tester which is affordable and user-friendly. The motivation is to build an affordable IC tester for testing the function of 74 series TTL Logic Gates and common flip-flop ICs. The test sequence provide in the database facilitates the detection of defective ICs. Furthermore, the IC tester must be easy to operate, compact, lightweight, portable, and low power consumption.

Next, the motivation is to provide an IC tester in portable mode which is easy and convenient to carry around. In addition, we wish to build an IC tester which provides result storage capability to users.

Nowadays, virtually all desktop and laptop PCs on the market provide USB ports so the motivation is to provide an IC tester with USB interface communication to users. The Universal Serial Bus provides a single, standardized, easy-to-use way to connect to a computer.

1.4 Objectives

For this project, the development of a logic IC functional tester will be carried out. The programmable test system will be designed so that end user has total control over testing of ICs. Hence, IC tester must allow users with no programming expertise to generate tests and operate system quickly and efficiently.

This IC tester should be able to test logic ICs in Portable Mode or PC Mode. In the PC Mode, users can select available ICs in Preset list to be tested and the ICs are not available in list can be tested in Customized Testing. When the microcontroller received a command from the Personal Computer through USB communication interface, it will respond by transmitting pulses to the testing circuit to start the test. After completed testing, the microcontroller will send results back to

PC. The results can be stored in hard drive of PC. The test system must incorporate with friendly Graphical User Interface which allows users with or without programming knowledge to operate the tests.

The tester can also function effectively without using the computer. In Portable Mode, users just need to enter IC model by using the keypad and microcontroller will perform testing on the selected IC. Result will display on LCD after test routine is done. The results can be written to memory card through a memory slot.

Those ICs that can be tested on this IC tester are basic gate and common flip-flop. The models of basic gates are 74LS00 (NAND), 74LS02 (NOR), 74LS04 (NOT), 74LS08 (AND), 74LS32 (OR), 74LS86 (XOR) and 74LS266 (XNOR). The models of the flip-flops are 74LS109 (JK) and 74LS74 (D).

1.5 Scope of Work

The project is separated into two parts which are the hardware and the software. Software implementation for this IC tester project is done by the author. The program written would be able to control all the hardware parts. At the same time, hardware development is done by Miss. Law Siew Li.

1.6 Structure of the Report

This report consists altogether 6 chapters and a short summary for each chapter is presented in this section. Chapter 1 briefly describes the aim, background, motivation and objectives to construct the logic IC functional testers. Furthermore, this chapter also briefly explains and discusses how the project is being carried out and to declare the work scope between the group members.

Chapter 2 explains the types of integrated circuit tester in the market, review the history of the IC tester, fundamental of testing, the researches that have been done are basically on the components needed and the suitable programming in the project and also the methods that can be used to develop the system.

In chapter 3, the methods used to develop the project will be discussed based on the research carried out earlier. Moreover, it includes details of how software design and development will be performed. This section will also mention the steps forward of the project and to discuss the flow of the design and the development of the project.

The chapter 4 discusses about the research of software which are used to develop the project. The chosen assembly software and programming software will be discussed detail in this chapter. This section will attempt to describe the microcontroller port assignments and how the IC tester system works on programming part. This makes readers understand better about the programming flow in the project.

The chapter 5 discusses the testing carried out and the results obtained. This chapter will explain the problems faced and what are the steps are taken to solve and mitigate the problem. This includes the explanation why the error happens and what the corrective actions are taken to prevent further failures.

Chapter 6 will round up and conclude the project along with some suggestion that can be developed and added to the project. This section will highlight the problems faced and presents an overview of Gantt charts in this project. The chart is used for tracking the software development schedules for the whole project.

CHAPTER 2

LITERATURE REVIEW

The main objective of the chapter is to cover the concepts and theory that are used in the project. Furthermore, we read up about the various methods that has been tried to solve the problem or different ways of implementing the system, and compare their achievements.

2.1 IC Tester Introduction

Integrated Circuit (IC) is the main component of each and every electronic circuit can be used for wide variety of purposes and functions. However, sometime due to faulty IC so causes the circuit does not work. Indeed it is lot tedious work to debug the circuit and confirm whether the circuiting is creating problem or the IC itself is dead. Hence, in order to solve these sorts of problems, we intend to make a tester project which would confirm whether the IC under consideration is working properly or not. An Integrated Circuit (IC) tester can be used to test standard and configured on an IC. Each IC is tested by applying test patterns to input pins of the chip and then examining the corresponding outputs for correctness.

2.2 IC Tester in Market

This section focuses on the literature review of similar product done by others. As a result of growth in this market, the consolidation of competitors is an ongoing process. Comparing the IC tester in the market is important in order to define what feature and device are needed by users, and the quality of the IC tester can be improved as well. General summarise on different ways has been tried to implement the system, and compare their achievements then construct a unique IC tester for users.

2.2.1 SU-300 IC Tester

Figure 2.1: SU-300

SU-300 (from Leaptronix Company) is a tester which can works in stand-alone or PC-Based mode. The tester is because of its modular design (SU- 300) is able to work with different types of IC through various cartridges or adaptors. Table 2.1 shows the feature for SU-300.

Table 2.1: Feature SU-300

Features:
● Support different IC families and IC packages through various cartridges and adaptors.
● Each module has protection function for sale, stable, and fast programming.
● Graphic design lets programming status clear.
● Support low voltage "green" IC.
● User friendly and process programming by pressing ↑, ↓, Enter and ESC buttons
● Auto detect function: wrong device insertion, bad pin connecting, etc.
● With auto-search FLASH/EPROM brand and serial number function.

2.3 Digital Based IC Tester

This IC tester by Ranjith (2008) can test digital ICs having a maximum of 24 pins. Since it is programmable, any number of ICs can be tested within the constraint of the memory available. This IC tester can be used to test a wide variety of ICs which includes simple logic gates and also sequential and combinational ICs like flip-flops, counters, shift registers and so on. It is portable and easy to use.

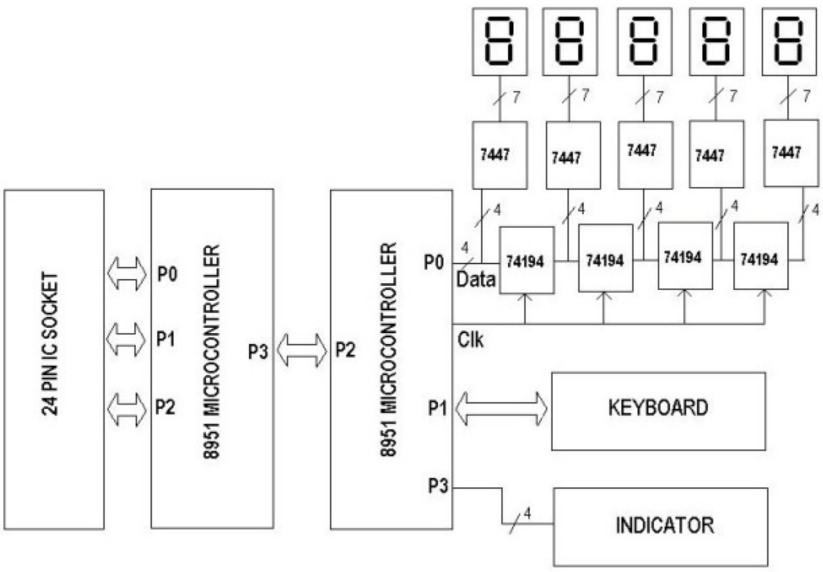

Figure 2.2: Block Diagram of Programmable Digital IC Tester (Ranjith, 2008)

As shown in figure 2.1, it consists of two 8951 microcontroller ICs, a 24-pin IC socket, a keyboard unit, a display unit and indicators. By inserting a IC into the IC socket and enter the IC number using keyboard and then press the “ENTER” key can test a particular digital IC. The IC number is displayed in the 7-segment display unit. Four LEDs are provided as indicators. If the IC is being tested is a logic gate, then each of the 4 indicator LEDs correspond to the 4 gates of the IC. If inserted IC is not a logic gate, all the 4 LEDs work as a single indicator.

2.3.1 Model 570A Analogue IC Tester

Figure 2.3: Linear IC Tester Model 570A

The Model 570A Analogue and Model 575A Digital hand held IC Testers from B&K Precision are compact, hand held, battery powered testers offer advanced functionality and ease of use. The 2-line x 16 character dot matrix LCD shows the result of the test as a PASS or FAIL, together with individual pin diagnostics, test made, and possible equivalents.

The Model 570A Analogue IC Tester’s built-in test library includes common Analogue ICs including op-amps, comparators, voltage regulators, voltage references, analogue switches & multiplexes, opto-isolators & couplers, and audio ICs. Besides that, this Model 575A Digital IC Tester’s built-in test library includes broad range of TTL, CMOS, memory, LSI, interface and other devices of up to 40 pins.

Table 2.2: Features of Linear IC Tester Model 570A

Features
• Auto identification mode • Functional test unit emulates passive circuitry to implement a • comprehensive test in a variety of configurations and gain settings • Displays diagnostic information down to individual component pins • Rugged, hand held, battery operated • Built-in membrane keypad, 2 x 16 dot matrix alphanumeric LCD, and high quality 16 pin ZIF socket

2.3.2 Model 575A Digital IC Tester

Figure 2.4: Digital IC Tester Model 575A

The Model 575A (from B&K Precision Corporation Company) is able to locate temperature related faults by using its unconditional or conditional loop testing modes. Unknown device identification is easily accomplished by selecting SEARCH from the menu, selecting the number of pins on the device and activating Search Mode. The 575A will search its library and identify the device, displaying possible functional equivalents for replacement. As part of the IC test, the specific IC number, the functional description of the device, and the status of faulty pins are scrolled through on the built-in display.

Table 2.3: Features of Digital IC Tester Model 575A

Features
• Comprehensive device library covers most TTL, CMOS, memory and interface devices • 40 pin capability (NAND gates or CPUs) • Identifies unmarked and house-coded devices • Detects intermittent and temperature related faults • Displays diagnostic information for individual pins • Battery operated

2.4 Synthesize

We comprehended the integrated circuits are easily identified and tested with these IC Testers. Two models are available, one for linear and one for digital ICs, both with extensive built-in libraries. Small, handheld design is battery powered for portability in the field or lab. Furthermore, very large scale integration allows for advanced functionality, ease-of-use and reliable testing of hundreds of ICs. This removes the guesswork because it can identify, test and verify unknown ICs, quickly and reliably. There are built-in membrane keypad and 2x16 dot matrix alphanumeric LCD shows the result of the test as a PASS or FAIL.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter explains on how this project will be implemented. It included each process from the beginning until the end of this project. Each process and method of how software design and development will be performed in this project is explained in detail.

3.1 Project Overview

i. First of all, the project Logic IC Functional Tester titled is chosen from the list of topics for the final year project provided by University Tunku Abdul Rahman (UTAR).

ii. After being approved by supervisor, this project is then divided into two parts which is the hardware and software. The researches have been done to get better understanding on the building of the logic IC functional tester and also the methods that can be used to develop the system. The related information from the online journals, online articles, data sheet and reference books are very important in design the logic IC functional tester.

iii. In this report, software section will be discussed in detail which about microcontroller. The research and studies on the project need to be carried out. Hardware and Visual Basic section will be discussed in my partner's report.

iv. An appropriate programming language will be chosen to develop the software needed in constructing a logic IC functional tester. Hence, the program uses the assembly programming language that can be complied, built and also programmed into the PIC16F877A and PIC18F2620microcontroller. The software is integrated into the hardware to make sure the logic IC functional tester is functioning properly. Although the program is already functional, it must still be modified to suit the functionality of system.

v. This project would be designed for improvement, troubleshooting and a set of the steps taken to solve and mitigate the problem in the testing. A lot of necessary or appropriate skills and knowledge will be applied in this project.

vi. Hardware part and software part will be integrated. Testing and troubleshoot will be applied in order to make sure the logic IC functional tester in good performance. Studied and analyzed the possible enhancement to the IC tester.

The Figure 3.1 shows the progress and pace of this project throughout the two semesters worked based on the overview as below.

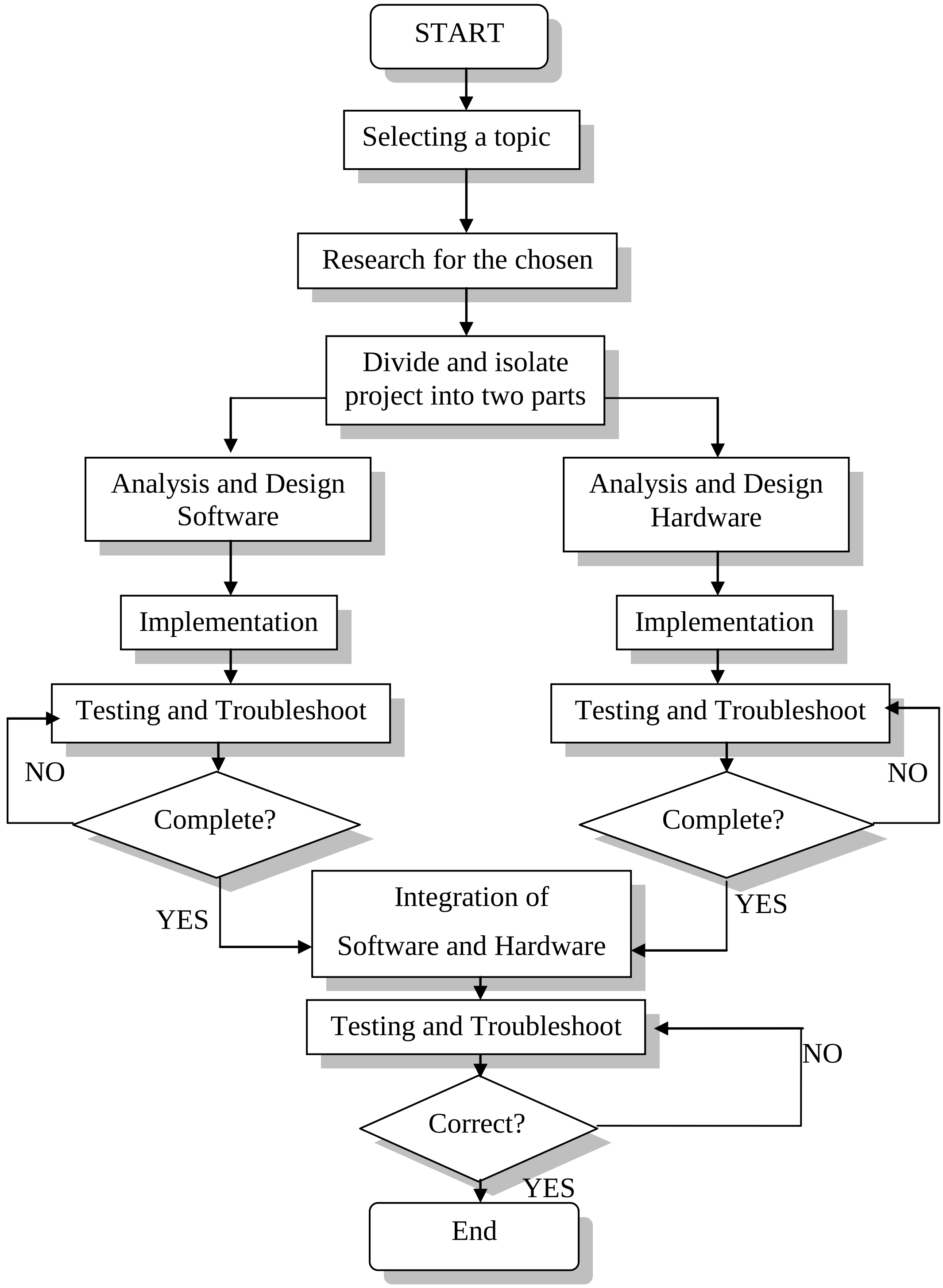

Figure 3.1: Project Overview

3.2 Software Methodology

In order to implement these design aims, it was first necessary to understand the logic IC functional tester. Hence, research and studies on the project need to be carried out. This is a programmable based project, so the tester can be totally control by programming must be designed. The unexpected condition may occur during the process and development so the knowledge about the corresponding solutions was attained through testing the application and trying each method.

Furthermore, the numerous type of programming language will be studied and analyzed afterward choosing an appropriate language to be used in the project and then start to study and perform the language. The information about the USB interface, result storage and other electronic component in the project will be studied. This was extremely important as a good understanding about the components used in order to produce a high-quality Logic IC Functional Tester.

