DEVELOPMENT OF ASYNCHRONOUS SERIAL DATA COMPARATOR USING FPGA

ARWENA BINTI ABDULLAH

Faculty of Electrical & Electronics Engineering

Kolej Universiti Kejuruteraan & Teknologi Malaysia

(University College of Engineering & Technology Malaysia)

KOLEJ UNIVERSITI KEJURUTERAAN & TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DEVELOPMENT OF ASYNCHRONOUS SERIAL DATA COMPARATOR USING FPGA**

SESI PENGAJIAN: **2005/2006**

Saya **ARWENA BINTI ABDULLAH (820519-02-5866)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Kolej Universiti Kejuruteraan & Teknologi Malaysia.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

NO:27,LORONG INTAN A/8
TAMAN MEWAH,08000,
SUNGAI PETANI,KEDAH

MD RIZAL B OTHMANL
(Nama Penyelia)

Tarikh: **20 APRIL 2006**

Tarikh: : **20 APRIL 2006**

CATATAN:
- * Potong yang tidak berkenaan.
- ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
- ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

Signature : _____

Author : ARWENA BINTI ABDULLAH

Date : 20 APRIL 2006

DEVELOPMENT OF ASYNCHRONOUS SERIAL DATA COMPARATOR USING FPGA

ARWENA BINTI ABDULLAH

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Kolej Universiti Kejuruteraan & Teknologi Malaysia

APRIL, 2006

ACKNOWLEDGMENT

In the name of Allah swt the Beneficient, the Merciful…

There are just a few of the many important people who have provided enormous contribution towards the success of this thesis. I truly appreciate all their contribution and would like to express our sincere indebtedness to them . Their contribution either in the forms of comments, suggestions and even criticisms is highly appreciated. For me, this work is not my achievement rather it comprises of effort and cooperation from them.

Firstly, I would like to extend our sincere appreciation to our academic associates, especially to the supervisor Encik Md Rizal B Othman for his continuous support and encouragement. Secondly, to the Dean of the Faculty of Electrical & Electronic Engineering. Besides that, I would like to thanks our lecturers , staff of the laboratory of electrical and electronic engineering, my parents and to my friends for their valuable support, contribution and help.

Finally, thanks go to the anonymous editor and manuscript reviewer from books and internet who had help to improve the final version of this thesis.

To all those mentioned and many more, I personally indebted to your invaluable cooperation. I take full responsibility for any errors, omissions, and negligence in this thesis. My sincere hope that, after reading this thesis it would give positive feedback about this project.

ABSTRAK

Penggunaan FPGA yang lebih mudah dan semakin meluas dalam aplikasi harian membuka peluang kepada para pelajar untuk lebih mempelajari penggunaan dan aplikasi FPGA dalam bidang elektronik. Terdapat banyak aplikasi yang boleh diimplementasikan ke dalam FPGA dan salah satu contohnya adalah dalam bidang Penghantaran perbandingan data sesiri secara asynchronous

Secara ringkasnya, projek ini bertujuan untuk mengkonfigurasi dan mengawal papan FPGA sebagai litar luaran tambahan untuk berkomunikasi antara MAX233 dengan PC untuk penghantaran data secara sesiri untuk menyimpan dan mengeluarkan data dalam pengunaan perbandingan ID kod dengan ID nombor.Program ini dibina dalam proses digit system dimana data adalah dalam bit bit.Data akan dikenali dengan 8 bit dan akan diperkenalkan sebagai data bit yang akan menentukan ID nombor dengan kod yang dihantar.Setiap bit ID nombor akan disimpan di dalam flipflop yang berbeza oleh 8 daftar anjak.Sekiranya keluaran flipflop adalah sama dengan ID nombor,litar akan menghasilkan paparan logik tinggi.

ABSTRACT

The wide and simpler usage of FPGA in everyday applications opened the door for students to study its usage and applications in electronics. There are many applications that can be implemented in FPGA and one of them is in developing asynchronous serial data comparator using FPGA.

This project is involved the design, implementation and test of a digital logic system by the application of the FPGA as the programmable gate logics where will be implemented as the program to communicate with the MAX233 and the PC for serial data comparator transferring and to control in storing and accessing the data in the application of comparison ID code and ID number. The program developed is processed in digital form that is the data bit generated is in digital. It will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code. Each bit of the ID number will stored in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output. This circuitry then will burn in to the ALTERA and will communicate with the data that transmit.

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : _____

Name : MD RIZAL BIN OTHMAN

Date : 20 APRIL 2006

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLE

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In today's world the term digital has become part of our everyday vocabulary because of the dramatic way that digital circuit and digital techniques have become so widely used in almost all areas of life: computers, automation ,robots, medical science and technology, transportation, entertainment, space exploration, and on and on. One of the most common operation that occur in any digital system is the transmission of data from one place to another .The data can be transmitted over a distance as small as a fraction of an inch on the same circuit board, or over distance of many miles. The data that is transmitted is in binary form and generally represented as voltage at the outputs of a sending circuit that are connected to the inputs of receiving circuit. There are two basic methods for digital data transmission: parallel and serial. By far the most common use of flip-flops is for the storage of data or information. These data are generally stored in group of FFs called registers. The transfer operations are performed in synchronous or asynchronous transfer Synchronous serial transmission requires that the sender and receiver share a clock with one another, or that the sender provide a strobe or other timing signal so that the receiver knows when to "read" the next bit of the data. Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver.

In this project, the application of the FPGA is use as the programmable gate logics where will be implemented as the program to communicate with the MAX233 and the PC for serial data comparator transferring.

This project involved the design, compile, implementation and simulation of a system using the MAX+pluss II. In this project application, it will working as the serial data comparator to compare the ID code and ID number. This project processed in digital form that is the data bit generated is in digital. It will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code .Each bit of the ID number will stored in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output. This will produces the LED and the buzzer on. at the FPGA board.

The transmitted serial bit data in baud rate 96000act as the input to FPGA. Baud is a measurement of transmission speed in asynchronous communication .So this project can used many applications such as in security entry system. This system allows us to deactivate the alarm so that we can enter the building. If the correct digit has been entered on the keypad, the 8 bits on the inputs of the comparator are same as the 8 bits on the keypad and the comparator produces a HIGH on the output.

1.2 OBJECTIVES

The objective of this project is to develop a asynchronous serial data comparator using field programmable that implemented on FPGA. The FPGA, acts as an external hardware is used as a medium for communication between the PC with DB9 and the external MAX233 that will be connected directly with the FPGA. The logic circuit will be created by following the stages in the ***MAX+plusII*** software The other software will be developed is in Visual Basic for the serial port communication between the FPGA and PC. To accomplish such objective, both of the software must work well and communicate with each other in order for the transfer of reading and writing of data file to be successful.

Other than the objectives above, this project is to show how the FPGA works and how it can be a great tool for a microcontroller board in integrated circuit applications. Understanding its connections and the downloading process also will be learnt from this project. After all the downloading in the ALTERA chip and testing are being done, its intended function to communicate with the PC.

1.3 SCOPE

The project is expected to determine the ID code number using asynchronous data and will recognizes the 8 bits and it represent as the data bits that determine the ID number and compare the code using Max+pluss. Each bit of the ID number will store in the different flip-flop of 8 shift register. If each flip-flop output matches the ID number, the circuitry produces a high output to the circuitry.

1.4 PROBLEM STATEMENT

The idea of problem is based on comparison in operation in the system of compare ID code for example in ID number invalid or valid. By using FPGA, the ID number will be recognizes the ID code and compare it in the circuitry then produces a high output .This problem make to develop of asynchronous data comparator using FPGA to transmit the data. At this stage, objective and project scope clearly clarify to solve the problem.

CHAPTER II

LITERATURE REVIEW

The literature review is the second step to overview and study about the program flow by using the block diagram. The study done briefly to understand the implementation on FPGA using a programmable logic device to develop and to answer what, why, and how about the operation. The study used to next step of specification and design according from the source and the related project.

2.1 FPGA

FPGA stands for **F**ield-**P**rogrammable **G**ate **A**rray. It is a technology for making integrated circuits. The key difference from other chip technologies is that it is *field programmable*, meaning that you get to make the chip do what you want without having to get an external manufacturer. This means much less risk because if you make a mistake you do not have to wait for weeks to get another chip made, and you do not have to spend a lot of money to make the fix. All you do is modify your design and re-program your chip, much like you fix a C program, recompile it, load it into memory, and run it.