The assembly and C language was studied and understood. The assembly and C language coding will be loaded into microcontroller. These coding are used to control the algorithm of sending the test vector into Device Under Test (DUT) and obtain the desired response generated by the DUT. At the same time, Microsoft Visual Basic has been chosen as the software to build the Graphical User Interface (GUI) was also studied. The GUI is used to communicate the PC with the microcontroller.

The flow chart of the testing algorithm was then generated. All information about the software development methodology is available at this flow chart. The flow chart served as the basis for the source code written for the microcontroller. After that, the source code will be programmed into the microcontroller. Testing and calibration will be done to make the IC tester perform optimally and correctly. Mistakes may be found then will be troubleshot and be corrected.

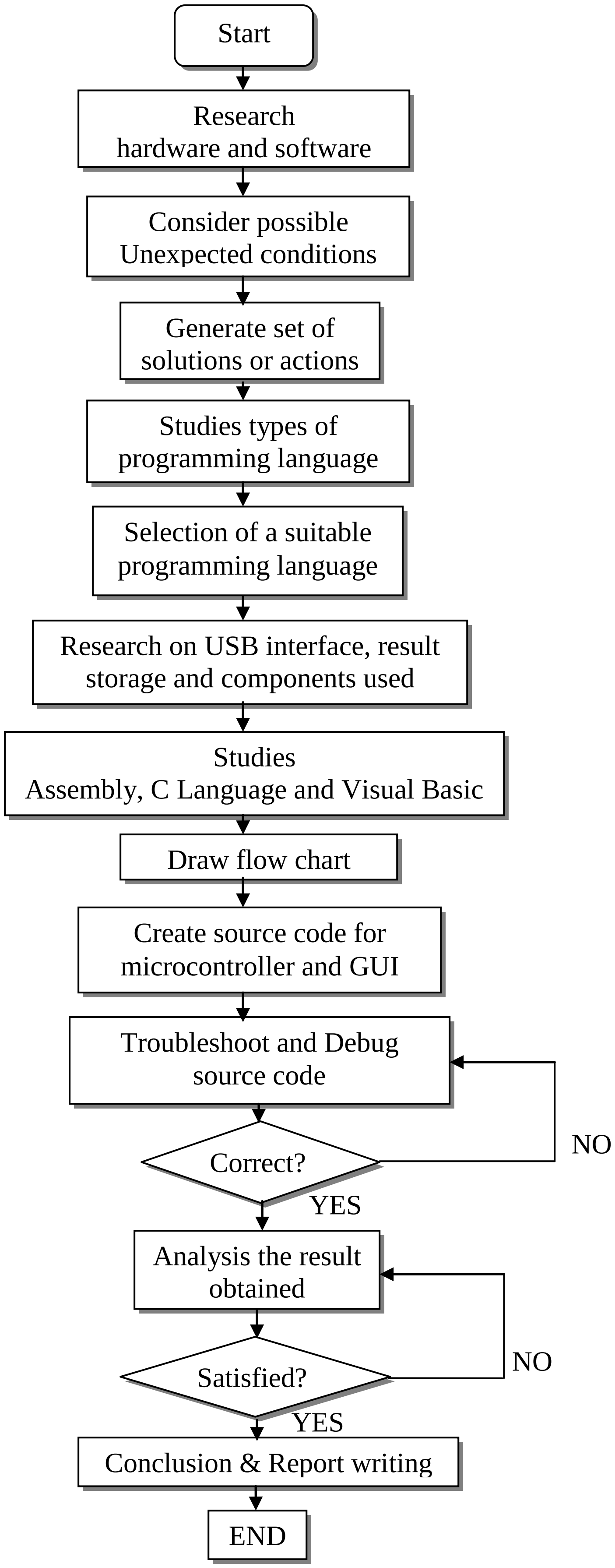

Figure 3.2: Software Implementation Methodology

3.3 Project Planning

For the period of 2 semesters which is 28 weeks, many tasks will be perform to develop the system while trying to follow the time scheduled. Based on the project planning, the progress of the project can be considered satisfactory as most of the work planned have been completed.

Table 3.1 shows the Gantt charts in this project. The chart is used for tracking the software development schedules. It is useful to show additional information about the various tasks or phases of the project. Hence, the tasks and project is completed successfully at the end.

Table 3.1: Gantt-Chart Checklist

First Semester	**June**	**July**	**August**	**September**	**Onward till end of 2nd semester**
Choose project title					
Research on title					
Selection of Software					
Study Assembly programming language					
Study C language					
Write Progressive Report					
Prepare Presentation Slide					
Second Semester	January	February	March	April	May
Writing programming					
Testing and Troubleshoot					
Combine with hardware and debugging					
Final Testing					
Writing Final Report					
Prepare Presentation Slide					

Legends:

Scheduled completion period | Actual completion period | Progress ongoing

CHAPTER 4

HARDWARE IMPLEMENTATION

4.1 Hardware Implementation

Figure 4.1 shows overall circuit connection for logic IC functional tester. The schematics of the circuit are shown in Figure 4.2 and Figure 4.3.

9V battery is the power source for this logic IC functional tester. Voltage regulators were used to reduce voltage level become 5V and 3.3V. 3.3V is required for power source of memory card while others components' power sources were connected to 5V. Main power switch used to control the power of the circuit either is turn on or turn off.

PIC16F877A microcontroller acts as the brain of the circuit. It controlled LCD, keypad, serial to USB, ZIF socket and PIC18F2620 by sent and received command. The function of PIC18F2620 microcontroller is the memory card control module. Reset button is needed for reset two microcontrollers when either microcontroller is malfunction. ZIF socket is used for users place their model ICs that wish to test in the circuit.

MAX232 used to convert signal from or to PIC to or from TTL level for serial data transmission. Serial to USB converter is used to connect circuit and Personal Computer (PC) so data can be transmitted and received from PC to circuit.

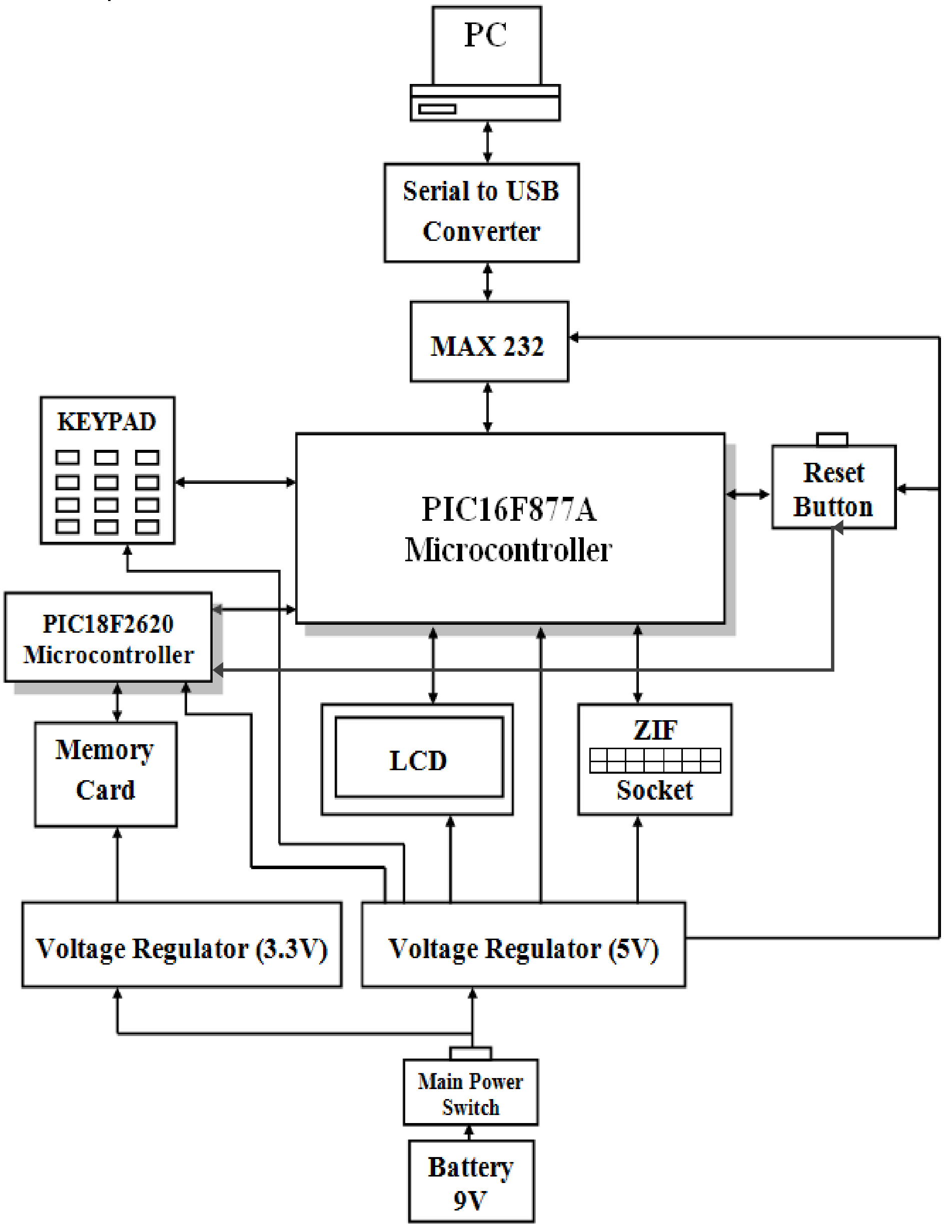

Figure 4.1: Overall Logic IC Tester Circuit Block Diagram

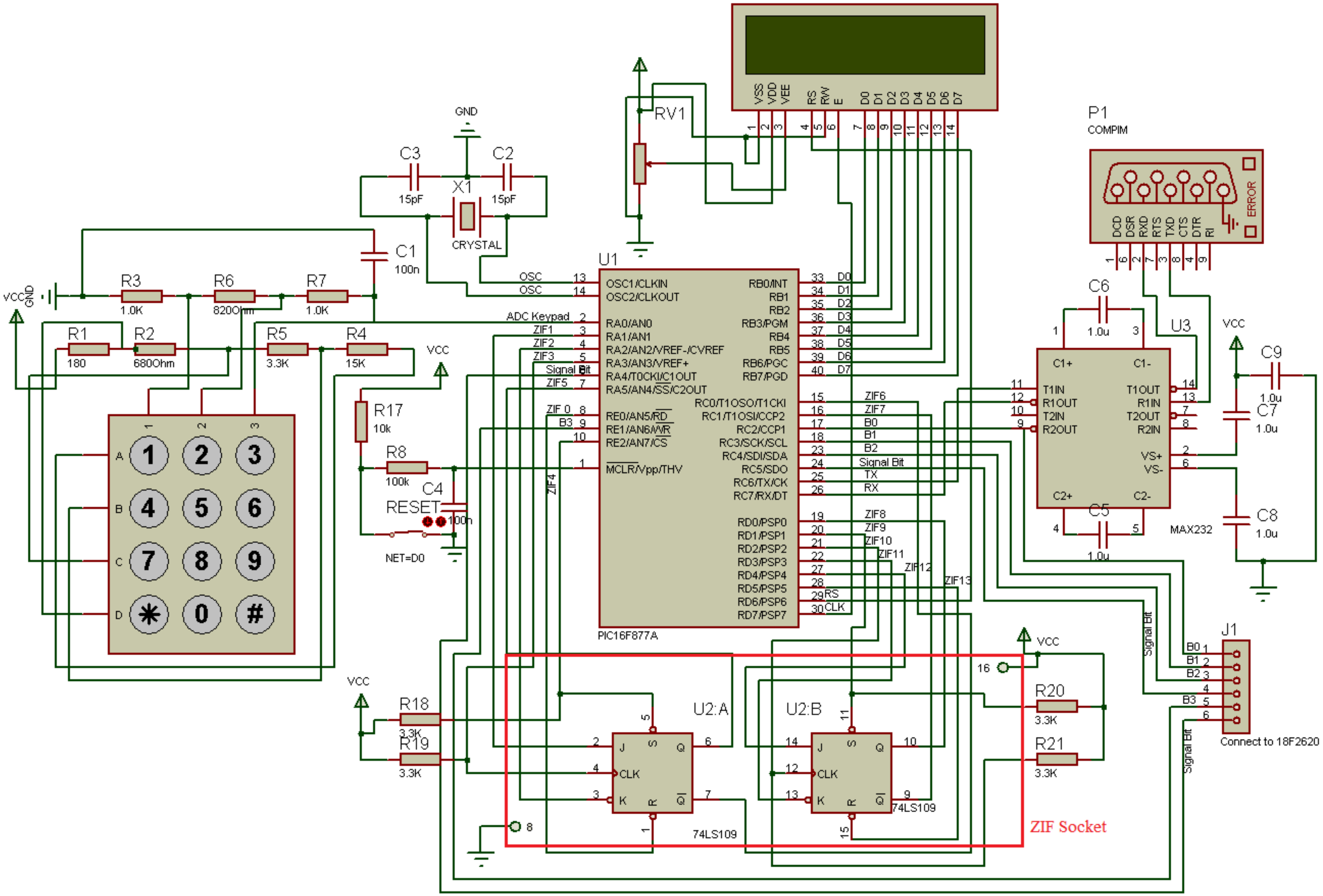

Figure 4.2: Schematic Diagram of Logic IC Functional Tester (1)

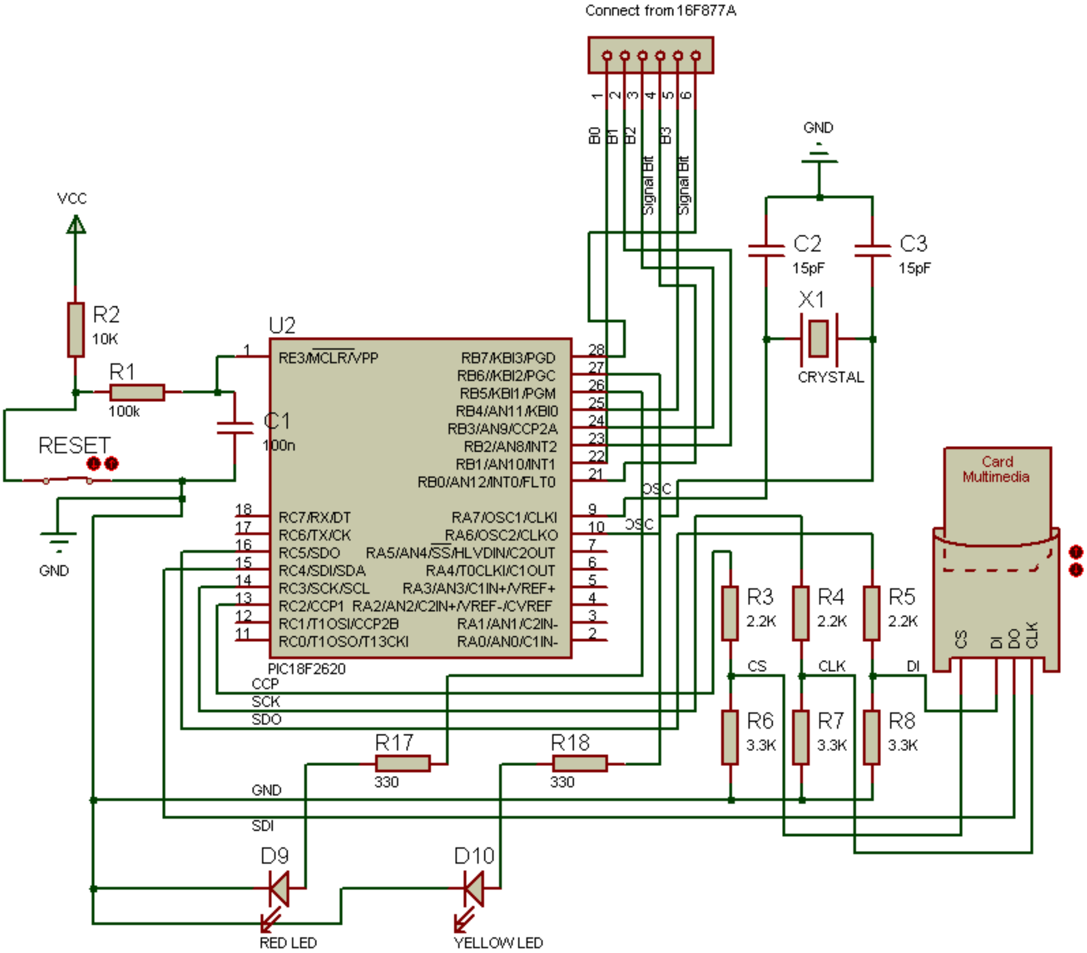

Figure 4.3: Schematic Diagram of Logic IC Functional Tester (2)

4.2 PIC16F877A Microcontroller

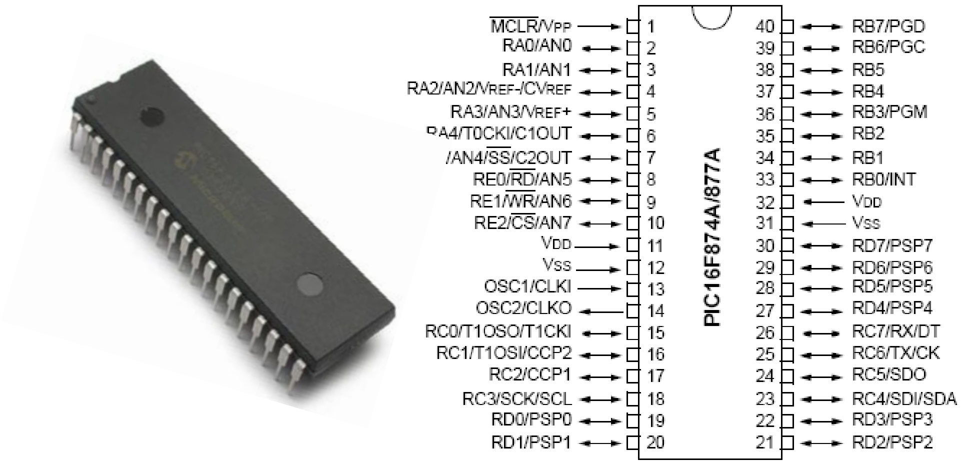

Figure 4.4: Microcontroller PIC16F877A

Microcontroller can perform numerous kind functionalities and can be used to control many types of application. The PIC16F877A microcontroller is chosen to control the overall function of the Logic IC Functional tester project. The PIC16F877A is a 40-pin FLASH-based 8-bit microcontroller and its versatility enables it to be program by not only one type of programming language and it is ideal for the IC tester project work because of its low power consumption, miniature size and also affordable price.

The microcontroller's operating frequency is about 20MHz with 368bytes of internal memory and 256bytes of external memory. From the specification sheets, it has 5 classes of input/output ports, 3 reserved pins for the time, self-programming capability, 2 comparators, and 8 channels of 10-bit Analogue to Digital converter (A/D) and has 2 capture/compare/PWM functions. The microcontroller can be programmed by many types of programming language including Assembly language and also C-programming.

The PIC16F877A has 33 I/O pins that divided into 5 ports which are Port A, Port B, Port C, Port D and Port E. The I/O port can be used as input or output by program their corresponding TRIS register. As example, setting TRIS bit to 1 will make the corresponding port to an input while clearing TRIS bit to 0 will make the corresponding port to an output.

The PIC16F877A contains Universal Synchronous receiver Transmitter (USART) module. The microcontroller not only for the test IC purposes only but it also acts as the transfer medium between the Personal Computer (PC) with the external environments and devices. Hence, the USART is configured as a full-duplex asynchronous system that communicates with USB port of the personal computer by the USB to Serial port converter cable.

4.3 18F2620 Microcontroller

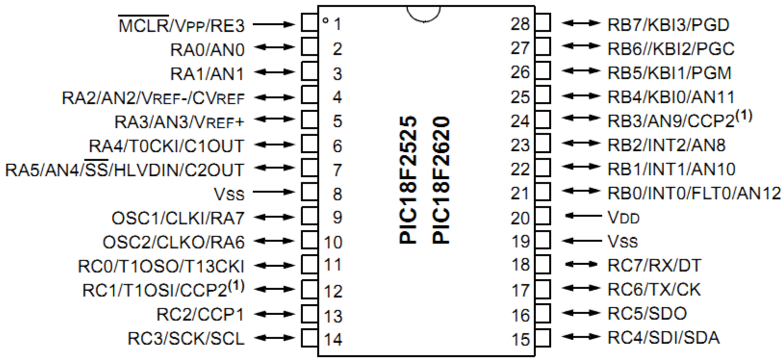

Figure 4.5 : Microcontroller 18F2620

The reason that we used this microcontroller because the 16F877A not afford to transfer data to memory card which the memory card needed data bus up to 512 bytes but 16F877A just has 368 bytes per data bus. 18F2620 has enough data bus which up to 3986 bytes. We chose 28 pins PIC instead of 40 pins because the PIC just needed to interface with 4 pins of Memory card. Hence this can reduce the unused pins. Figure 4.5 shows the pin diagram of 18F2620.

4.4 Ports Assignment

Table 4.1 shows the port assignment for PIC16F877A microcontroller while Table 4.2 shows the port assignment for PIC18F2620 microcontroller. Different pin of microcontroller has different of function. Table 4.1 and Table 4.2 are used to explain the function and pin name of each pin for both microcontroller.

Table 4.1: 16F877A Microcontroller Port Assignment

Device	Pin NO.	Pin Name	Function
Reset button	Pin 1	MCLR/Vpp	1= Programming Voltage, 0=Master Clear
ZIF Socket	Pin 3 – Pin5 , Pin7	RA1 – RA3, RA5	Input / Output port for ICs
	Pin 8, Pin 10, Pin 15-16	RE0, RE2, RC0-RC1	
	Pin 19-22, 27, 28	RD0-RD3, RD4, RD5	
Keypad Input	Pin 2	AN0	Analogue input into ADC
18F2620	Pin 6	RA4	As Output waiting data from 18F2620, RB7
	Pin 9	RE1	As data bit
	Pin 17, Pin 18, Pin 23, Pin 24	RC2, RC3, RC4, RC5	RC5 as switch for 18F2620, RB0, RC2-RC4 as data bits
MAX232	Pin 25	Tx	USART asynchronous transmit
	Pin 26	Rx	USART asynchronous receive
LCD	Pin 29, 30	RD6, RD7	RD6 as RS signal, RD7 as E pulse
	Pin 33-40	RB0-RB7	Send 8 bits data to LCD
Crystal Oscillator	Pin 13 – Pin 14	OSC1, OSC2	As clock input for PIC

Table 4.2: 18F2620 Microcontroller Port Assignment

Device	Pin NO.	Pin Name	Function
Reset button	Pin 1	MCLR/Vpp	1= Programming Voltage, 0=Master Clear
16F877A	Pin 21	RB0	As Input waiting signal bit from 16F877A, RC5
	Pin 22 – Pin 25	RB1-4	As data bit
	Pin 28	RB7	As Output sending signal bit to 16F877A, RA4
Memory Card	Pin 13	RC2	Send data bit to Chip Select (CS) pin on Memory Card, Pin 1
	Pin 14	SCL	Send clock to Serial Clock (SCK) pin on Memory Card, Pin 5
	Pin 15	SDI	Receive data from Memory Card, Pin 7 (Data Out, DO)
	Pin 16	SDO	Send data to Memory Card, Pin 2 (Data In, DI)
LED indicator	Pin 26	RB5	Connected red LED
	Pin 27	RB6	Connected yellow LED
Crystal Oscillator	Pin 9 – Pin 10	OSC1, OSC2	As clock input for PIC

CHAPTER 5

SOFTWARE IMPLEMENTATION AND DEVELOPMENT

5.1 Software Overview

5.1.1 MPLAB IDE

MPLAB Integrated Development Environment (IDE) was used to write the assembly language for 16F877A. MPLAB is a free software provided by Microchip Technologies Ins. It runs as a 32-bit application on Microsoft Windows and provides a host of free software components for application development and debugging. It can convert the source code into a HEX file, which HEX file is needed before it can be program into the microcontroller. MPLAB IDE also provided a powerful full-featured text editor which allows users to create and modify the source code easily. Source code can be debugged easily with the aid of a build results window that displays the bug detected by the compiler when generating the executable files.

5.1.2 mikroC Compiler for PIC

MikroC compiler was used to write C language into microcontroller 18F2620 which is interface with the memory card. The reason that we chose this software for writing C programming is the mikroC is an extremely simple compiler which comes along with a lot of useful and straight forward libraries includes memory card. This will made our implementation easier especially for a beginner.

5.1.3 Proteus VSM

Proteus Virtual System Modeling (VSM) is a software that combines mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs.

When the coding of microcontroller done writing in MPLAB, HEX file of the source code which is compiled by the MPLAB is load into the Proteus VSM circuit for simulation. Through the simulation, we can straight debug the error in the source code and correct the error immediately. Without this software, we need to burn the HEX file into microcontroller and test or debug error in real circuit. Hence, with Proteus VSM will save a lot time when debugging the error.

5.1.4 PICkit 2 Development Programmer/Debugger

PICkit2 is a free software which used to program the HEX file which is generated by MPLAB or mikroC into microcontroller. The software is developed by Microchip Company. It can program a large range of PIC chips from the PIC10, PIC12, PIC18 and PIC24 families. It takes advantage of Microchip's Full-Speed USB device, significantly speeding up the development programming.

5.2 Software Development

5.2.1 Microcontroller

In this project, MPLAB IDE has been chosen to program the microcontroller 16F877A in assembly language and generate the HEX files of the source code during the compilation which is a stage of converting the assembly language into machine language.

Another microcontroller which is 18F2620 is programmed by mikroC compiler with C programming language. The mikroC is used because the libraries are easier to use compare with MPLAB IDE especially for memory card interface with 18F microcontroller. It just required a few command to write a text file from 18F microcontroller into memory card. It makes us save a lot of time not only on the develop progress and also shorten the learning time. Not only that, mikroC software comes along with few examples which are quite helpful during learning progress.

In the nutshell, 16F877A microcontroller acted as brain of the whole circuit system while the 18F2620 microcontroller controls the memory card module of the circuit.

5.2.2 Serial Port Communication Interface Development

The main interface between computer and microcontroller is through a USB to RS232 converter. It will be utilized as the interface for the tester circuitry connects with PC. Hence Universal Synchronous Asynchronous Receiver Transmitter (USART) has to set in the microcontroller 16F877A to transfer data between the PC and microcontroller.

USART is a Serial Communication Interface (SCI). The USART can be in synchronous (half duplex) or asynchronous (full duplex). Full duplex asynchronous operation is both transmission and reception can occur at the same time. The transmission and reception can be independently enabled. Hence, we utilize full duplex asynchronous operation in the project.

The most important configuration concerns the USART parameters which to configure the baud rate of the microcontroller and the USART peripherals to make it ready to communicate with the serial port. These setting have to be set on the microcontroller side and PC side separately to make sure the both device had connected and ready to communicate. The setting used is listed as below:

Table 5.1: Settings of Serial Port

SETTINGS	
Baud Rate	**57600bps**
Data Bits	**8**
Parity	**-**
Stop Bit	**1**

Next will mention about how to set up the USB to RS232 cable interface the PC with the circuit correctly. Before using the USB to RS232 converter, “PL-2303_Driver_Installer” need installed to PC. Plug in the USB port into the PC, the device will show in the device manager as Figure 5.1. Right click the Prolific USB-to- Serial Comm Port and go to properties and setup the detail as Figure 5.2. Lastly click the “Advanced” button to setup the COM port number to COM4. The device now can interface circuit and PC perfectly.

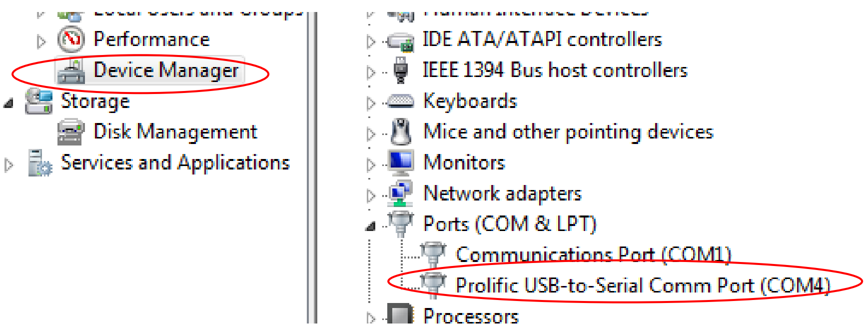

Figure 5.1: Device shows on Device Manager

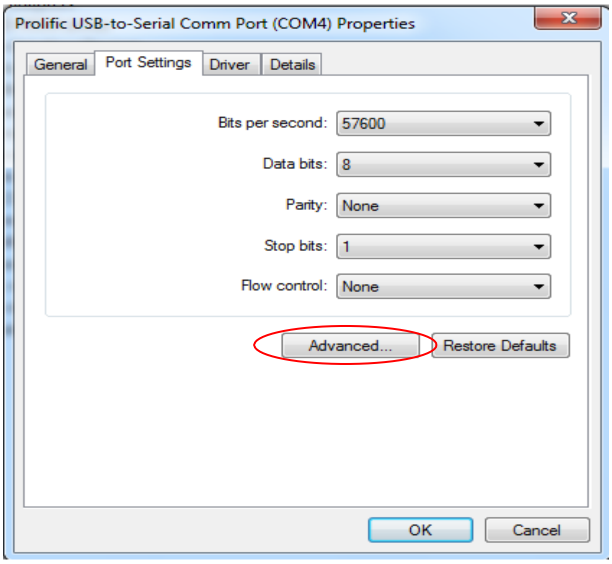

Figure 5.2: Properties of Profilic USB-to-Serial Comm Port

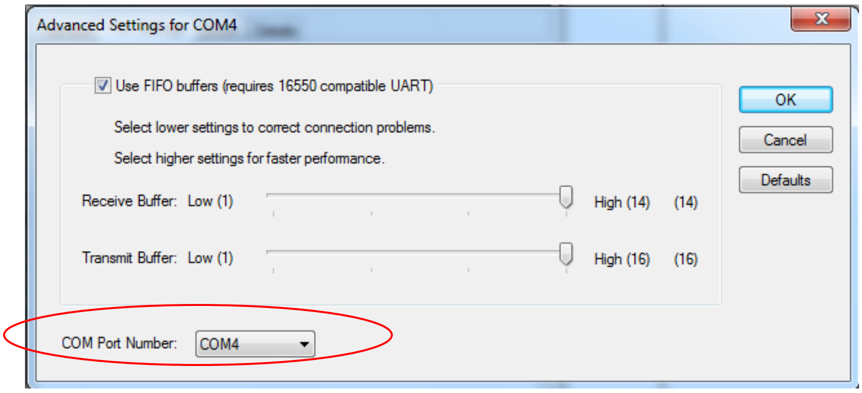

Figure 5.3: Advanced Settings of Profilic USB-to-Serial Comm Port

5.2.3 Keypad Development

As shows in Figure 5.4, the keypad of Logic IC Functional Tester is connected to an ADC input pin (RA0) of microcontroller with a resistor matrix. The concept (by Schimidt; 2009) extremely feasible and possible better than the conventional keypad because it has advantage of reducing the number of pins used which required one Input/ Output (I/O) pin of microcontroller. The resistor matrix control the key input according to voltage divided from the voltage dividing resistors. Hence, different keypad button is pressed will produce different voltage values input to ADC then causes different hexadecimal codes will be produced. Different ranges of the hexadecimal codes value will represent different buttons which determined by us as shown in Table 5.2. When a button of keypad is pressed, the ADC will produce a Hexadecimal value and compare with the hexadecimal value range to determined which button is pressed. The number buttons of keypad are used when the tester prompt message to request users to enter the logic ICs model number or the number of testing that users need to test. The ‘*’ button acted as clear button and the ‘#’ button acted as enter button. The program flow of the keypad is shown in Figure 5.5.