This capability provides lots of opportunities for doing interesting things, including just building chips full of logic. It makes a lot of the previous difficulties of building hardware easier. Some even call this *soft* hardware.

For example, you can now think of building compute engines that do their computations by configuring hardware solutions rather than executing a bunch of instructions.

(Stephen D. Brown, 1992)

2.2 PROGRAM FLOW

The first phase is development of the system block diagram of asynchronous serial data comparator using FPGA. The figure shown below is giving a draft idea on the flow of system. Further and detail explanations will be given in this section.

The data send from the PC is in the serial data with 8 bit data binary number. The connection using the DB9 as the serial communication part with the FPGA board through MAX 233 as the medium to transmit the data using the 96000 baud rate.

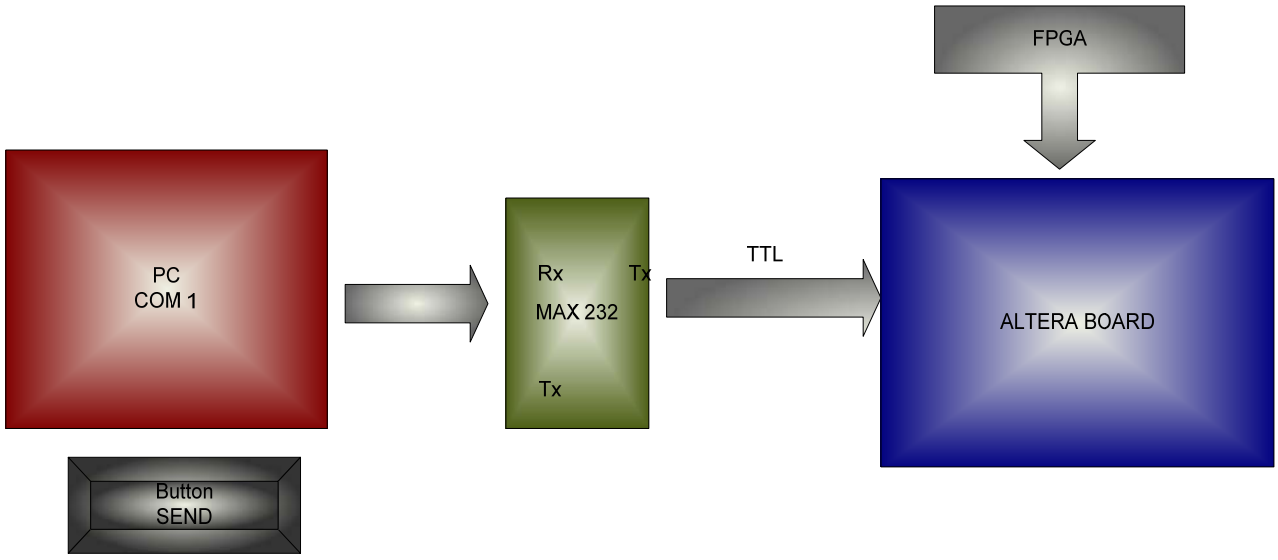


Figure 2.1: Block Diagram of Asynchronous Serial Data Comparator

FPGA

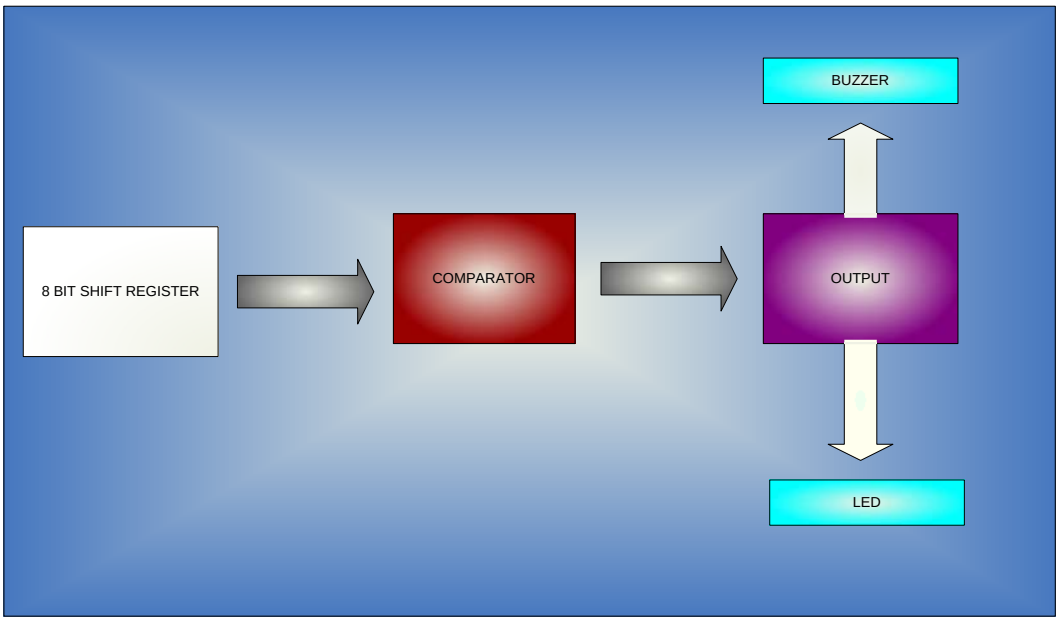


Figure 2.2: Block Diagram of Design Flow

2.2.1 Shift Register:

A shift register is a storage circuit where data is input serially from one end or in parallel and emerges from the other end after a specified number of clock cycles. In digital logic circuit, a one-bit shift register can be built using J-K flip-flop or D flip-flop. “D” is the input data, clock is the enable signal of J-K or D flip-flop. After one clock cycle, the value of D is shifted out as output Q. Other signals can be used to enhance basic operation of the shift register. They are “reset” and “preset” signal and “p” as presetted value. Using this 8-bit shift register is a convenient way to map a serial stream of signals to parallel output

2.2.2 Comparing Circuitry:

The purpose of this circuitry is to determine the ID code that has been transmitted and to compare this code to the ID number . Each bit of the ID number is stored in a different flip-flop of an eight-bit shift register in order to decode and determine the ID number. The output of each flip-flop will be compared to the ID number for each flipflop. If each flip-flop output matches the preset number, the circuitry produces a high output toup the tone generating circuitry. If the ID codes do not match, the output of the decoding logic remains low.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION OF FPGA

With the introduction of the **Field Programmable Gate Array** (‘FPGA’ - a configurable- logic chip) in the early 80’s, the hardware engineer was empowered to implement chip-level designs in silicon without having to fabricate a chip. As these devices and their software tools matured, the use of FPGAs expanded from testing and verifying digital designs to in-system use. This overview describes the fundamentals as well as current uses of this technology.

FPGAs perform the function of a custom LSI circuit, like a gate array, and are user programmable. The most significant advantage of using FPGA devices is the ability to produce a prototype logic design, implementing it in silicon within hours, while conventional gate array devices can take months and many dollars to develop and produce working silicon. Since their introduction, FPGAs have continued to increase in useable gate count, while decreasing in price. They are currently being used as glue logic, for test / verification logic in system designs, for adaptable system designs and more recently as co processing devices. FPGAs are also used to emulate other component architectures, and are applicable for rapid prototyping. With the next generation of SRAM based FPGAs (designed with computing in mind) a whole new generation of computing applications will result.

A field-programmable gate array or FPGA is a semiconductor device containing programmable logic components and programmable interconnects. The programmable logic components can be programmed to duplicate the functionality of basic logic gates (such as AND, OR, XOR, NOT) or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these programmable logic components (or logic blocks, in FPGA parlance) also include memory elements, which may be simple flip-flops or more complete blocks of memories.

A hierarchy of programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, somewhat like a one-chip programmable breadboard. These logic blocks and interconnects can be programmed after the manufacturing process by the customer/designer (hence the term "field-programmable") so that the FPGA can perform whatever logical function is needed.

FPGAs are generally slower than their application-specific integrated circuit (ASIC) counterparts, can't handle as complex a design, and draw more power. However, they have several advantages such as a shorter time to market, ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. Vendors may offer less flexible versions of their FPGAs that are cheaper. The development of these designs is made on regular FPGAs and then migrated into a fixed version that more resembles an ASIC due to lack of ability to modify the design once it is committed. Another alternative is complex programmable logic devices CPLD.

3.2 APPLICATIONS

Applications of FPGAs include DSP, software-defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, and a growing range of other areas. FPGAs originally began as competitors to CPLDs and competed in a similar space, that of glue logic for PCBs. As their size, capabilities and speed increased they began to take over larger and larger functions to the state where they are now marketed as competitors for full systems on chips. They now find applications in any area or algorithm that can make use of the massive parallelism offered by their architecture.

3.3 OVERVIEW FPGA CONSTRUCTION

There are four main categories of FPGAs currently commercially available: symmetrical array, row-based, hierarchical PLD, and sea-of-gates as shown in Figure 3.1.

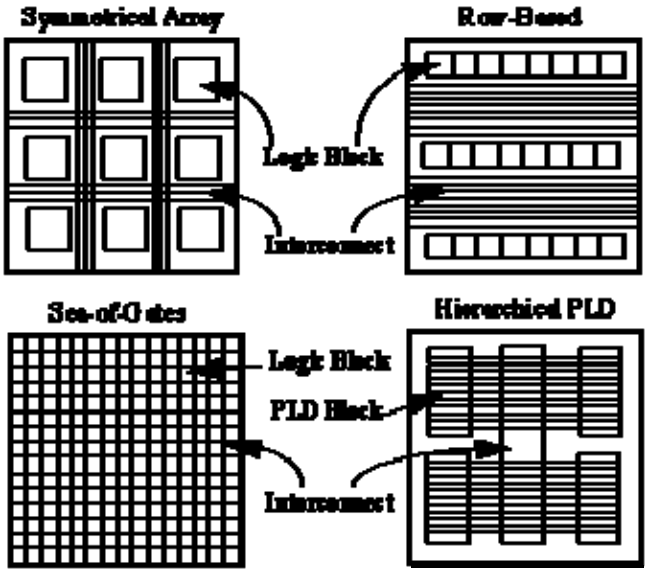


Figure 3.1: Classes of FPGA

In all of these FPGAs the interconnections and how they are programmed vary. Currently there are four technologies in use. They are: static RAM cells, anti-fuse, EPROM transistors, and EEPROM transistors.

3.3.1 Static RAM Technology -- In the Static RAM FPGA programmable connections are made using pass=transistors, transmission gates, or multiplexers that are controlled by SRAM cells. The advantage of this technology is that it allows fast in-circuit reconfiguration. The major disadvantage is the size of the chip required by the RAM technology.

3.3.2 Anti-Fuse Technology -- An anti-fuse resides in a high-impedance state; and can be programmed into low impedance or "fused" state. A less expensive than the RAM technology, this device is a program once device.

3.3.3 EPROM / EEPROM Technology -- This method is the same as used in the EPROM memories. One advantage of this technology is that it can be reprogrammed without external storage of configuration; though the EPROM transistors cannot be re-programmed in-circuit. The following table shows some of the characteristics of the above programming technologies.

3.4 THE FPGA

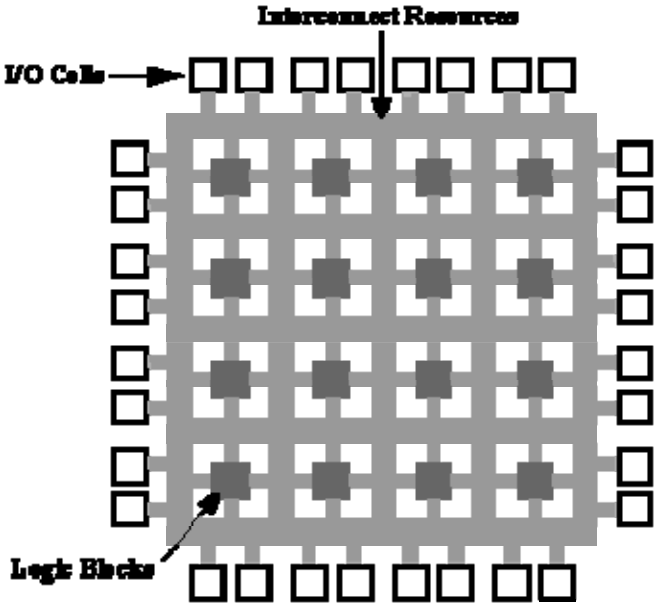


Figure 3.2: The FPGA structure

The FPGA has three major configurable elements: configurable logic blocks (CLBs), input/output blocks, and interconnects. The CLBs provide the functional elements for constructing user's logic (Figure 3.2). The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and internal connections implemented in the FPGA.

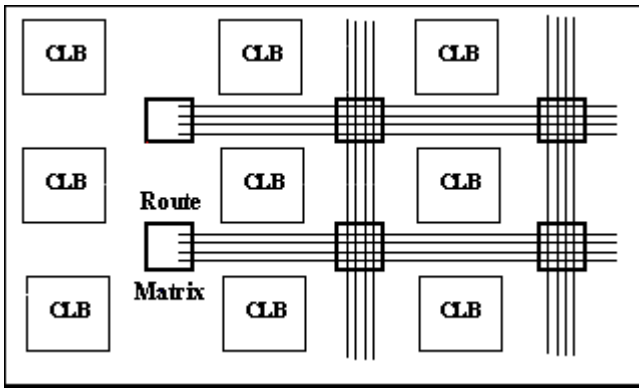


Figure 3.3: CLBs interconnect

Figure 3.3 depicts a FPGA with a two-dimensional array of logic blocks that can be interconnected by interconnect wires. All internal connections are composed of metal segments with programmable switching points to implement the desired routing. An abundance of different routing resources is provided to acheive efficient automated routing. There are four main types of interconnect, three are distinguished by the relative length of their segments: single-length lines, double-length lines and longlines. (NOTE: The number of routing channels shown in the figure are for illustration purposes only; the actual number of routing channels varies with the array size.) In addition, eight global buffers drive fast, low-skew nets most often used for clocks or global control signals.

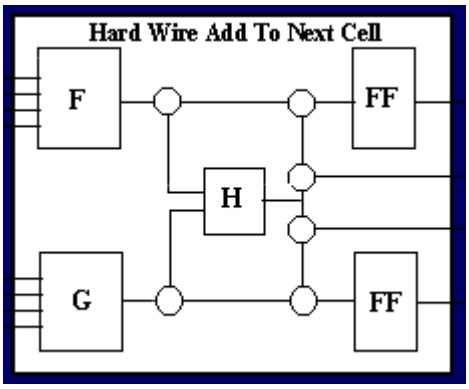


Figure 3.4: Configurable Logic Blocks

The principle CLB (Configurable Logic Block) elements are shown in Figure 3.4. Each CLB contains a pair of flip-flops and two independent 4-input function generators. These function generators have a good deal of flexabilty as most combinatorial logic functions need less than four inputs. Configurable Logic Blocks implement most of the logic in an FPGA. The flexabilty and symmetry of the CLB architecture facilitates the placement and routing of a given application.

The FPGA provide the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array. The key difference from other chip technologies is that it is *field programmable*, meaning that it can make like a chip do what design circuit without having to get an external manufacturer. This means much less risk because if have make a mistake it will do not have to wait for weeks to get another chip made, and do not have to spend a lot of money to make the fix. All is to do is modify the design and re-program the chip, much like fix a C program, recompile it, load it into memory, and run it. This capability provides lots of opportunities for doing interesting things, including just building chips full of logic. It makes a lot of the previous difficulties of building hardware easier. Some even call this *soft* hardware. The FPGA used in this project is for the control in storing and accessing the data in the remote unit. The FPGA acts as an internal hardware and connects with the microcontroller of the base unit to store the data. The program developed is processed in digital form that is the bit stream generated is in digital. Furthermore the process in the digital is chosen because it contributes to this main advantage, which is sharpness.

3.4.1 ADVANTAGE IN FPGA COMPARED TO COMON TTL

i. Save time because FPGA only needs to use software without the need of physical connection for combining the gates that are want to be used.

ii. Errors in connection are easily detected without the physical connection.

iii. Lower costs because various gates can be implemented into ***ALTERA MAX+plusII***

iv. New circuit connection will not interfere with the section of the circuit that has already been done. Therefore, combining sections of circuits is easier.