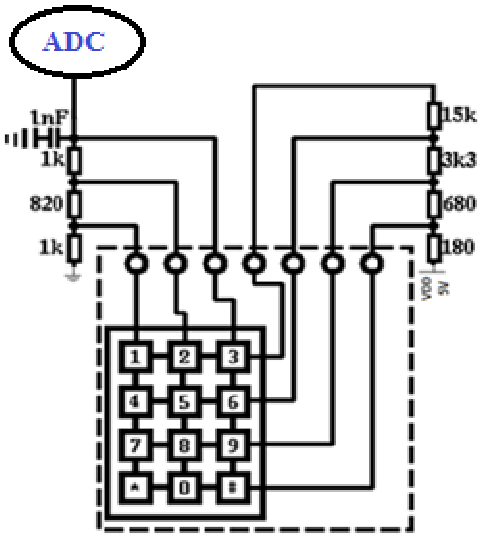

Figure 5.4: Resistor Matrix Keypad (Schimidt;2009)

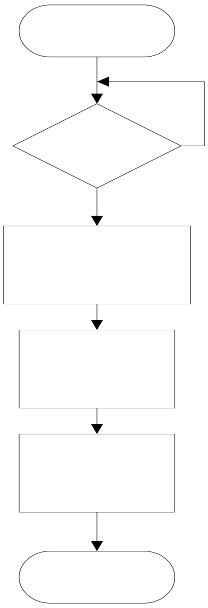

Figure 5.5: Flow Chart of Keypad

Table 5.2: ADC Value of Each Button

Keypad Button	8-bit-AD value	
	Min	Max
No button to be pressed	0	0
1	A	F
2	11	17
3	1C	22
4	2B	33
5	47	4F
6	5F	6F
7	7F	8F
8	A5	B5
9	BD	C7
*	CE	D9
0	DD	E8
#	E9	F5

5.2.4 SD Card Development

Memory card that we chose is Secure Digital (SD) card. SD card give the IC tester ability to store results and it is easy and affordable. SD Card is a non-volatile memory card format. At the physical level, SD supports at least three transfer modes:

1-bit SD mode: Separate command and data channels and a proprietary transfer format.

4-bit SD mode: Uses extra pins plus some reassigned pins.

SPI mode: Serial Peripheral Interface Bus, a simpler subset of the SD protocol for use with microcontrollers.

SD card is chosen for the results storage which interfaced with 18F2620 microcontroller via SPI mode. Hence, the SPI mode must be configuring for the microcontroller to interface with SD card. Table 5.4 shows the function of each pin for SD card use in SPI mode.

The SCK pin of PIC is provided with some clocks by sending some data through the SPI. The SD card enable by lowering its CS pin. Another primitive function is enabling the CS to the SD card. Since the SD card has a negative CS (the signal is low to select the card) means when 0 is selecting the card, and when 1 is not selecting the card. Hence, the SPI mode 0 is used to select the SPI of SD card. As described in the SPI specification, the PIC starts every bus transaction by asserting the CS signal low. All the commands and transmit data are sent to the media card on the Data In (DI) line of SD card. All the command response and receiving data are received from the media card on the Data Out (DO) line of SD card. CLK is used to synchronize the data transfer on the bus. (Ibrahim, 2008)

The communication between the PIC and the SD card can be simply divided into two phases: card initialization phase and data access phase. In initialization phase, the PIC uses a series of commands to reset the card, set the card to a known operational mode, set the block length for data access. In data access phase, PIC uses a series of commands to read and/or write data.

Table 5.3: Function of SD Card's Pins

SD card Pins	Function
DI	Data Out from SD card and receive by microcontroller
DO	Data In to SD card and sent by microcontroller
SCK	The clock is used to drive data out and receive data on the SD card.
CS	The chip select signal enables the card during data and command transfer.

Table 5.4: Pin Assignments of SD Card

Pin	Pin Name	Function (SD Mode)	Function (SPI Mode)
1	DAT3/CS	Data Line 3	Chip Select/Slave Select [CS]
2	CMD/DI	Command Line	Master Out/Slave In [DI]
3	VSS1	Ground	Ground
4	Vdd	Voltage Supply	Voltage Supply
5	Clock	Clock	Clock [SCK]
6	Vss2	Ground	Ground
7	DAT0/D0	Data Line 0	Master In Slave Out [DO]
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

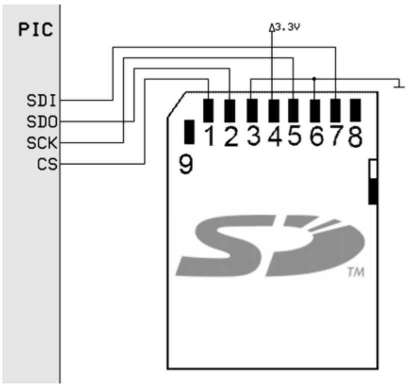

Figure 5.6: Pins connection of SD card to microcontroller

5.2.5 LCD Development

LCD that used in this project is 2×16 line LCD display which has 14 pins to interface with microcontroller. LCD in this project is used to display command, display number that users entered and display result.

Pin 1 and pin 5 for LCD is connected to ground. Pin5 (RW) is connected to ground because in this project we just write the LCD display but did not read the data from LCD. Pin 2 is connected to 5V and the pin 3 is connected to a potentiometer for control the contrast of LCD. Pin4 (RS) is connected to bit 6 of PORTD while pin 6 (E) is connected to bit 7 of PORTD. The 8-bit of data bus are connected to PORTB.

When Register Select (RS, pin4) is 0 mean that the instruction command code register is selected which microcontroller can send a command such as clear display to LCD. If RS is 1 mean that data register is selected which microcontroller can send data to display on the LCD. When the data is supplied to data pins, a high-to-low pulse needed to supply to Enable (E, pin6) pin in order for the LCD to latch in the data present at the data pins.

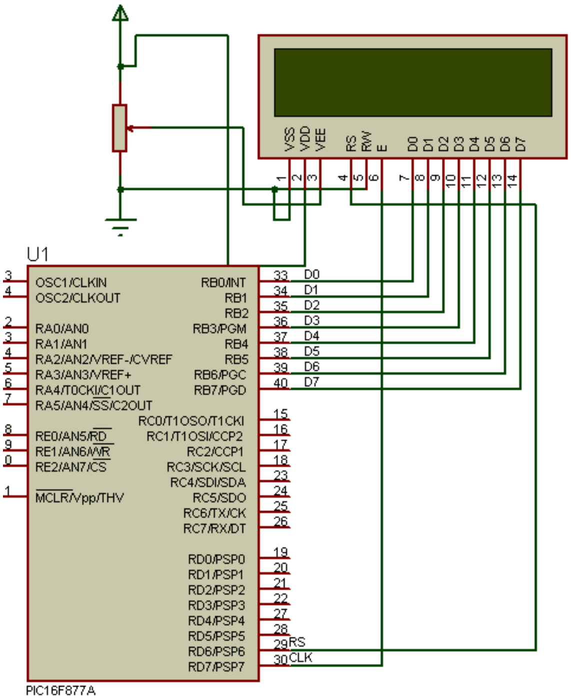

Figure 5.7: Schematic Circuit of LCD interface with PIC16F877A

5.3 IC Tester System

The IC tester system can operate in PC mode or Portable mode. Hence, the system will detect the signal either from the remote host PC (PC mode) or the Keypad (Portable mode). Figure 5.8 shows the general flowing of the IC tester system with these two modes.

When the tester turns on, users can choose to use either PC mode or Portable mode. For the PC mode, users need connect the circuit with USB port at the PC via the USB to Serial converter cable. Through the Visual Basic, users can choose either Preset testing or Customized testing. Preset mode has 9 models ICs available for users to test the Logic ICs whether are well function or not. The result will be shown on the VB and can be saved in the PC. The Customized mode allows users to test Logic ICs which not in the list of 9 model ICs by selecting the input, output, Clock and Vcc of the IC that wanted to test on the Visual Basic. The result also can be saved in the PC. For the save setting, when the saved setting is select, the result will be automatically saved into the directory that users desire it to be saved in.

Generally in PC Mode, the microcontroller is ready to receive data signal from the remote host PC. When the microcontroller detected the data signal of PC, it will test whether the transmitted character matched the ASCHII character that received. If the code does not match any microcontroller characters, the microcontroller will ignore the signal and continue to wait for the signal send from the PC. If the signal matches the microcontroller characters, it will go to the corresponding subroutine to perform the operation requested by the users. After the test, microcontroller will send the result back to the remote host PC.

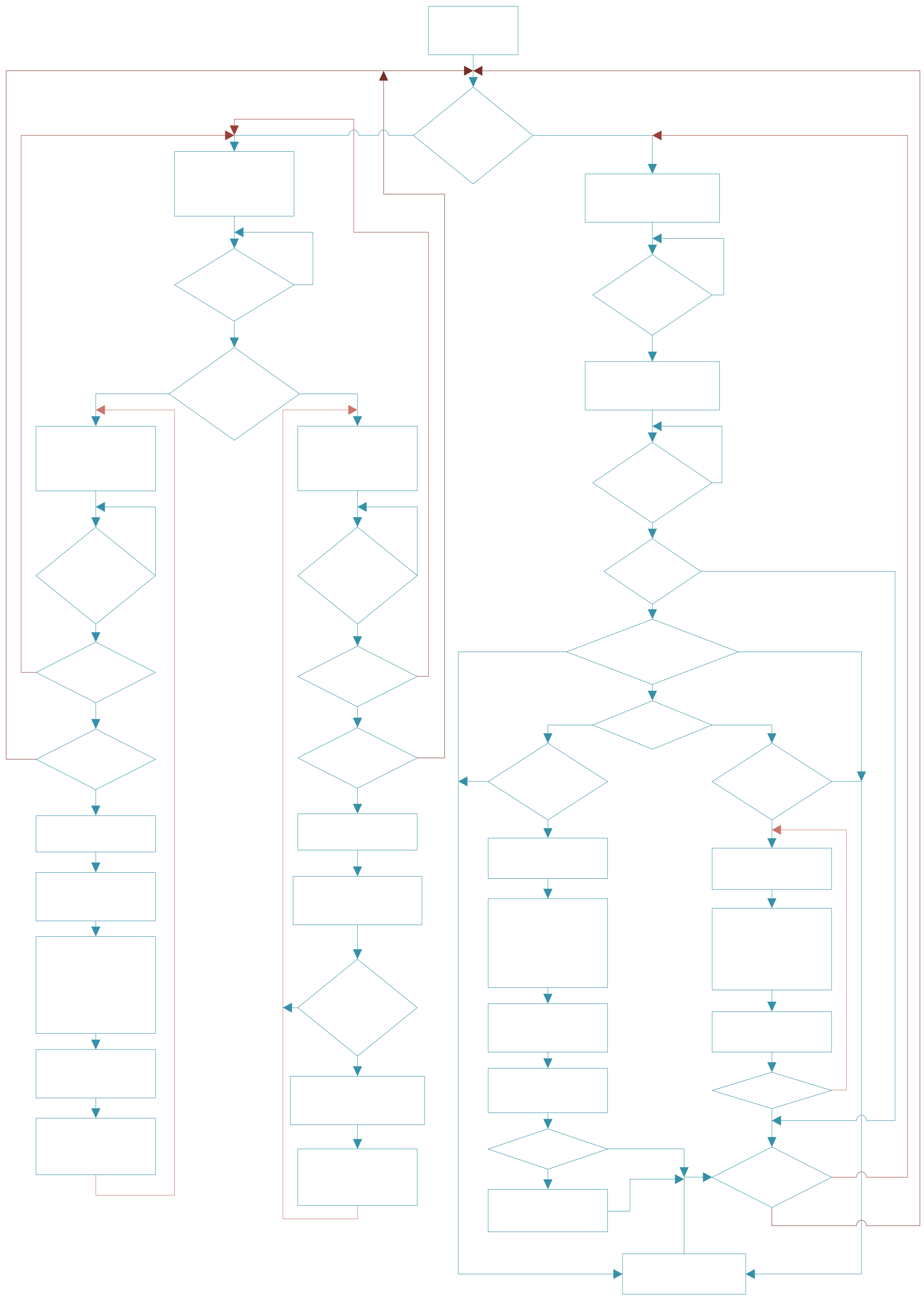

Figure 5.8: Overall Program Flow

In Portable Mode, users need to key in the IC model they wish to test and the number of test that they desire to test on the Logic ICs. The microcontroller will started to check whether the IC model users key in is available. If the model is not available, the program will send a signal to the LCD display to inform users the model is not available. If the model is available, the program will move into the corresponding IC testing subroutine that requested by users.

If the number of loop is 1 time, the system will start to test the each gate in the IC. The program will retrieve the value obtained from the gate's output and compared it with the expected value. After the comparison, the program will send the results to the LCD. The LCD will display IC result as either "PASS" or "FAIL". If the result is fail, the program will send a signal to the LCD in order to show which IC gate is not function properly. The system will then ask the users want to save the result into memory card or not. If users selected to save the result, a text file with the name of ICTESTER.txt will be created and result will be saved into it. Lastly the tester will ask the users want to continue testing or quit the IC functional tester.

If users enter the number of testing more than 1 time, the tester will looped the testing subroutine, compare result and save the result into memory card based on the number that users had entered. The flow of the Portable Mode system is illustrated Figure 5.8.

.

5.4 Codes Description

This section will explain about the meaning of code that implemented on each function for the 2 microcontroller in this project. The subroutines, function of each part, and the special codes used will be described clearly as well. The Visual Basic code will be explained by my partner, Miss Law on her report.

There are 2 programming language that been used for this project which are Assembly Language and C Language. Assembly Language used in MPLAB IDE software to write source code and programmed HEX file into microcontroller 16F877A using PICKIT2. C Language is used in the mikroC software which written the code and generated the HEX file to program into microcontroller 18F2620 using PICKIT2.

5.4.1 16F877A Source Code

5.4.1.1 Macro Function

```
BANK0       MACRO
BCF         STATUS,RP0
BCF         STATUS,RP1
ENDM

BANK1       MACRO
BSF         STATUS,RP0
BCF         STATUS,RP1
ENDM

E_PULSE     MACRO
BSF         PORTD,7
PAGESEL     DELAY5
CALL        DELAY5
BCF         PORTD,7
ENDM

TOLCD       MACRO
BSF         PORTD,6
MOVWF       PORTB
ENDM

CLKFF1      MACRO
PAGESEL     DELAY150
CALL        DELAY150
CLRF        PCLATH
BSF         PORTA,3
PAGESEL     DELAY150
CALL        DELAY150
CLRF        PCLATH
BCF         PORTA,3
ENDM
```

Figure 5.9: Codes of Macro Function

Figure 5.9 shows the Macro functions that will be called in the main code. Macro functions were used due to save time when written code and save space of editor. If the Macro function did not be used, there will be always repeating the same codes when needs it. For example, Marco of CLKFF1, there were 8 lines in the function. With the Marco function, the code will be simplified into 1 line instead of 8 lines.

5.4.1.2 Port Setting

```
BANK1
MOVLW       b'00001110';right justified, set only RA0 is anologue
MOVWF       ADCON1
MOVLW       b'00010001';set RA0,RA4 as input
MOVWF       TRISA
CLRF        TRISB
MOVLW       b'11000000';Set RC6, RC7 as input
MOVWF       TRISC
CLRF        TRISD
CLRF        TRISE
```

Figure 5.10: Port Setting of PIC16F877A

Figure 5.10 shows port setting for the 16F877A microcontroller. ADCON1 have to configure before TRISA had been setting up. Last 4 bits of ADCON1 determine the analogue or digital port configuration for PORTA. As Figure 5.10 shown, binary ‘1110’ had move to ADCON1 which mean RA0, pin 2 of 16F877A had set as analogue input output pin while others as digital input output pins. Setting up bit0 and bit4 of TRISA to ‘1’ will change the RA0 and RA4 to input and set the bit to ‘0’ as output. RA0 is analogue input which connected to keypad which required changing the analogue input to digital input through ADC of AN0. The digital value will be used to determine which button of keypad is pressed by users.

As RC6 and RC7 were connected to MAX232 for serial-to-USB converter, bit 6 and bit7 of TRISC had to set as input. Hence, the microcontroller will ready to receive or transmit data from or to Personal Computer (PC).

5.4.1.3 USART Setting

```
MOVLW       b'00010100'
MOVWF       SPBRG       ;57600 baud fpr 20MHz
MOVLW       b'00100100'
MOVWF       TXSTA       ;enable async, brgh high
BANK0

MOVLW       b'10010000'
MOVWF       RCSTA       ;enable async reception

MOVF        RCREG,W ;flush receive buffer
MOVF        RCREG,W
MOVF        RCREG,W
```

Figure 5.11: Code of USART Setting

Figure 5.11 shows the setting for the USART in the 16F877A. The setting of SPBRG depends on the crystal frequency. In this project crystal that we used is 20MHz, if we change the crystal to 4MHz, the SPBRG has to reconfigure. The baud rate of microcontroller is set to 57600bps. The high speed asynchronous mode is enabled by set the bit 2 of TXSTA register. The serial port and continuous receiving is enabled to constantly detect the incoming data bit set the bit4 and bit7 of RCSTA register.