3.4.1 BENEFIT OF FPGA AND MAX+plusII IN DIGITAL CIRCUIT DESIGN.

i. Save time because every design can be simulated and its output can be determined just by software and errors can be corrected fast.

ii. MAX+plus software makes circuit designs more convenient with its mixtures of design software and with its library features complete with thousands of basic components.

iii. Output results can be determined by simulation. This will save cost because output results will be known without ever using the intended hardware.

iv. Simpler implementation, fast and efficient because of all the gates needed for most of the designs of digital system are consists in single software.

v. Maintenance costs for one unit of FPGA is much lower because the gate usage and the space size for a system can be reduced.

vi. Low risk software because the changes in design will not include the changes of components physically.

vii. Designs that are newly developed by the FPGA can be marketed fast because of the limited time in circuit design process.

viii. Advantages in reprogramming the FPGA that allows different logics to be implemented into a design by the same FPGA at different time.

3.5 GETTING COMPONENT

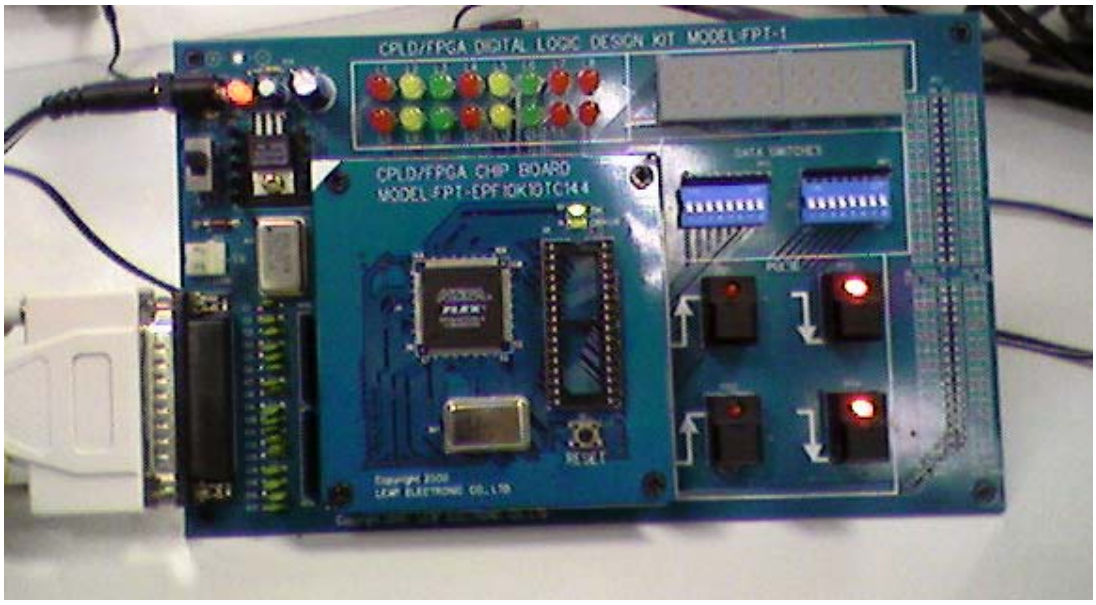
Electronic components are quite expensive to be afforded by a student. Fortunately, the faculty could supply most of the components to project student. In other words, faculty financially supports students.

There are some procedures that we have to go through to get the component from faculty. First, students should have to get the request form component. The, the student could fill in the form according to the components list. After this, students need to get the supervisor's approving signature. Then, students could proceed to get the components.

3.6 HARDWARE IMPLEMENTATION

3.6.1 INTRODUCTION ON FPGA MODEL:FPT-EPF 1010TC144 BOARD

In this project, the CPLD/FPGA CHIP board MODEL:FPT-EPF 1010TC144 was used as the base for the FPGA to work on. To view the circuit board figures shown below.. The board was developed by LEAP ELECTRONIC CO. LTD.It is one of the newest generations of FPGA board available in the market today. What makes this board stand out from the crowds is that is has a wide selection of expansion modules. The expansion module includes items such as LEDs, Switches, 7-segment displays, peripheral connectors and compact flash. It is also regarded as a fast and economical prototyping of professional designs and it is ideal for advanced-student projects. All of the features stated above clearly show why this board is a great tool for this project.



- **Figure 3.5 : CPLD/FPGA CHIP BOARD**
- **MODEL FPT-EPF10K10TC144**

Here hardware specification:

Support Devices:

- (1)ALTERA EPF10K10TC144 TQFP144pin
- FPT-EPF10K10TC144
- 8x2 LED shown output.
- 8x2 Logical input toggle.
- Four pulse keystrokes producer (two positive
- pulses two negative pulses) .
- Six digits and seven nodes monitor.
- Own Red main power guiding lights.
- Within 10MHz oscillator.
- Own main power switch to exchange Adaptor
- with Extend Power Pin.
- 25pin D Type Connector (Printer Port Download
- FPGA .

- Use DC 9V Adaptor or Extend Power Pin provided
- for user. Specification :DC 5V.
- Support ALTERA MAX +Plus II Baseline

Here the listed features:

- Use CPLD/FPGA software and hardware to learn
- new logical IC design, in order to replace TTL/
- CMOS complicated hardware design.
- Use Graphic and VHDL ABEL AHDL to develop circuits.
- Use Print Port Download directly under original
- manufacturer development system.
- Programmed finished file to EPROM (FLASH) and
- operated it independently.

3.6.2 **ALTERA** 'Flex EPF10K10'

Now the discussion is about the software to design the circuit that simulate on the programmable logic device. Software is the programs that run on programmable hardware and change the operation depending on the input.

Altera is one of the pioneers of Programmable Logic, following notable early leaders Signetics and MMI in introducing PLDs. Altera develops many features that is gearing towards system-on-a-programmable-chip (SOPC) capability. Some of the more recent examples include embedded memory, embedded processors, and high-speed transceivers. Altera is famous for its operational excellence. The success in 130nm and 90nm product launches are good case studies. Altera's Nios and Nios II soft core processors and HardCopy II & HardCopy devices are extending Altera's reach in addressable markets, and put Altera in the world of embedded processors and structured ASICs respectively. Among its chief competitors are Xilinx, Lattice Semiconductor, Actel, Quicklogic and Atmel.

Although their software suite extensively supports VHDL and Verilog as primary languages, Altera is the developer of the Hardware Description Language known as AHDL.

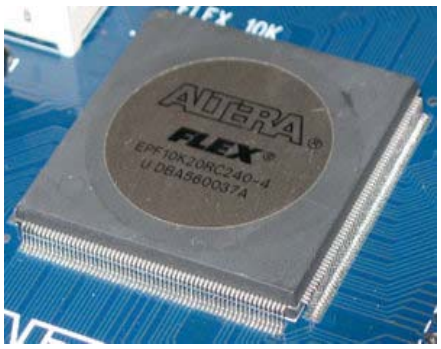


Figure 3.6: Flex EPF10K10'

3.7 SOFTWARE IMPLEMENTATION

3.7.1 INTRODUCTION TO MAX+PLUS II

The MAX+PLUS® II development software provides a complete design environment that easily adapts to specific design needs. Regardless of whether use a personal computer or a workstation, MAX+PLUS II ensures easy design entry, fast processing, and straightforward device programming.

MAX+PLUS II software is a fully integrated, architecture-independent package for designing logic with Altera programmable logic devices, including Classic™, ACEX 1K, MAX 3000, MAX® 5000, MAX 7000, MAX 9000, FLEX® 6000, FLEX 8000, and FLEX 10K devices. (MAX+PLUS II also allows to program FLASHlogic™ and APEX devices.) MAX+PLUS II offers a full spectrum of logic design capabilities: three design entry methods for hierarchical designs; floorplan editing; powerful logic synthesis; design partitioning; functional, timing, and board-level-type linked simulation; detailed timing analysis; automatic error location; and device programming and verification. MAX+PLUS II also reads standard EDIF netlist files, VHDL netlist files, Verilog HDL netlist files, and OrCAD Schematic Files, and writes EDIF, VHDL, and Verilog HDL netlist files, including VITAL-compliant files, for a convenient interface to other industry-standard CAE software.

We can integrate existing designs created with Altera's A+PLUS™, SAM+PLUS™, and MAX+PLUS (DOS) software packages into MAX+PLUS II designs. In addition, MAX+PLUS II for UNIX workstations allows to run the Synopsys Design Compiler and FPGA Compiler automatically, which allow to process both VHDL and Verilog HDL designs. The MAX+PLUS II Compiler ensures that a design--called a project in MAX+PLUS II--fits into the device architecture in the most efficient way possible.

MAX+PLUS II offers a rich graphical user interface complemented with an illustrated, easy-to-use on-line help system. The complete MAX+PLUS II system includes ten fully integrated applications that take through every step from design entry to device programming.

Many features and commands are shared by the different MAX+PLUS II applications, so that learning to use one application gives a head start on learning to use the others. For example, to use identical commands in each MAX+PLUS II application to open files, to assign project devices, and to begin compiling the current project. The design editors in MAX+PLUS II--the Graphic, Text, and Waveform Editors--and the auxiliary editors--the Floorplan and Symbol Editors--also share numerous design entry tools and features. Each editor allows to perform similar tasks, such as assigning a pin, in the same way.

It can easily combine different types of design files in a hierarchical project, choosing the design entry format that works best for each functional block. Over 300 primitives, megafunctions, and macrofunctions, AHDL™, and the built-in EDIF, VHDL, Verilog HDL, and OrCAD interfaces simplify design entry. Architecture-independent design entry gives the freedom to create logic without worrying about the final device implementation.

It can work with different MAX+PLUS II applications at the same time. For example, can open multiple design files and transfer information between them, while simultaneously compiling or simulating another project. It can view an entire hierarchy of design files and move smoothly from one hierarchical level to another. As open a design file, MAX+PLUS II automatically starts the appropriate design editor.

The MAX+PLUS II Compiler lies at the heart of the MAX+PLUS II system, providing powerful design processing that can customize to achieve the best possible silicon implementation of the project. Automatic error location and extensive documentation on error and warning messages make design modifications as simple

as possible. It can create output files in a variety of formats for simulation, timing analysis, and device programming, including EDIF, Verilog HDL, and VHDL files for use with other industry-standard EDA tools. At every step in the design process, MAX+PLUS II software makes it easy for us to focus on our design--not on how to use the software.

The superb integration of MAX+PLUS II software improves efficiency and productivity, putting in control of our logic design environment.

Implementing a logic design with an FPGA using Altera- Max+plus II software usually consists of the following steps.

o in this project to design the circuit of the remote unit, the MAX+plusII software can be used to implement all further development stages which are:

1. Design Entry- entering the circuit design into MAX+plusII.

2. Functional Compilation-compiling the circuit so that it may be simulated

3. Functional Simulation-simulating the circuit to ensure that it functions correctly.

4. Floorplan-fitting the circuit into map of the programmable device.

5. Design Compilation-compiling the circuit ready for downloading to the device.

6. Download-downloading the complied circuit to the device.

3.7.2 INTRODUCTION TO MAX+PLUS II

There are steps to create new design – design entry

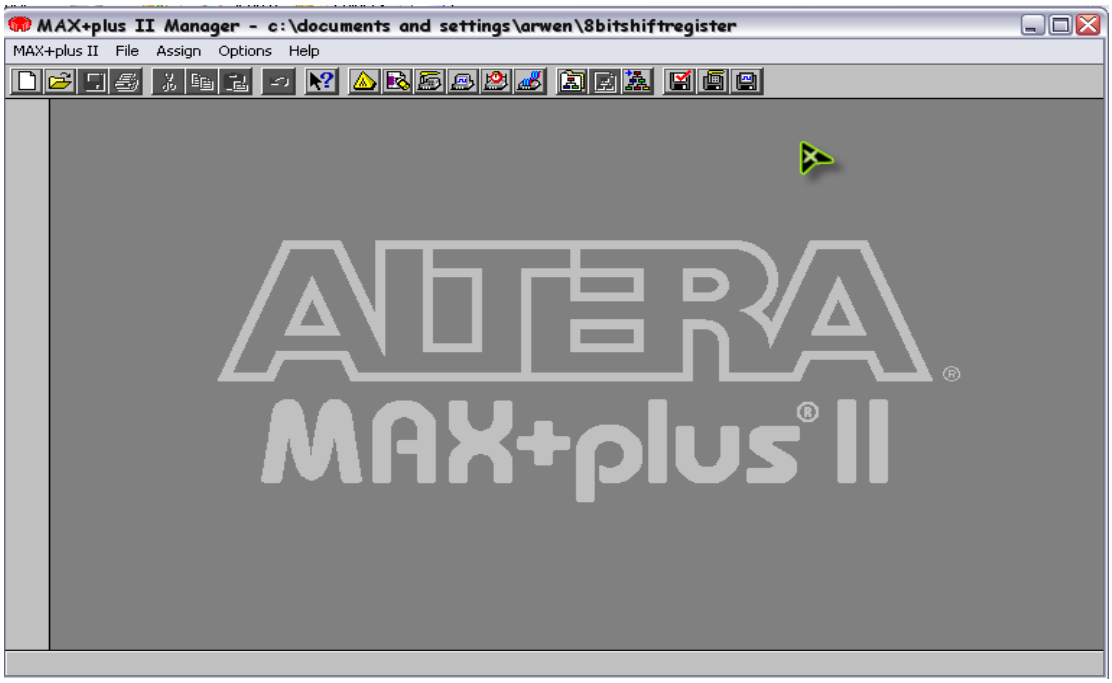


Figure 3.7: The MAX+PLUS II Start Up screen

Firstly choose the menu bar “File”,”New” and select the graphic editor file as depicted in figure 3.8.

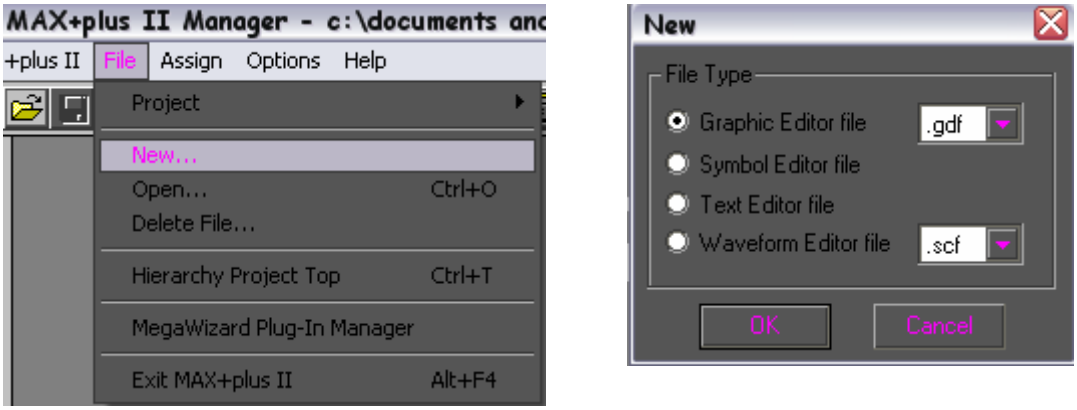


Figure 3.8 : opening new graphic editor file

Each of the above methods will result in a blank graphic editor windows being displayed as shown on figure 3.9