5.4.1.4 Initialize LCD Setting

```
LCD_INI
            CALL        DELAY20
            BANK0
            BCF         PORTD,6
            MOVLW       B'00000001'
            MOVWF       PORTB
            E_PULSE
            MOVLW       B'00000010'
            MOVWF       PORTB
            E_PULSE
            MOVLW       B'00000110'
            MOVWF       PORTB
            E_PULSE
            MOVLW       B'00001101'
            MOVWF       PORTB
            E_PULSE
            MOVLW       B'00111000'
            MOVWF       PORTB
            E_PULSE
            RETURN
```

Figure 5.12: Code of LCD Initialization

The coding is used to initialize the LCD as shown in Figure 5.12. The coding is to set the LCD display on, on the cursor, cursor blink and put the cursor at the beginning of the first line of LCD. Hence, the LCD will start to display the character at the left hand side. The initialize LCD setting function have to run before displaying any data on the LCD.

5.4.1.5 ADC Keypad Subroutine

```
;********************** ADC Keypad *****************
ADC
            BANK0
            MOVLW       b'10000001';ADC turn ON
            MOVWF       ADCON0
            BSF         ADCON0,0
            CLRF        COUNTER
            CLRF        T
            CLRF        T1
            CLRF        NADC
            CLRF        W
            CLRF        RESULT
            MOVLW       0x30;Make 30 loop delay
            MOVWF       COUNTER
loop
            DECFSZ      COUNTER,F
            GOTO        loop
            PAGESEL     DELAY3
            CALL        DELAY3; SENSITIVE BUTTON
            CLRF        PCLATH
            BSF         ADCON0,GO

test
            BTFSC       ADCON0,GO
            GOTO        test
            BANK0
            MOVF        ADRESH,W
            MOVWF       NADC;the ADC convertion value save to NAI
            BCF         ADCON0,0
```

Figure 5.13: Subroutine of ADC

Figure 5.13 shows the most important code of this project. Without this code, the keypad will not function well. Bit 5-3 of ADCON0 was set to ‘000’ mean channel 0 (AN0, RA0) as analogue channel. Set bit 0 of ADCON0, the ADC will turned on. When the ADC is finished a conversion, the bit 2 of ADCON0 will turn to 0. Hence, a ‘test’ loop is performed to check whether the ADC conversion is done. After the conversion, the converted value that is stored to ADRESH register because ADFM flag which is bit7 in the ADCON1 was set to 0. If the ADFM flag is set to ‘1’, the converted value will save in ADRESL register. Lastly the converted value is stored in NADC register.

The NADC value next will be compare to the value range which is shown in Table 5.2 to determine which button is pressed. During the progress will check J register. If J is not 0, keypad button will be locked except ‘#’ and‘*’.

Figure 5.14 shows the source code flow for checking NADC value after get the NADC value from the ADC conversion.

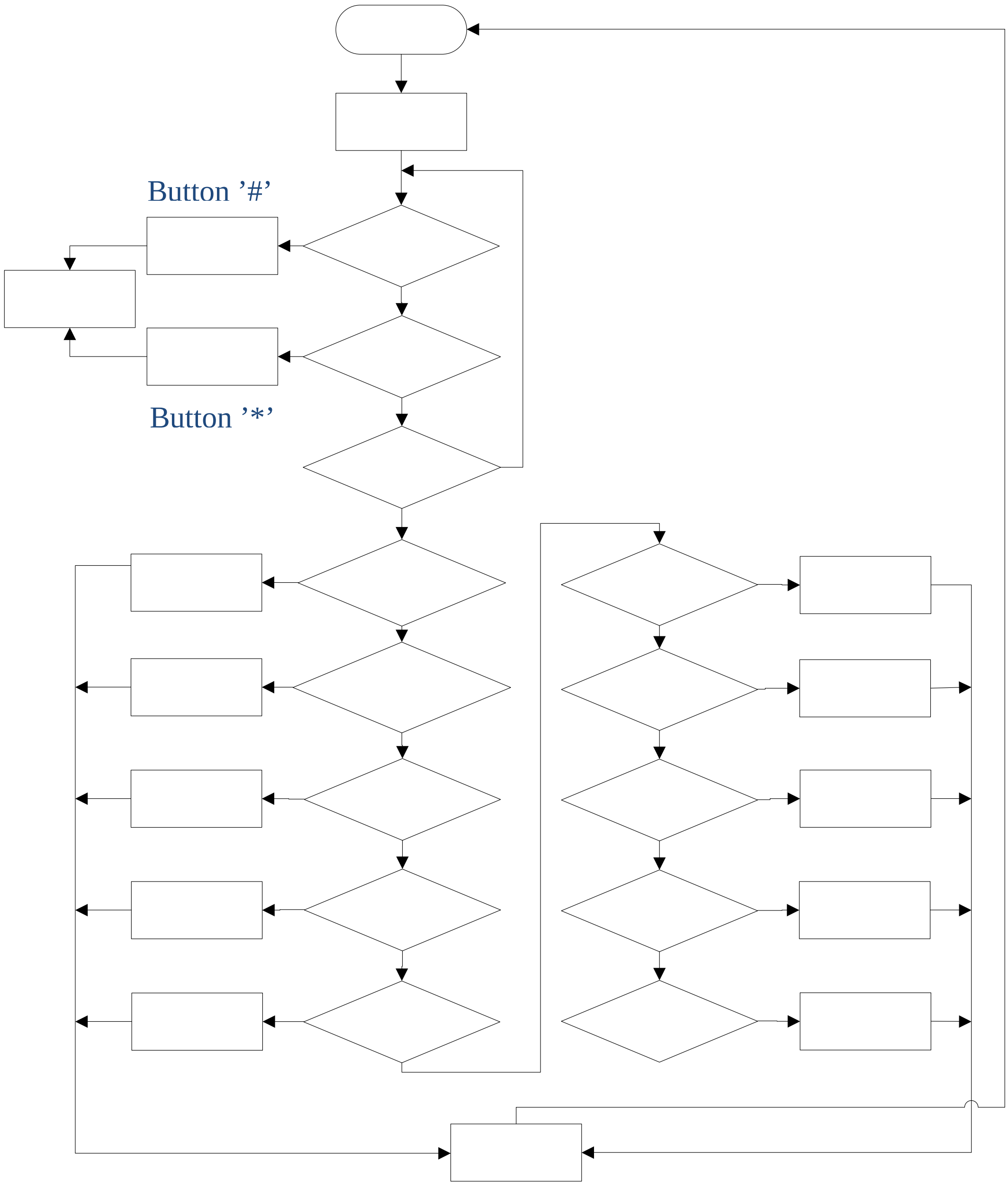

Figure 5.14: Program Flow of Checking Keypad Button

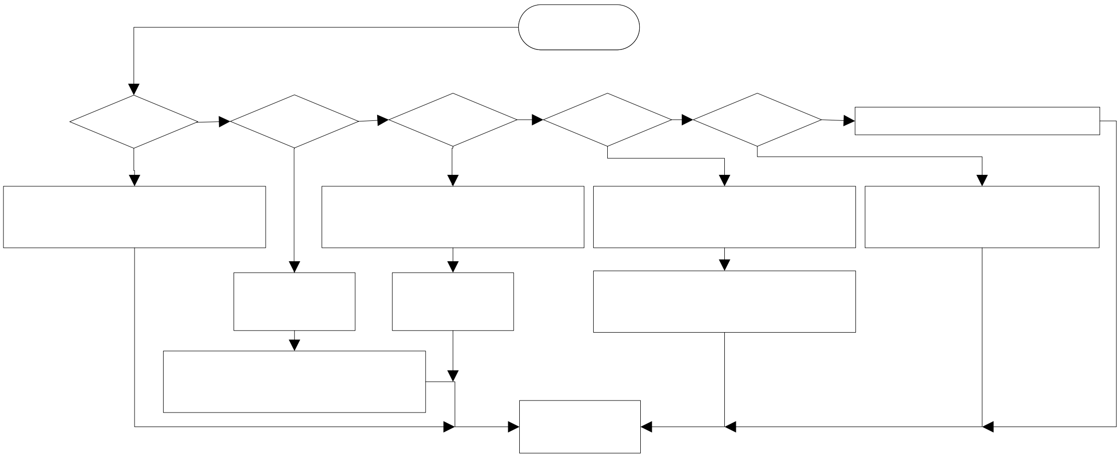

Figure 5.15: Program Flow of ADD Function

Figure 5.15 shows the ADD function which will be called when the number of keypad is pressed. The ADD functions basically just some way to add up the value that is pressed by user into H_NIBBLE1, L_NIBBLE1 or EX_NIBBLE1 register. Hence the program will know the total value that entered by users. The main menu of IC functional tester will show the choice selection of PC mode or Portable mode. PC mode is selected when * button is pressed while Portable mode is selected when the # button is pressed.

5.4.1.6 PC Mode Data Receive, Transmit and Check

Under PC mode, the RX and TX loop are used to receive and transmit data from PC as shown in Figure 5.16 and Figure 5.17

```
;*************** Receive data from PC **************
RX          NOP
            BTFSS       PIR1,RCIF
            GOTO        RX
            MOVF        RCREG,W
            MOVWF       L_NIBBLE
            RETURN
```

Figure 5.16: Subroutine of Data Receiving From PC

For data receiving from PC, the program will continuously check the RCIF (USART Receive Interrupt Flag bit) of PIR1 (Peripheral Interrupt Register). The RCIF will be set which mean receive buffer is full and data is received from PC. The receive data stored in the RCREG (USART Receive Register) will move to register W and then the data in the W will move to register name as L_NIBBLE.

```
;*************** Transmit data to PC **************
TX          MOVWF       TXREG
WAIT        NOP
            BTFSS       PIR1,TXIF
            GOTO        WAIT
            RETURN
```

Figure 5.17: Subroutine of Data Transmission to PC

For the data transmission, the data in register W will move into the TXREG (USART Transmit Register). Hence, the program will check the TXIF (USART Transmit Interrupt Flag bit) of PIR1. TXIF will be set when the data is sent. Afterwards the program will return to the subroutine which calls the transmission subroutine.

```
CALL        RX
MOVF        L_NIBBLE,W
XORLW       0xAA
BTFSC       STATUS,Z
GOTO        pc_PRE
MOVF        L_NIBBLE,W
XORLW       0xDD
BTFSC       STATUS,Z
GOTO        pc_CUS
MOVF        L_NIBBLE,W
XORLW       0xEE
BTFSC       STATUS,Z
GOTO        Start
```

Figure 5.18: Code of PC mode Checking

When users selected the PC mode, the 16F877A microcontroller will ready to receive data from PC. The 16F877A microcontroller will detect hexadecimal value either is ‘AA’, ‘DD’ or ‘EE’ as shown in Figure 5.18. If ‘AA’ is received means that the users selected the Preset Testing in the visual basic and LCD will display ‘PRESET’. ‘DD’ represented the users selected the Customized Testing in visual basic and ‘CUSTOMIZATION’ will be shown in LCD. ‘EE’ means the users had quit the Visual Basic (VB) program and LCD will display the main menu for user select PC mode or Portable mode.

While in the Preset Testing, the 16F877A microcontroller will detect hexadecimal ‘BB’ or ‘EE’. ‘BB’ represented the users selected back to visual basic menu for selecting the ‘PRESET’ or ‘CUSTOMIZATION’ Testing. ‘EE’ means quit the VB program. While in the Customized Testing, 16F877A will detect ‘CC’ or ‘EE’. ‘CC’ represented users selected back to visual basic menu for selecting the ‘PRESET’ or ‘CUSTOMIZATION’ Testing.

5.4.1.7 Model Checking Subroutine

```
;*************** MODEL CHECKING *****************
Chk_Model
            MOVF        L_NIBBLE,W
            XORLW       0x5C
            BTFSC       STATUS,Z
            GOTO        G_XNOR
            MOVF        L_NIBBLE,W
            XORLW       0x49
            BTFSC       STATUS,Z
            GOTO        G_JK
            MOVF        L_NIBBLE,W
            XORLW       0x86
            BTFSC       STATUS,Z
            GOTO        G_XOR
            MOVF        L_NIBBLE,W
            XORLW       0x74
            BTFSC       STATUS,Z
            GOTO        G_D
```

Figure 5.19: Subroutine of Model Checking

Neither in Preset Testing of PC Mode nor Portable Mode, Model Checking subroutine as shown in Figure 5.19 will be called. When the program completely received the IC model that users desire to test which will store in L_NIBBLE, L_NIBBLE will move into the register W. The program will compare the data in W with the hexadecimal value. If the data in W match with any hexadecimal value, the program will call the subroutine according to the hexadecimal value. The hexadecimal value represented each subroutine is listed in the Table 5.5.

Table 5.5: Hexadecimal Value for Each Logic IC

Hexadecimal value	Logic IC model
0x5C	XNOR
0x49	JK
0x86	XOR
0x74	D
0x08	AND
0x32	OR
0x04	NOT
0x02	NOR
0x00	NAND
0xFF	None of the model

In the subroutine of the logic IC model, first will declare the input output pin of the IC model. Then, the subroutine will just test each of the gates or flip-flops in the IC by sending the combination of bits into the inputs and detects the output to the desired value. Figure 5.20 shows the subroutine of XNOR gate.

```
G_XNOR

            BANK1
            MOVLW       B'00011001'
            MOVWF       TRISA
            MOVLW       B'11000000'
            MOVWF       TRISC
            MOVLW       B'00000110'
            MOVWF       TRISD
            MOVLW       B'00000100'
            MOVWF       TRISE
            BANK0
            CALL        G_XNOR00
            CALL        G_XNOR01
            CALL        G_XNOR10
            CALL        G_XNOR11
            RETURN
```

```
G_XNOR00
            BCF         PORTA,1
            BCF         PORTA,2
            BCF         PORTA,5
            BCF         PORTC,0
            BCF         PORTC,1
            BCF         PORTD,0
            BCF         PORTD,3
            BCF         PORTD,4
            BSF         PORTD,5
            BTFSS       PORTA,3
            INCF        G1
            BTFSS       PORTE,2
            INCF        G2
            BTFSS       PORTD,1
            INCF        G3
            BTFSS       PORTD,2
            INCF        G4
            RETURN
```

Figure 5.20: Subroutine of XNOR Gate

The program in Figure 5.20 will set pin3 of PORT A, pin 1 and pin 2 of PORT D and pin 2 of PORT E as inputs. While the other pins are set as output pins except the pin 6 and pin 7 of PORT C which is used in serial port connection, pin 0 and pin 4 of PORT A which used for ADC keypad and signal from 18F2620. Next, the program will call the test subroutine that has input 00, 01, 10 and 11 into the input of the XNOR gates. In the test subroutine, the program will compare the result obtained and store it in a register.

5.4.1.8 Transmit Result to PC

```
PC_RESULT
            MOVF        G1,W
            ADDWF       G2,W
            ADDWF       G3,W
            ADDWF       G4,W
            ADDWF       G5,W
            ADDWF       G6,W

            MOVWF       G_RESULT
            XORLW       0x00
            BTFSC       STATUS,Z
            GOTO        PC_FREE_EROR
            MOVF        G1,W
            XORLW       0x00
            BTFSC       STATUS,Z
            GOTO        G2ERR
            MOVLW       0x65
            CALL        TX
```

Figure 5.21: Subroutine of Result Transmission to PC

Under the Preset Testing of PC mode, after the Model Checking subroutine, result will be transmitted to PC during the PC_RESULT subroutine. Figure 5.21 shows the subroutine will start by adding up the register G1 till G6 and store the value into G_RESULT. If the G_RESULT is ‘0’ which means no error detect in the IC’s gate of flip-flop. Then the program will go to PC_FREE_EROR and send hexadecimal 0x64 to PC. If the G_RESULT value not equal to ‘0’, next will check G1 till G6 individually to determine which gate is fail. Table 5.6 shows the hexadecimal value that will be send to PC for represented which gates is failed.