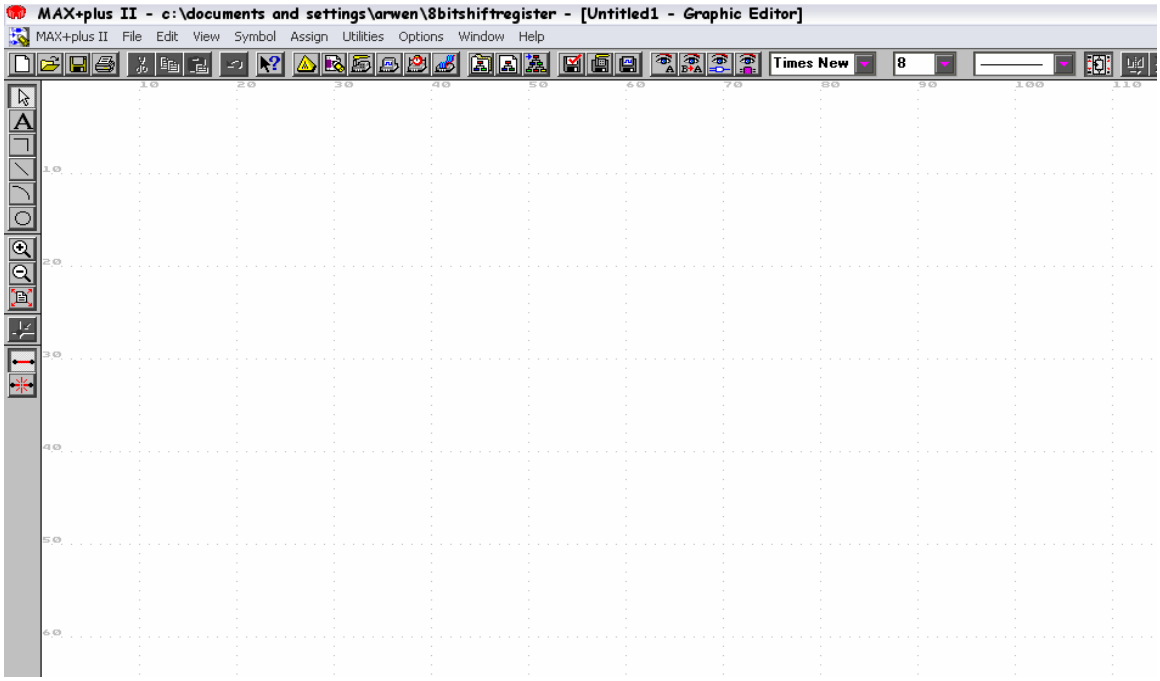
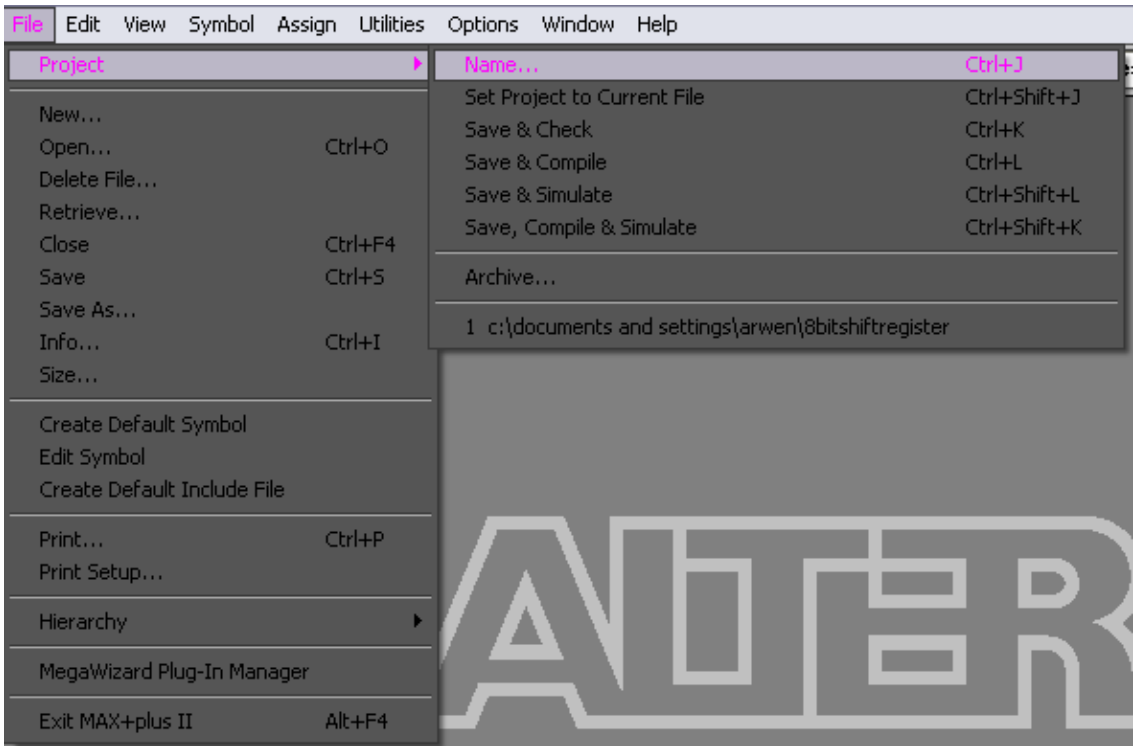


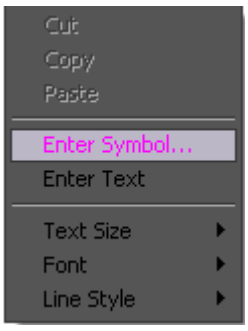
Figure 3.9: Blank editor windows

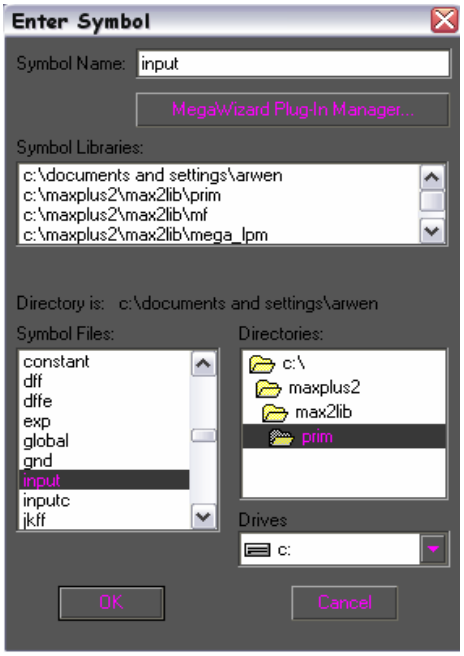
Designing circuit using Max+Plus II is project-oriented. For each circuit design, the project must be created and named. This project will then be used to group the files created within MAX+PLUSS II that are associated with that design. To named the project file name , must go to “File”, “Save As..” from the menu bar, and then entering a name for the file. So in the project name the “**8 bit shift register.gdf** ”

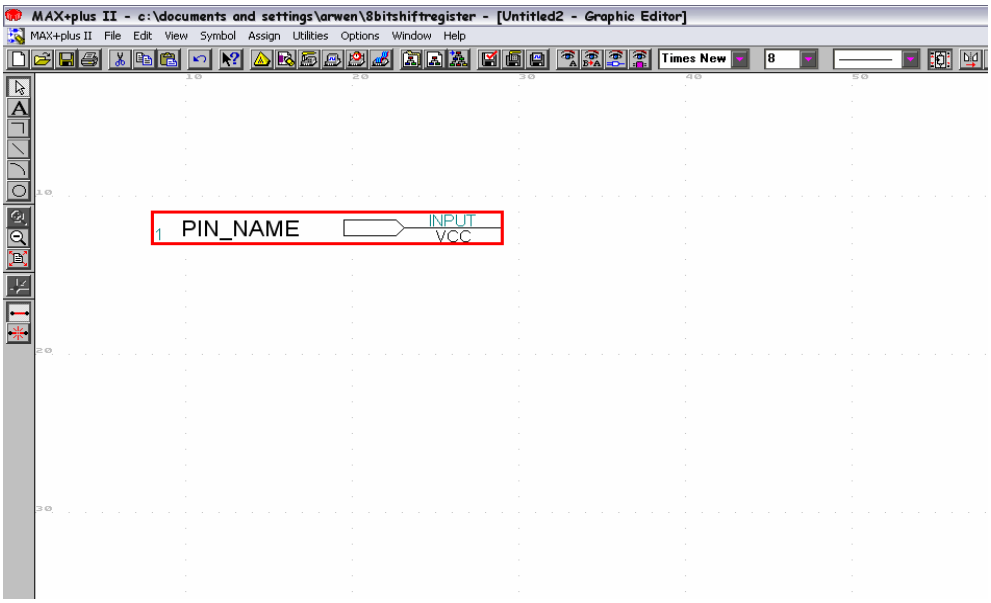
After the file is save, set the file to the current project file. To do this, choose the “File”, “Project”, “Set Project to Current File” from the menu bar.

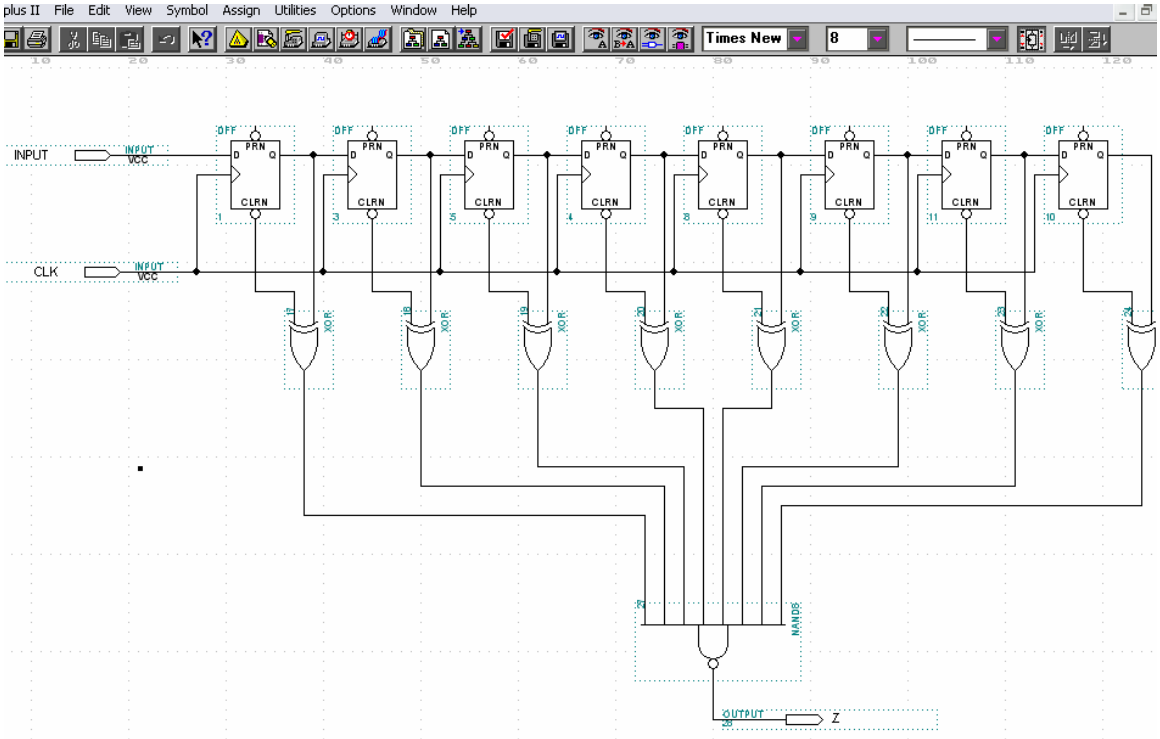


The components such as logic symbols can be placed on the graphic editor screen simply by double-click the left-hand mouse button on the point where insertion is required or right-click the mouse button and choose “Enter Symbol”. Enter the “input” at the Symbol Name and then click “OK”. The “input” symbol will appear at the graphic editor screen. So here the circuit can be design.



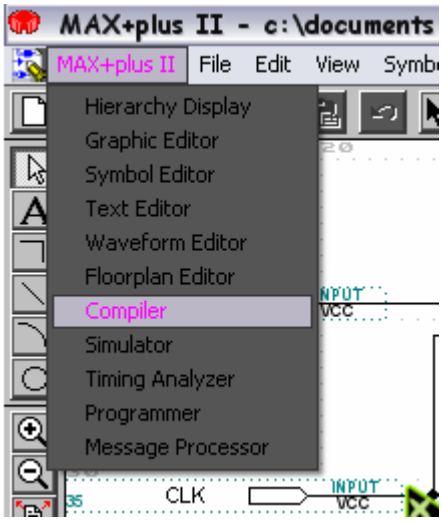



MAX+plus II - c:\documents and settings\arwen\8bitshiftregister - [Untitled2 - Graphic Editor]
MAX+plus II File Edit View Symbol Assign Utilities Options Window Help
Times New
8
PIN_NAME
INPUT
VCC

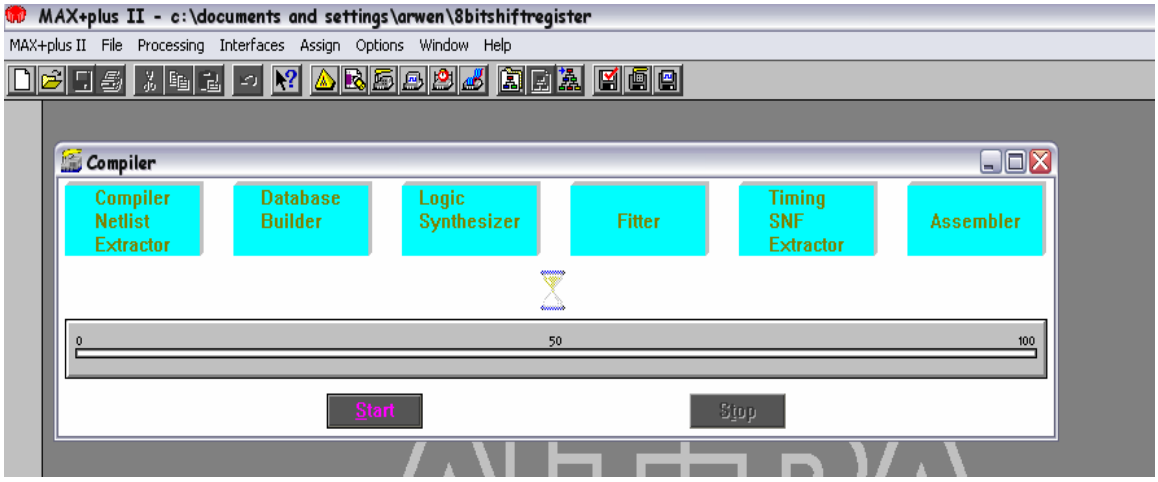

plus II File Edit View Symbol Assign Utilities Options Window Help
Times New
8
INPUT
CLK
INPUT
VCC
DFF
PRN
D
Q
CLRN
XOR
NAND6
OUTPUT
Z

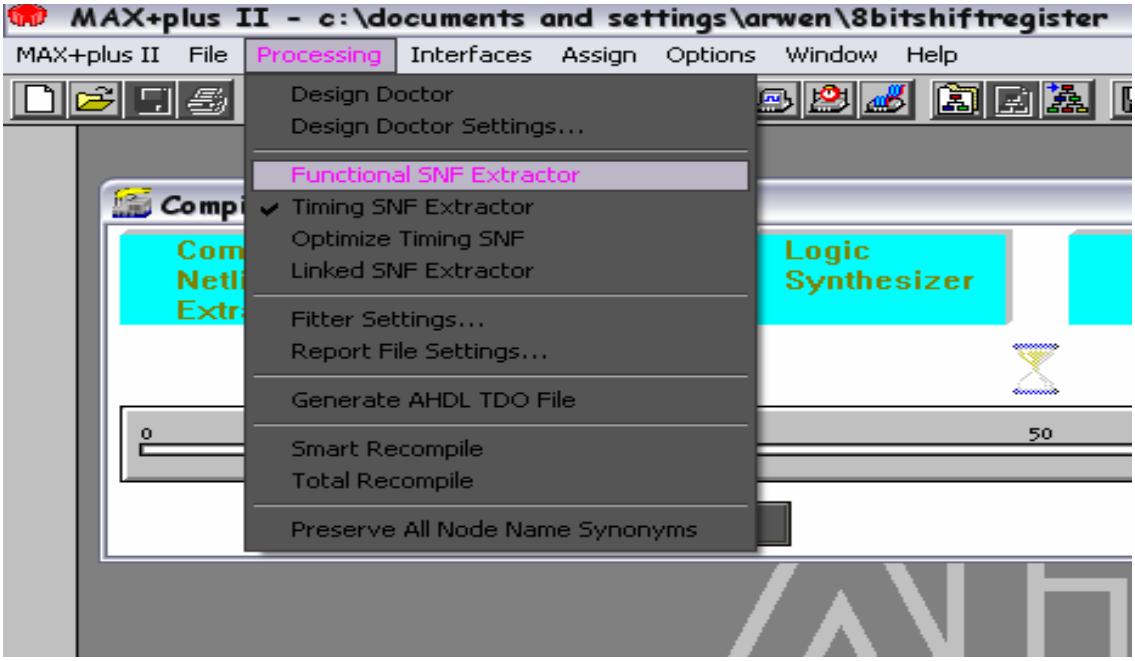
3.7.2.1 Functional Compilation

After connected all the connection,“save” the file. Then select the compiler by choosing the “MAX+PLUS II”, or “Compiler” from the menu bar, or clicking the compiler icon.



When the compiler window opens it will probably default to full compilation change the compiler setting at the new menu heading labeled “Processing” appeared on the menu bar. In order to configure the compiler for the function compilation , choose “Processing”, “Function SNF Extractor” from the menu bar.