Table 5.6: Hexadecimal Value of Each Error Gate

Hexadecimal value	Error Gate
0x64	No error
0x65	Gate 1 Error
0x66	Gate 2 Error
0x67	Gate 3 Error
0x68	Gate 4 Error
0x69	Gate 5 Error
0x6A	Gate 6 Error

5.4.1.9 Customized Testing

```
CUSTOMIZE

        ;Output setting
            MOVF        L_NIBBLE,W
            XORLW       0xC0
            BTFSC       STATUS,Z
            CALL        C0

            MOVF        L_NIBBLE,W
            XORLW       0xC1
            BTFSC       STATUS,Z
            CALL        C1
```

```
        ;Set '1'
            MOVF        L_NIBBLE,W
            XORLW       0x0C
            BTFSC       STATUS,Z
            BSF         PORTE,0

            MOVF        L_NIBBLE,W
            XORLW       0x1C
            BTFSC       STATUS,Z
            BSF         PORTA,1
```

```
        ; Send Result
            MOVF        L_NIBBLE,W
            XORLW       0xF0
            BTFSC       STATUS,Z
            CALL        F0

            MOVF        L_NIBBLE,W
            XORLW       0xF1
            BTFSC       STATUS,Z
            CALL        F1
```

```
        ;Set '0'
            MOVF        L_NIBBLE,W
            XORLW       0x0D
            BTFSC       STATUS,Z
            BCF         PORTE,0

            MOVF        L_NIBBLE,W
            XORLW       0x1D
            BTFSC       STATUS,Z
            BCF         PORTA,1
```

Figure 5.22: Subroutine of Customized Testing

In the Customized Testing of PC Mode, the program will run in the subroutine which is shown in Figure 5.22 and can be divided into 7 parts which are:

i. Output Setting
ii. Input Setting
iii. Clock Setting
iv. Vcc Setting
v. Set ‘0’
vi. Set ‘1’
vii. Send Result.

The codes for the 7 parts are the same except for Set ‘0’ and Set ‘1’ as shown in Figure 5.22. There are different hexadecimal value will receive from PC to declare with setting that users need to set for their testing. Hence, Table 5.7 shows the different function of each hexadecimal value.

Table 5.7: Function and Affected 16F877A Port For Hexadecimal Value

Function	Hexadecimal value	Port of 16F877A	Pin of IC	Hexadecimal value	Port of 16F877A	Pin of IC
Set pin as Input	0xD0	E0	1	0xB0	D5	15
	0xD1	A1	2	0xB1	D4	14
	0xD2	A2	3	0xB2	D3	13
	0xD3	A3	4	0xB3	D2	12
	0xD4	E2	5	0xB4	D1	11
	0xD5	A5	6	0xB5	D0	10
	0xD6	C0	7	0xB6	C1	9
Set pin as Output	0xC0	E0	1	0xA0	D5	15
	0xC1	A1	2	0xA1	D4	14
	0xC2	A2	3	0xA2	D3	13
	0xC3	A3	4	0xA3	D2	12
	0xC4	E2	5	0xA4	D1	11
	0xC5	A5	6	0xA5	D0	10
	0xC6	C0	7	0xA6	C1	9

Function	Hexadecimal value	Port of 16F877A	Pin of IC	Hexadecimal value	Port of 16F877A	Pin of IC
Set pin as Clock	0xC7	E0	1	0xE7	D5	15
	0xC8	A1	2	0xE8	D4	14
	0xC9	A2	3	0xE9	D3	13
	0xCA	A3	4	0xEA	D2	12
	0xCB	E2	5	0xEB	D1	11
	0xCD	A5	6	0xEC	D0	10
	0xCE	C0	7	0xED	C1	9
Set pin as Vcc	0xA8	E0	15	-	-	-
Set 0 to pin	0x0C	E0	1	0x0A	D5	15
	0x1C	A1	2	0x1A	D4	14
	0x2C	A2	3	0x2A	D3	13
	0x3C	A3	4	0x3A	D2	12
	0x4C	E2	5	0x4A	D1	11
	0x5C	A5	6	0x5A	D0	10
	0x6C	C0	7	0x6A	C1	9
Set 1 to pin	0x0D	E0	1	0x0B	D5	15
	0x1D	A1	2	0x1B	D4	14
	0x2D	A2	3	0x2B	D3	13
	0x3D	A3	4	0x3B	D2	12
	0x4D	E2	5	0x4B	D1	11
	0x5D	A5	6	0x5B	D0	10
	0x6D	C0	7	0x6B	C1	9

Table 5.7 shows all the hexadecimal value that will be received by 16F877A microcontroller from PC in the Customized Testing. For example, when the users set

pin 2 of logic IC in VB to input, PC will send a hexadecimal value of 0xD1 to 16F877A microcontroller and subroutine D1 will been called by microcontroller . Therefore, the microcontroller will clear bit 1 of TRISA register. There is another way can say that it set RA1 as output. Next, when the user set the value of pin 2 of logic IC in VB to 1, 0x1D will send to 16F877A and so the command of ‘BSF PORTA, 2’ is run to set the output pin as Figure 5.22.

```
; Send Result
    MOVF        L_NIBBLE,W
    XORLW       0xF0
    BTFSC       STATUS,Z
    CALL        F0

    MOVF        L_NIBBLE,W
    XORLW       0xF1
    BTFSC       STATUS,Z
    CALL        F1
```

```
;Send Result to PC
F0          BTFSS       PORTE,0
            GOTO        OUT_0
            GOTO        OUT_1

F1          BTFSS       PORTA,1
            GOTO        OUT_0
            GOTO        OUT_1

OUT_0
            MOVLW       0x00
            PAGESEL     TX
            CALL        TX
            PAGESEL     OUT_0
            RETURN

OUT_1
            MOVLW       0x01
            PAGESEL     TX
            CALL        TX
            PAGESEL     OUT_1
            RETURN
```

Figure 5.23: Subroutine of Result Transmission to PC in Customized Testing

Figure 5.23 shows the subroutine will run by 16F877A microcontroller when users requested the result in the Customized Testing. The PC will send hexadecimal value from F0 till F6 or E0 till E6 depend on which pins is output. When the 16F877A receives the hexadecimal, for example F1, it will call the subroutine of F1. The subroutine will check the related pin either is ‘0’ or ‘1’, for this case the subroutine of F1 will check Pin PORT A1. If the pin detected as ‘0’ which means result from output gate pin is ‘0’. For ‘0’, PIC16F877A will transmit hexadecimal 0x00 to PC. While ‘1’, microcontroller will transmit 0x01 to PC.

5.4.1.10 LCD Character Display

```
E_PULSE     MACRO               LCD_MSG5
BSF         PORTD,7                         MOVLW     A'F'
PAGESEL     DELAY5                          CALL      LCD_DISP
CALL        DELAY5                          MOVLW     A'A'
BCF         PORTD,7                         CALL      LCD_DISP
ENDM                                        MOVLW     A'I'
                                            CALL      LCD_DISP
TOLCD       MACRO                           MOVLW     A'L'
BSF         PORTD,6                         CALL      LCD_DISP
MOVWF       PORTB                           MOVLW     A'!'
ENDM                                        CALL      LCD_DISP
                                            MOVLW     A'!'
                                            CALL      LCD_DISP
                                            RETURN

                                LCD_DISP
                                            TOLCD
                                            E_PULSE
                                            RETURN
```

Figure 5.24: Code of Character Displaying on LCD

Figure 5.24 shows the code of character displaying on LCD. If it want to display character ‘F’ on the LCD, first it must be set the RS pin which is connected to bit 6 of PORTD of 16F877A. Next, the 16F877A microcontroller will supply a clock pulse to the E pin of LCD which connected to bit7 of PORTD of 16F877A. Through this way, the LCD will successfully display the character ‘F’.

5.4.1.11 Get Number of Loop For Testing

ADC keypad play important role in the Portable Mode. After the users enter the IC model which they wish to test, users will be requested to enter the number of testing. If users enter 0, the tester will not perform any checking on the model and will straight go to quit menu. If the users enter 1, the tester will run the test for once as shown in Figure 5.8. If users enter more than 1, the tester will keep on looping the check subroutine and save the result until the number of testing is reached.

In this section will explain the code as Figure 5.25, getting number of loop from keypad that enters by users. In section 5.4.1.5 already explained in detail how to detect the keypad button and save the number that pressed. After user entered the number of loop, the number will save in H_NIBBLE1 register. For example, users enter 33 for the number of loop, H_NIBBLE1 will store hexadecimal of 0x33 which is 41 for decimal. The step changes the hexadecimal to decimal value which user entered will be shown as below:

I. COUNTER = 00110011 & 11110000 (0x33 = 00110011)
= 00110000

II. COUNTER = 00000011 = 0x03

III. TIMES = 6 × 0x03 = 0x12

IV. H_NIBBLE1 = H_NIBBLE1 – TIMES
= 0x33 – 0x12
= 0x21
= 33

V. TIMES = H_NIBBLE1 = 33

TIMES value is used as a counter for loop testing.

```
        MOVLW   b'11110000'
        ANDWF   H_NIBBLE1,W
        MOVWF   COUNTER
        SWAPF   COUNTER
        MOVF    COUNTER,W ; if the counter = 0, skip the loop
        XORLW   0x00
        BTFSC   STATUS,Z
        GOTO    LESS_10
loop_1
        MOVLW   D'6'
        ADDWF   TIMES,F     ; TIMES = COUNTER x 6
        DECFSZ  COUNTER,F   ; Check COUNTER = 0?
        GOTO    loop_1
LESS_10
        MOVF    TIMES,W
        SUBWF   H_NIBBLE1,F ; H_NIBBLE1 = H_NIBBLE1 - TIMES
        MOVF    H_NIBBLE1,W
        MOVWF   TIMES
```

Figure 5.25: Code of Getting Number From Keypad

5.4.1.12 Portable Mode IC Model Checking

As 5.4.1.7 subsection have mentioned about Chk_Model subroutine which is a subroutine that performs the IC model checking for either Portable or PC mode, the program will perform some checking before go into Chk_Model subroutine for Portable mode. Those checking used to check whether the entered IC model by users in the listed model or not.

```
            MOVF        H_NIBBLE,W
            XORLW       0x74
            BTFSC       STATUS,Z
            GOTO        CHECK1;if 74 is entered, proceed to next checking
            GOTO        CHK_FAIL ;no model detect
CHECK1
            MOVF        EX_NIBBLE,W
            XORLW       0xAA
            BTFSC       STATUS,Z
            GOTO        CHK_OK ;proceed to model checking
            GOTO        CHK_FAIL ;no model detect
```

Figure 5.26: Subroutine of IC Model Checking

Figure 5.26 shows the value for H_NIBBLE register will be checked first whether the users entered first and second numbers were hexadecimal 74 or not. If it is correct, the programs will next check the EX_NIBBLE register is hexadecimal AA or not. If EX_NIBBLE is 0xAA, the program will continue check the entered model. If both of the H_NIBBLE and EX_NIBBLE register are not correct, the program will go into the CHK_FAIL subroutine which will display "Model Entered is Not Available" in LCD

5.4.1.13 Portable Mode Result Subroutine

After the Chk_Model subroutine which mention in section 5.4.1.7, the program will call the G_RESULTs subroutine to display result of testing on LCD. As Figure 5.27 shows, the subroutine will first check which model of IC being tested and display the IC model on LCD. Next the subroutine will checked the G_RESULT register. If the register is ‘0’, LCD will continue display “PASS”. While the register not ‘0’, “FAIL” will shows on LCD. If the IC is failed, the subroutine will next show the error gates or flip-flops in the IC.

```
G_RESULTs
            MOVF        G1,W
            ADDWF       G2,W
            ADDWF       G3,W
            ADDWF       G4,W
            ADDWF       G5,W
            ADDWF       G6,W

            MOVWF       G_RESULT
            PAGESEL     LCD_MSG3
            CALL        LCD_MSG3

            CLRF        PCLATH
            MOVF        L_NIBBLE,W
            SUBLW       0x5C
            BTFSS       STATUS,Z
            GOTO        CHK_JK
            PAGESEL     LCD_XNOR
            CALL        LCD_XNOR
CHK_JK
            CLRF        PCLATH
            MOVF        L_NIBBLE,W
            SUBLW       0x49
            BTFSS       STATUS,Z
            GOTO        CHK_XOR
            PAGESEL     LCD_JK
            CALL        LCD_JK
```

```
CHK_FINISH
            CLRF        PCLATH
            MOVF        G_RESULT,W
            XORLW       0x00
            BTFSC       STATUS,Z
            GOTO        G_FREE_EROR
            PAGESEL     LCD_MSG5
            CALL        LCD_MSG5
            CLRF        PCLATH
            CALL        ADC
            PAGESEL     LCD_MSG6
            CALL        LCD_MSG6
            CLRF        PCLATH
            MOVF        G1,W
            XORLW       0x00
            BTFSC       STATUS,Z
            GOTO        G_EROR2
            MOVLW       A'1'
            PAGESEL     LCD_DISP
            CALL        LCD_DISP
            MOVLW       A' '
            PAGESEL     LCD_DISP
            CALL        LCD_DISP
G_EROR2
            CLRF        PCLATH
            MOVF        G2,W
            XORLW       0x00
            BTFSC       STATUS,Z
            GOTO        G_EROR3
            MOVLW       A'2'
            PAGESEL     LCD_DISP
```

Figure 5.27: Result Subroutine in Portable Mode

5.4.1.14 IC Model and Gate's Result Transmission to PIC18F2620

```
;****************** SEND DATA TO 18I
SEND_18
            BANK0
            BSF         PORTC,5;tel

            MOVF        L_NIBBLE,W
            SUBLW       0x5C
            BTFSS       STATUS,Z
            GOTO        SEND18_JK
            BSF         PORTC,2
            BCF         PORTC,3
            BCF         PORTC,4
            BCF         PORTE,1
            CALL        Wait1
SEND18_JK
            MOVF        L_NIBBLE,W
            SUBLW       0x49
            BTFSS       STATUS,Z
            GOTO        SEND18_XOR
            BCF         PORTC,2
            BSF         PORTC,3
            BCF         PORTC,4
            BCF         PORTE,1
            CALL        Wait1
```

```
                 .
                 .
                 .
                 .
                 .
SEND18_B
            BCF         PORTC,2
            BCF         PORTC,3
            BCF         PORTC,4
            BCF         PORTC,5;
            CALL        Wait1
            RETURN

Wait1
            BTFSS       PORTA,4;
            GOTO        Wait1
            RETURN
```

Figure 5.28: Subroutine of IC model and Gate's Result Transmission to PIC18F2620

As shown in Figure 5.28, once SEND_18 subroutine is called, bit 5 of PORTC is set to '1' which informs 18F2620 microcontroller the data are ready to send. Afterwards, 4 bit data of IC models will first transfer through bit 2, 3, 4 of PORT C and bit 1 of PORTE to 18F2620 microcontroller. Next, the result of the particular gate will be sent to 18F2620 microcontroller as well. When all the data are finish sent, the 4 bit and bit 5 of PORTC will be set to '0'. 'Wait1 subroutine is used to wait the 18F2620 microcontroller send a signal to inform 16F877A microcontroller the data is saved into SD card. Table 5.8 shows the 4 bits data represented for each data.

Table 5.8: Declaration of Combination Bits For Each Sent Data

IC Model	Bit data				IC Model	Bit Data			
	E1	C4	C3	C2		E1	C4	C3	C2
XNOR	0	0	0	1	OR	0	1	1	0
JK	0	0	1	0	NOT	0	1	1	1
XOR	0	0	1	1	NOR	1	0	0	0
D	0	1	0	0	NAND	1	0	0	1
AND	0	1	0	1					

Error Gate	Bit Data			Error Gate	Bit Data		
	C4	C3	C2		C4	C3	C2
0	0	0	1	5	1	1	0
1	0	1	0	6	1	1	1
2	0	1	1				
3	1	0	0				
4	1	0	1				

5.4.2 18F2620 Source Code

The 18F2620 microcontroller basically is used for interface with SD card only. Due to the 16F877A lack of RAM to interface with SD card and can only save data in hexadecimal, so 18F2620 microcontroller is utilized. The program is written by using mikroC software. Based on textbook of “Advanced PIC Microcontroller Projects in C” and the Support Centre of mikroC , we learn how to write C language for SD card in the mikroC software and generated the HEX file.

Figure 5.29 shows the overall program flow for 18F2620 microcontroller which controlling the SD card. Before the SD card library functions are used, microcontroller will first initialize SPI setting (“Spi_Init_Advanced”) (SPI library) as Figure 5.30 because SD card interface with 18F2620 through SPI mode of microcontroller. The setting is set the SPI work in:

I. Master Clock = F_{osc} / 4 (“MASTER_OSC_DIV16”)
II. Input data sampled in middle of interval (“DATA_SAMPLE_MIDDLE”)
III. Clock idle LOW (“CLK_IDLE_LOW”)
IV. Data transmit on low to high edge (“LOW_2_HIGH”)

Next, microcontroller will check the SD card whether is finished initialize or not as the coding shows in Figure 5.31. If no SD card is inserted, the microcontroller will not able to finish performing the Mmc_Init routine (Multi Media Card Library (mC PRO for PIC), 2010) and the Mmc_Init routine will return ‘1’ hence a red LED will light on which is connected to bit 5 of PORTB. When the program finished initializes the SD card, Mmc_Init routine will return ‘0’ to the program and hence microcontroller will turn off yellow LED and lighted on red LED.

Next, the microcontroller will initialize the FAT file of the SD card as shows in Figure 5.32. If the FAT file is finished initializing, the program will return ‘0’ which mean the program is not running the Mmc_Fat_Init routine otherwise the program will return ‘1’.

When the FAT file of SD card finish initialized, 18F2620 microcontroller will check whether “ICTESTER.txt” file exists in the SD card or not. If the file not exists, Create_New_File routine will be called to create the file. If the file exists in the SD card, 18F2620 will just append data into the file. The Subroutine of this is showing in Figure 5.33.

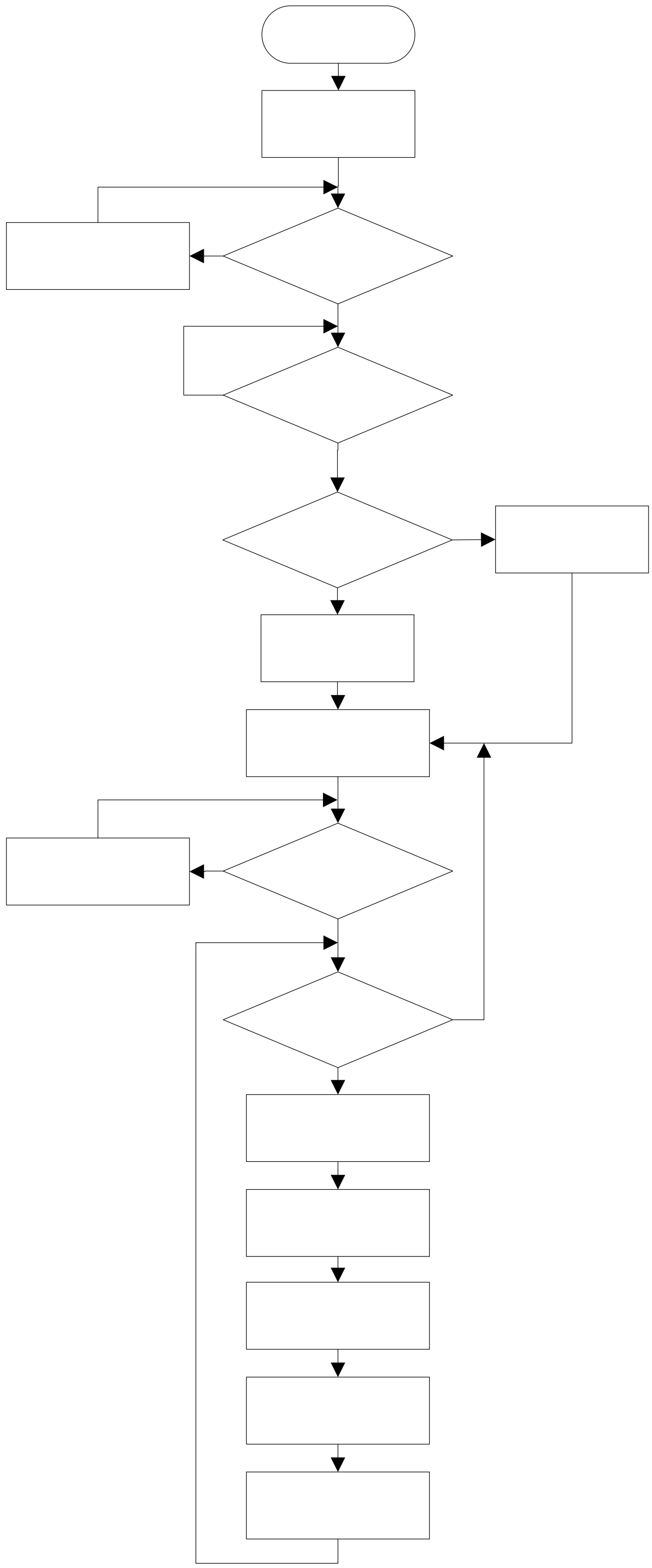

Figure 5.29: Overall Program Flow for PIC18F2620

```
//Initailise SPI bus
Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE, CLK_IDLE_LOW, LOW_2_HIGH);
```

Figure 5.30: Coding for Initialize SPI Bus of PIC18F2620

```
if(!Mmc_Init(&PORTC,2)){
    PORTB.f6 = 1;  //Detected memory card, red LED off, white on
    PORTB.f5 = 0;
    }

else{
    //no memory card detect red LED on
    PORTB.f5 = 1;
```

Figure 5.31: Coding for Initialize SD Card

```
// Initialise the FAT file system
while(1) {
if (!Mmc_Fat_Init(&PORTC,2)) {
```

Figure 5.32: Coding for Initialize FAT File System of SD Card

```
//Check the file exist in SD card or not
ret_status = Mmc_Fat_Assign(&filename,1);

if(!ret_status)
Create_New_File();      //Create new file
else
{
 Mmc_Fat_Assign(&filename,0);          //Append to the exist file
 Mmc_Fat_Append();
}
```

Figure 5.33: Subroutine of File Exist Checking

```
PORTB.f6 = 1;
PORTB.f5 = 0;

for(;;)
{
 while(Mmc_Init(&PORTC,2)){
  PORTB.f6 = 0;  //no memory carc
  PORTB.f5 = 1;
 }

if(Switch)
{
  No_Append();
  PORTB.f7 = 0;
  PORTB.f6 = 1;
  PORTB.f5 = 1;

  no_g = 0x00;
  delay_ms(5);
  no_g = PORTB & 0x1E;
  no_g = no_g>>1;
  switch(no_g)
  {
   case 0x01:
              XNOR_Append();
              PORTB.f7 = 0x01;
              break;
   case 0x02:
              JK_Append();
              PORTB.f7 = 0x01;
              break;
```

Figure 5.34: Subroutine of SD Card and Data Detection

When there is file existing, the yellow LED will be lighted on and red LED will be lighted off as shown in first two line of Figure 5.34. A checking of SD card whether is taken out from the holder will always be done in the endless ‘for’ loop. If the SD card is taken out, red LED will light ON and yellow LED is switched off. If SD card exists in the holder is confirmed, microcontroller will check the Switch register which is bit 0 of PORTB. Once 18F2620 microcontroller detects signal from bit 5 of PORTC for 16F877A microcontroller which is Switch, 18F2620 is ready to save the data that is sent from 16F877A. ‘No_Append’ subroutine is used to save list number into the text file. During the saving process, red and yellow LED will be lighted On. Once the number is saved, 18F2620 microcontroller will detect the data as shown in Table 5.8 from 16F877A which determines IC model. After the IC model in SD card is saved, 18F2620 will set bit7 of PORTB which is connected to bit 4 of PORTA for 16F877A to inform 16F877A that data is saved in SD card.

```
while(Switch)
{
  PORTB.f7 = 0;
  no_a = 0xFF;
  delay_ms(5);
  no_a = PORTB & 0x0E;
  no_a = no_a>>1;
  switch(no_a)
  {
   case 0x01:
                 Content0_Append();
                 PORTB.f7 = 0x01;
                 break;
   case 0x02:
                Content1_Append();
                PORTB.f7 = 0x01;
                break;
```

Figure 5.35: Subroutine of Gate's Result Saving to SD Card

After the IC model into SD card is saved, bit 7 of PORTB will be cleared and 18F2620 microcontroller is ready to receive the gate result from 16F877A microcontroller as shows in Figure 5.35. The different of gate result and IC model is the number of received data bits. IC model is determined by 4 bits while gate result is determined by 3 bits. Not only that, 'while loop' is used in receiving gate result because it needs to do a loop for receiving the gate's result while for IC model, it just need 1 loop. Others operation is totally same as the IC model saving to SD card.

CHAPTER 6

RESULT AND DISCUSSIONS

6.1 Overview

After we developed the codes for the microcontroller (Assembly Language and C Language) and the graphic user interface (Visual Basic), the Logic IC Functional Tester project works up and well functions. In this chapter, the obtained results from Logic IC Functional Tester will be discussed including the problems encountered.

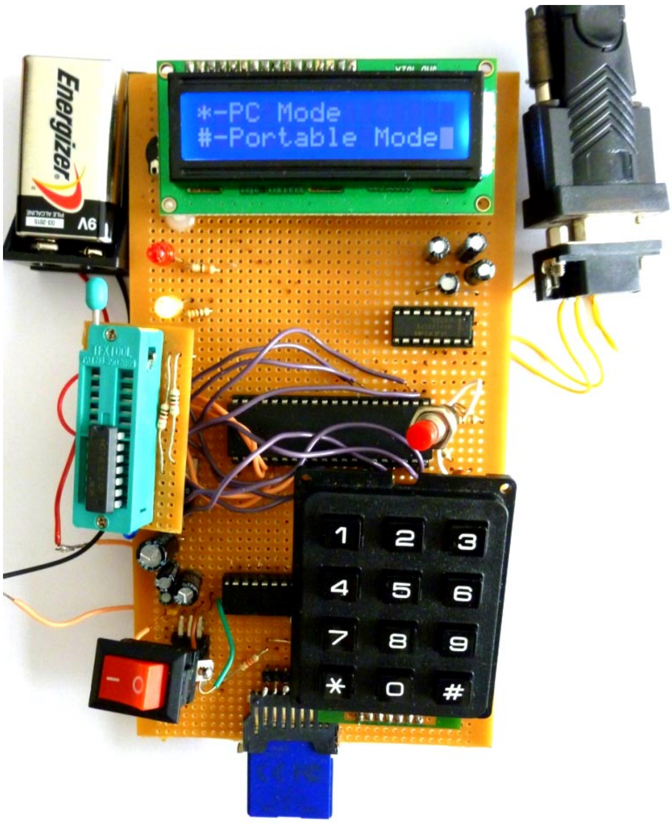

Figure 6.1: Logic IC Functional Tester Circuit

6.2 Portable Mode

There will be an indication shown on LCD for user to choose either processing the functional IC tester in Portable Mode or PC Mode as shown in Figure 6.2. For Portable Mode, users is required to enter the IC model which available for the Logic IC Functional tester via keypad. Figure 6.3 shows the user interface in Portable Mode.

Figure 6.2: Mode Selection

Figure 6.3: Entered IC Model

Figure 6.4: Number of Testing

After users entered available IC model as Figure 6.3 shown which entered model of D flip flop, they needed to press enter command via the "*" keypad button. Next, they are required to enter desired number of testing looping as shown in Figure 6.4. The number of testing is the results can be repeat generalized, which shows a degree of validity and reliability. It can assure quality results of IC Tester. For example, if number “4” is entered, program will loop the testing for 4 times and save the 4 times result into SD card.

The LCD will display “Model Entered Is Not Available” as shown in Figure 6.5 when users enter those ICs model which are not available for logic IC functional tester. IC models which are available to be tested are basic gates (74LS08 AND, 74LS32 OR, 74LS00 NAND, 74LS02 NOR, 74LS04 NOT, 74LS86 XOR, 74LS266 XNOR) and flip-flop ICs (74LS74 D-flip-flop, 74LS109 JK-flip flop).

Figure 6.5: Model Entered Is Not Available

Figure 6.6: All Result Saved

Figure 6.7: Continue or Exit Indication

After the looping test is finished and all the result will be saved into SD card for the specific IC, LCD will display “Result Saved” as shown in Figure 6.6. Lastly, the Logic IC Functional Tester will request users to choose whether want to exit the testing or continue testing.

If the users enter “1” for the “Enter Number of Testing” as shown in Figure 6.4, LCD will show the specific IC model’s name and the condition of IC’s gates. For example, the pass results of testing 74LS74 IC (D-flipflop) as shown in Figure 6.8. This shows that all the two internal gates in 74LS74 are working properly.

Figure 6.8: Result of Good IC

The fail results of testing 74LS74 IC (D-flipflop) as shown in Figure 6.9. The gate 2 in 74LS74 IC which is not functioning properly will be displayed in LCD as shown in Figure 6.10.

Figure 6.9: Result of Faulty IC

Figure 6.10: Result of Which Faulty Gates

Furthermore, this Logic IC Functional Tester provides the data storage for users which means the result of the testing can be saved in SD card. Hence, there is indication on LCD as shown in Figure 6.11 for user to choose either to save the result or skip the saving. If the option is selected then the file will be stored in memory card.

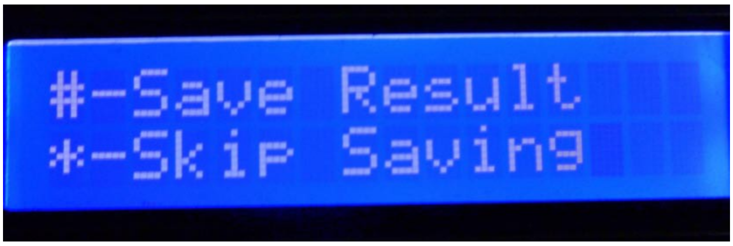

Figure 6.11: Saving Indication

After result is finish saved, a "Result Saved" indication will be shown on LCD as shown in Figure 6.6. "ICTESTER" file contents the 5 results as shown in Figure 6.12 which runs a number of 4 loops and runs 1 testing on D Flip Flop.

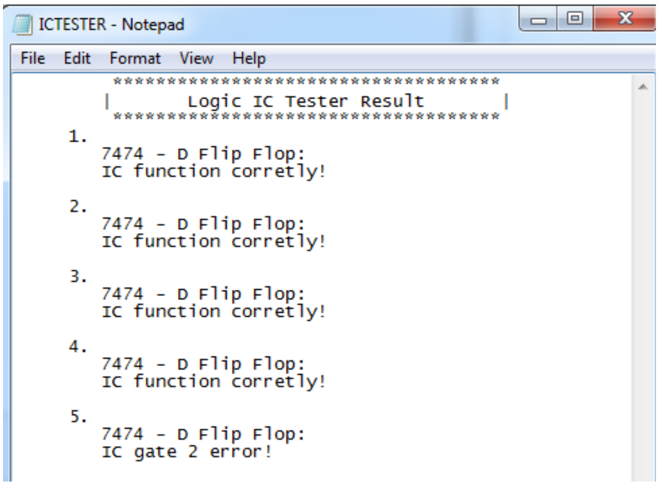

Figure 6.12: "ICTESTER" File

There is an indication as shown in Figure 6.7 for user to choose either continue next testing or exit the testing. If exit the testing is selected then it will return to Mode Selection interface as shown in Figure 6.2.

Figure 6.13 is showing the fail results of testing 74LS109 (JK-flip flop) in Portable Mode with 1 testing only. One of the pins is purposely being broke in order to create a faulty JK-flip flop IC. "JK FAIL!!" is shown on LCD means the 74LS109 is malfunction. Figure 6.14 shows one of the internal gates which is gate 2 in 74LS109 is not working properly.

Figure 6.13: “FAIL” Result on LCD (One Malfunction Pin)

Figure 6.14: Which Faulty Gates

Figure 6.15: “FAIL” Result on LCD (Entering Wrong IC Model)

Figure 6.15 is showing the result of testing 74LS02 in Portable Mode but with user intentionally entered 74LS08. In this condition, “FAIL” result will be displayed on LCD. This is due to different ICs have different internal gates and it will always lead to the internal gates to produce the incorrect output and subsequently produce a “FAIL” result.

Figure 6.16: Which Faulty Gates

Figure 6.16 is showing that the LCD indicated that there are four internal gates that producing incorrect output which is confirmed that the orientation of four internal gates of the 74LS02 is different compared with the internal gates of 74LS08.

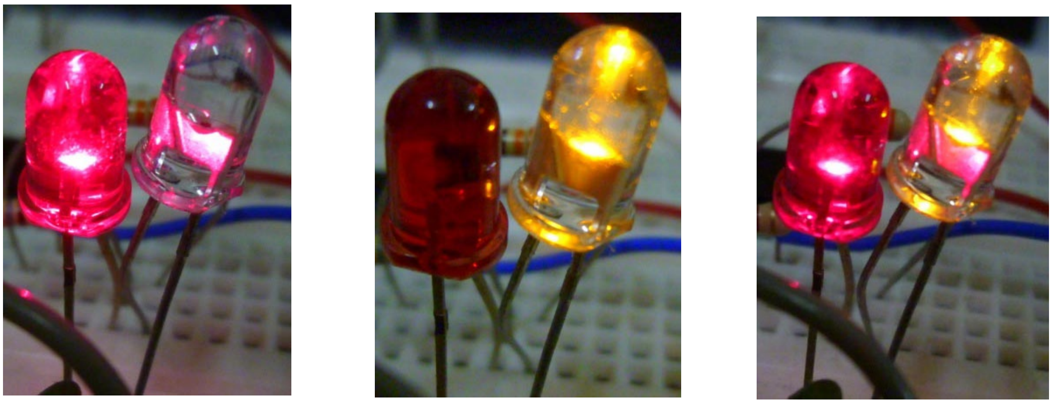

Figure 6.17: LED Indicator

When the SD card is not inserted in the memory card holder, the red LED will be lighted ON. Once 18F2620 microcontroller detected SD card, the red LED will be turned off and the yellow LED will be turned on. During the data or result are saving to the SD card, both LED will be lighted ON and red LED will be blinking frequently.

6.3 PC Mode

Users need to press '*' button in the main menu as shown in Figure 6.2 to run the Logic IC Functional Tester in PC mode. When the tester is operating under the PC mode, LCD will display as Figure 6.18. While at computer, after users enter the correct username and password for log in interface as shown in Figure 6.19, main menu will be shown as Figure 6.20.

Figure 6.18: Operating In PC mode

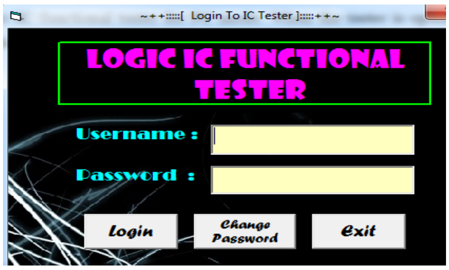

Figure 6.19: Log In Interface of Visual Basic

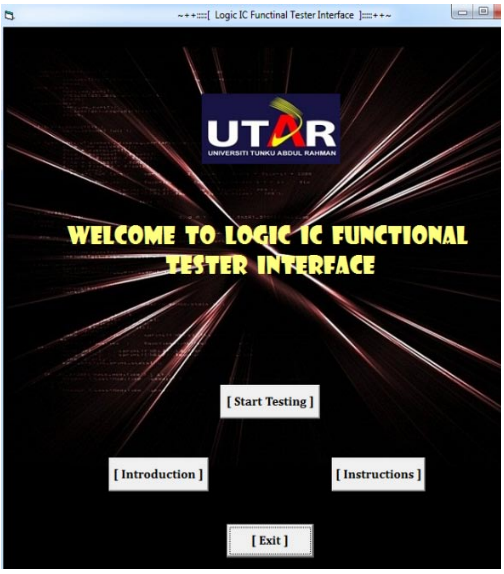

Figure 6.20: Main Menu Interface

When users selected Preset Testing in the mode selection interface as Figure 6.21, the interface of Preset Testing as Figure 6.22 will show on PC and LCD will display “Preset” as Figure 6.23.

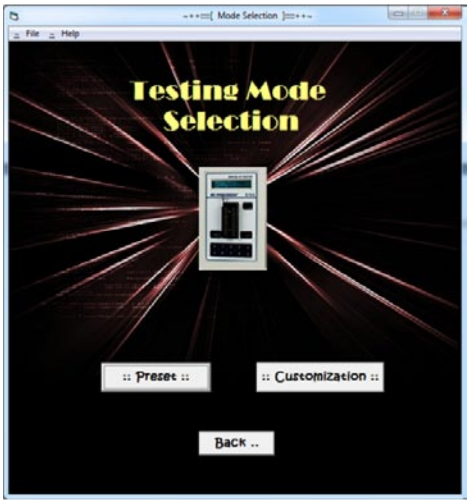

Figure 6.21: Mode Selection Interface

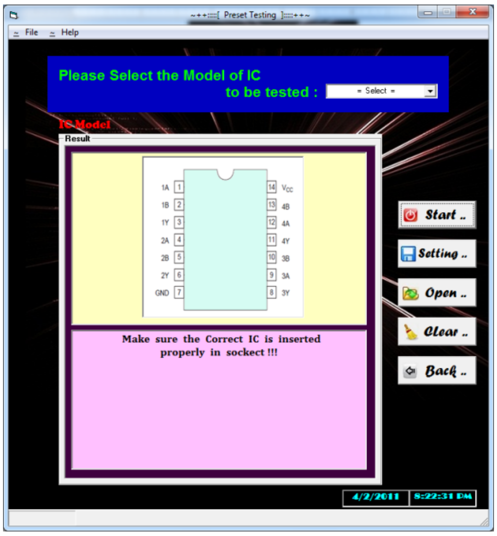

Figure 6.22: Preset Testing Interface

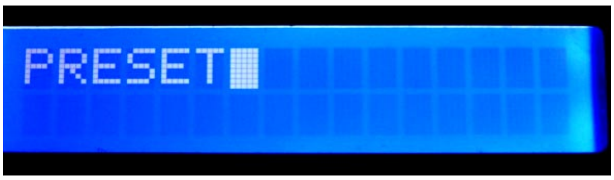

Figure 6.23: Preset Indication on LCD

While users selected Customized Testing in the Figure 6.21, the interface as Figure 6.24 will be shown in PC while "Customization" indication will display on LCD.

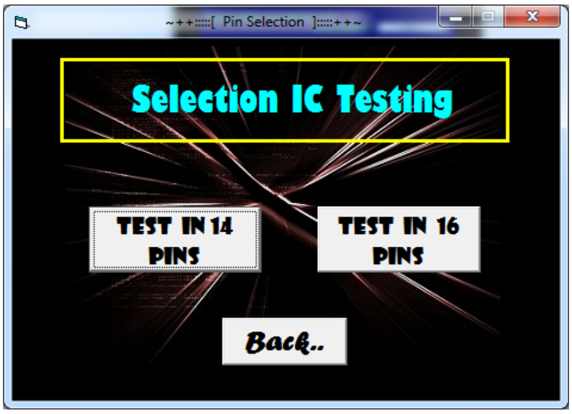

Figure 6.24: Customized Mode Selection

Figure 6.25: Customization Indication on LCD

When users quit the Graphic User Interface (GUI), LCD will show the main menu which to let users choose either is PC mode or Portable mode as shown in Figure 6.2. For the application of IC Tester's GUI and how does IC to be tested in PC Mode will be explained more detail and obviously by my partner (Misss Law) in her report.

6.4 Problem Encountered

When we start to develop the software and the hardware, we have encountered some problems during the implementation/development of the project. This section will explain the problems faced during the programming stage and what are the steps are taken to solve the problem.

Problem 1:

We are lack of ports in 16F877A microcontroller for controlling the 16 pins logic ICs. However, we do not want to use two microcontrollers to control ZIF socket because it is such tricky and complex to control by programming language.

Solution 1:

In order to save the I/O pins of microcontroller so we decided to connect keypad to an ADC with a resistor matrix in the project because it requires one dedicated I/O pin to function only rather than 7 I/O pins.

Problem 2:

Different IC models have different number of pins. The different of JK flip-flop IC with others is the numbers of pins. While the different of basic gates IC and flip-flop IC is the clock input.

Solution 2:

Hence we thought the other ways to solve this problem which is using program to control it. The problem can be solved by using different port declaration in program for different model of IC especially flip flop IC.

Problem 3:

During testing the each keypad buttons, the desired outcome displaying in the LCD are not consistent especially key “8”, “9”, “0”, “#”, “*”. For an example, when the key “0” button is pressed, the outcome displaying in the LCD is “9” which is wrong number.

Solution 3:

We found that it may possibly due to the tolerance value of the buttons. Hence, we try to add more delay on the codes and then always change the tolerance value of the troubled button until correctly outcomes.

Problem 4:

When the assembly language is become more and longer, while calling the subroutine and return from the subroutine especially send character for displaying in LCD, it displayed some unknown character.

Solution 4:

After studying from datasheet and researching from website, we found the existing error is due to codes over the memory of 1 page which is from 0000 till 07FF. Therefore, we need to rearrange the code to 2 pages which first page is 0000 till 07FF and second page is 0800 till 0FFF. Therefore, when calling subroutine in page 2 from page 1, "PAGESEL" command need to add before calling the subroutine.

Problem 5:

When done the USART setting in 16F877A, then try to interface with PC using the serial-to-USB cable, it is fail to get the desired result.

Solution 5:

At first we doubt is the cable problem, after checked the whole program, in the Chk_Model subroutine accidentally reset the bit 6 and bit 7 of PORT C become outputs. After recheck all the code and set the bits to input, problem is solved.

Problem 6:

After testing the 9 model logic ICs, we found that 74LS266 IC (XNOR) always shows error in all the four gates.

Solution 6:

When referring to the datasheet, we noticed that the output for the 74LS266 IC is open collector output. By connecting each output to a pull up resistor, the problem is solved. (International, 2010)

Problem 7:

When testing the button of ADC keypad, it will show multiple displays in LCD for 1 button.

Solution 7:

By adding delay of 0.2s in the ADC conversion, the problem is solved.

Problem 8:

When improving the Logic IC Functional Tester by adding multiple loop testing for the circuit, the 18F2620 seem cannot perform multiple data saving to SD card.

Solution 8:

I tried few solutions which changing the idea on writing receive and transmit code for both PIC, but it is still failed. Then, for PIC16F877A, before the SEND_18 subroutine returns to the called routine, WAIT_1 which is a subroutine to waiting signal from PIC18F2620 to inform that data is saved, added into the code, and add delay into the multiple looping, the problem is solved.

Problem 9:

The result of number of loop is running incorrectly when the desired number of loop for performing testing is entered.

Solution 9:

This is due to the number was hexadecimal. Therefore, by performing some mathematics ways to make it becomes decimal value which same as entered by ADC keypad, the tester is looping perfectly. The problem is solved.

CHAPTER 7

CONCLUSION

After the several months of development and debugging, finally the project has been successfully completed. The project's aim and the main objectives have been accomplished. The Logic IC Functional tester is basically a microcontroller-based project and gives response to the user within a few second. Using the computer, the tester still can function effectively through Graphical User Interface (GUI).

The Logic IC Functional tester is able to test basic gate (74LS08 AND, 74LS32 OR, 74LS00 NAND, 74LS02 NOR, 74LS04 NOT, 74LS86 XOR, 74LS266 XNOR) and flip-flop ICs (74LS74 D-flip-flop, 74LS109 JK-flip flop) sequence in 14 and 16pins respectively without any errors no matter in Portable Mode or PC Mode. Those ICs is not available in the Preset Testing can be test in Customized Testing, it is moderately user-friendly.

We utilized the programming language Visual Basic to help in design the graphic based windows interface and Assembly programming language for controlling overall IC tester circuit.

The microcontroller is programmed by using the MPLAB IDE. We simulated the code by using the simulation tools in the MPLAB before load it into the microcontroller.

The Graphical User Interface (GUI) is created by using Microsoft Visual Basic 6.0 software. The interface is designed as simple as possible in order to provide a user-friendly interface for users. In the interface, we also provided a set of general instructions to users for reference.

We used resistor matrix concept in keypad in order to save ports of PIC. Then, we decided to utilize SD card for our memory storage. The simplicity of the SD Card protocol and the flexibility in interfacing with these devices makes it ideal for us to use with small microcontrollers.

This report has highlighted many complexities and challenges that are faced in designing such a system. We can summarize the development of our final year project has been successfully achieved. The Logic IC Functional tester project was very successful and functional.

7.1 Future Recommendations

This project has an extremely large potential to be improved in many ways. The system has already been developed but much additional features can be built and integrated to this project so that it can work as a perfect whole. Below shows the few recommendations that can be added to improve the system.

One of the weaknesses of this IC Tester is that it can only test for 9 ICs in the Preset Testing. This is due to the insufficient I/O ports in PIC16F877A microcontroller for this IC Tester to test analogue and digital ICs with up to 40 pins. Furthermore, the IC tester can be improved to test for ICs not currently supported in the library can be added on request, including op-amps, comparators, voltage regulators, voltage references, analogue switches and multiplexers, and couplers, and audio ICs. Hence, we suggested in order countering this problem, a bigger (more pins) and more advanced microcontroller should be utilized to implement such IC Tester.

The IC tester project can also be enhanced to be able to identify unmarked devices listing through auto-identification mode and detects intermittent and temperature related faults so that a more powerful and functional IC tester can be provided for users.

Moreover, the wrong insertion of the IC can be detected by programming in microcontroller. For instance, when an IC is inserted incorrectly, buzzer will be sounded to inform users even the LCD will display “IC is inserted incorrectly”.

The graphic user interface can be formed to be more graphic and creative so that providing an outstanding interface in quality and clarity to user. The interface must be a clear and direct interface however it can come in many styles ranging graphically from very simple. Besides, the applications in this interface can be enhanced such as attach a video in Instruction form to show how to test an IC by using this IC tester even attach result printing feature in interface.

REFERENCES

Ahmed, M.S., Umair, I.M., & Mehboob, K. (2005). Microcontroller Based IC Tester. *Engineering Sciences and Technology, 2005. SCONEST 2005. Student Conference on* , -(-), 1-4.

Ali, L., Sidek, R., Aris, I., Wagiran, R. & Ali, M.A.M. (2004). Design of a SOC for low cost IC testing. *Semiconductor Electronics, 2004. ICSE 2004. IEEE International Conference on* , -(-), 5.

BK 575A Digital IC Tester. (2008). Retrieved August 2010, from http://www.tequipment.net/BK575A.html.

Schimidt,G. (2009). *Connecting a keypad to an AVR.* Retrieved from http://www.avr-asm-tutorial.net/avr_en/keypad/keyboard.html.

How to Use MMC/SDC. (2010). Retrieved November 3, 2010, from http://elm-chan.org/docs/mmc/mmc_e.html.

Ibrahim, D. (2008). Advance PIC18 Projects-SD Card Projects. In *Advance PIC Microcontroller Projects in C*. USA: Newnes.

International, J. (2010). *Open Collector Output.* Retrieved March 15, 2011, from http://www.jlcinternational.com/application_notes/Open_collector_output.pdf.

mikroC PRO for PIC. (2011). Retrieved from http://www.mikroe.com/eng/products/view/7/mikroc-pro-for-pic/.

Model 570A - Linear IC Tester. (2008). Retrieved 2008, from http://www.bkprecision.com/products/model/570A/linear-ic-tester.html.

MPLAB Integrated Development Environment. (2011). Retrieved from http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002.

Muhammad Ali Mazidi, Rolin D. Mckinlay, & Danny Causey. (2008). *PIC Microcontroller and Embedded System.* New Jersey: Pearson Education.

Multi Media Card Library (mC PRO for PIC). (2010). Retrieved January 2011, from http://www.mikroe.com/esupport/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=139.

Nolan, L., Moi-Tin Chew, Demidenko, S. & Ooi, M. (2010). Virtual Instrumentation Based IC Parametric Tester for Engineering Education. *Electronic Design, Test and Application, 2010. DELTA '10. Fifth IEEE International Symposium on*, -(-), 310-315.

PICkit2 Development Programmer/Debugger. (2011). Retrieved from http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023805.

Proteus VSM. (2010). Retrieved from http://www.labcenter.com/products/vsm_overview.cfm.

Rajsuman, R.(2002). Can IC test learn from how a tester is tested. *Test Conference, 2002. Proceedings. International*, -(-), 1186.

Ranjith. (2008). *Digital IC Tester*. Retrieved September 2010, from http://electrofriends.com/projects/microcontrollers/digital-ic-tester/.

SPI library. (2010). Retrieved from http://www.mikroe.com/esupport/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=150.

Strunz R, Toal D, McGowan C. (2010). *University of Limerick*. Retrieved June 8, 2010, from http://www.ul.ie/~toald/Publications/Imc.pdf.

SU-300 stand-alone IC Programmer. (n.d.). Retrieved from http://www.sumberpanca.com/product_info.php?cPath=67&products_id=243.

West, Gary L., Nagle, H. Troy, & Nelson, Victor P. (1980). A Microcomputer-Controlled Testing System for Digital Integrated Circuits. *Industrial Electronics and Control Instrumentation, IEEE Transactions on* , *IECI-27*(4), 279-283.