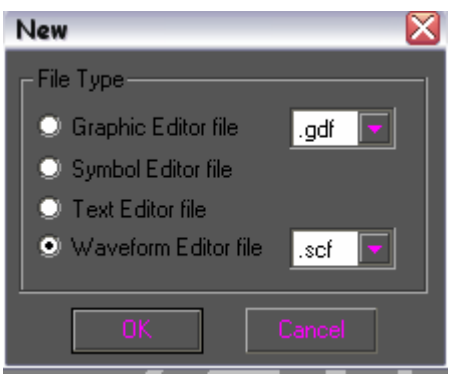


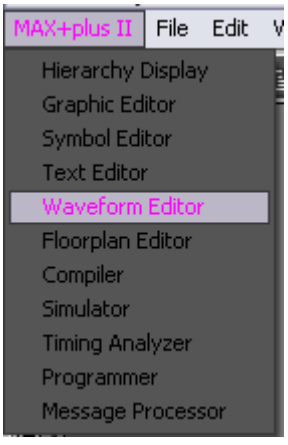


3.7.2.2 Functional Simulation

In order to simulate the compile design, it necessary to create what is known as a simulator channel file. The file is created using the waveform editor, and comprises waveform information that will apply to the input of the circuit. Later when run, the simulator will use the information to produce the output waveforms.

The waveform editor is open by clicking the new design file icon and selected a waveform editor file with a “scf” extension as a shown below. Alternatively, the waveform editor is start by choosing the “MAX+PLUS II”, “Waveform Editor” from the menu bar.





A blank waveform editor window shown in figure

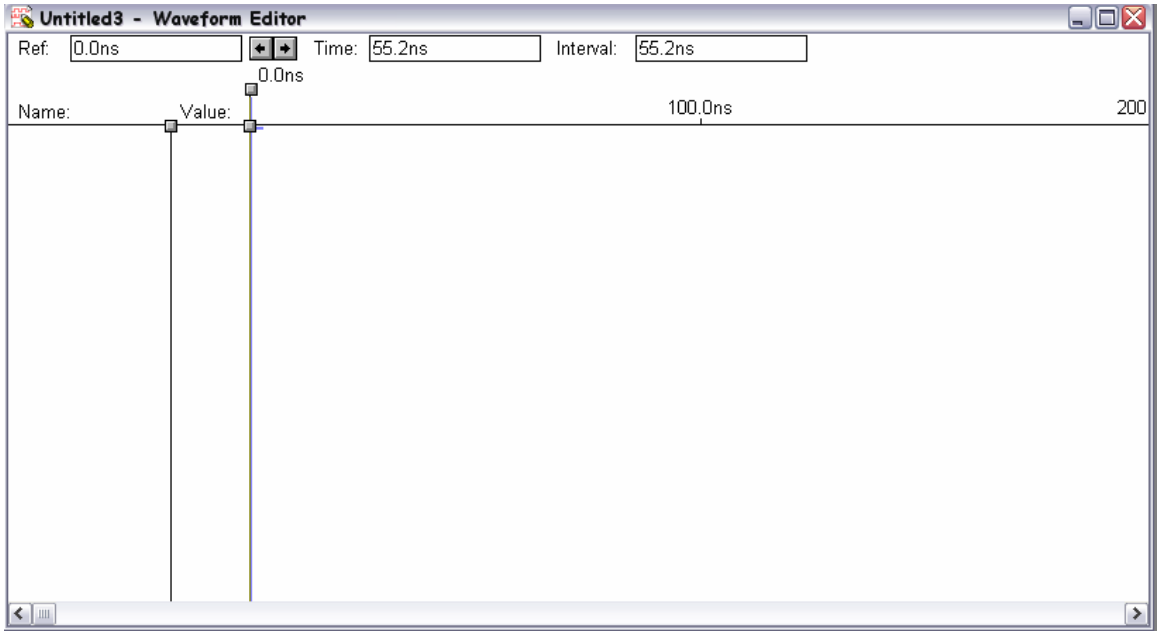
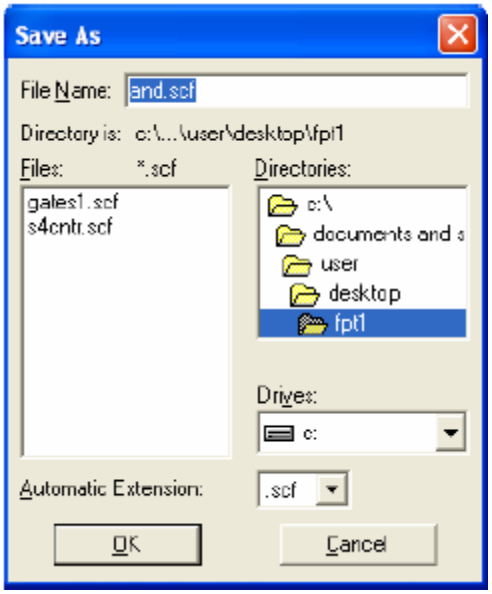


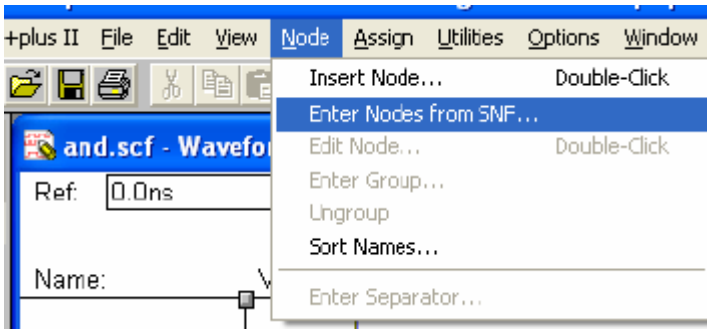
Figure 4.0: Blank waveform editor window

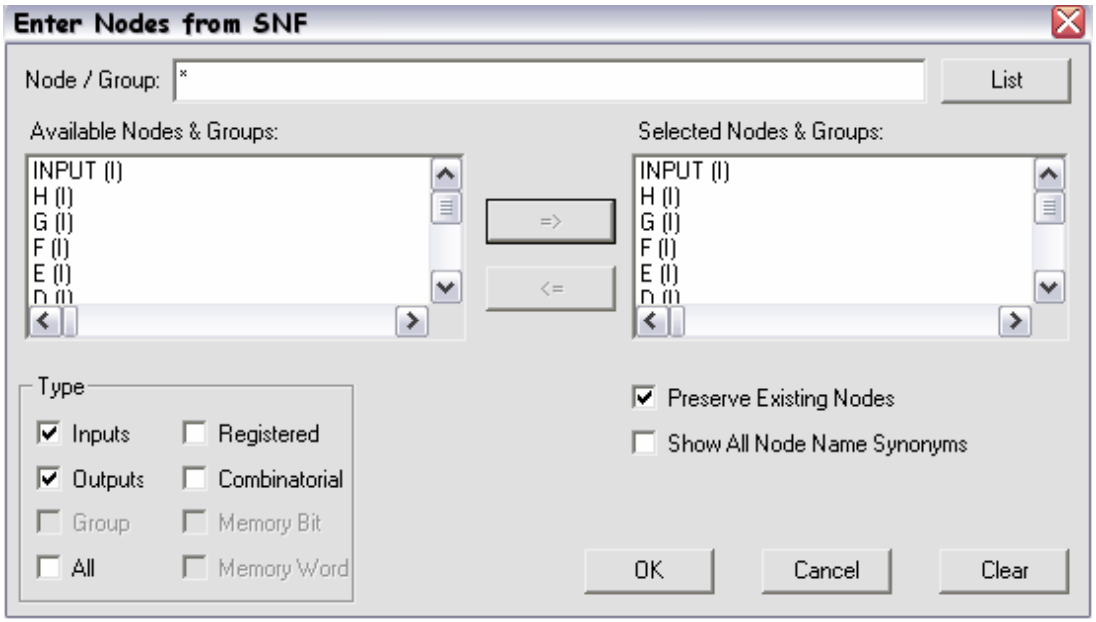
Name and save the new waveform editor file by either clicking the “save” file icon or choosing “File”, “Save as..” from the menu bar.



MAX+PLUS II allows any or all of the nodes to be added to the waveform diagram. To do this, choose “Node”, “Enter Nodes from SNF..” from the menu bar.

This will open a new window, keeping the default setting, click the “List” button in the top-right corner of this window.





The “Enter Nodes form SNF” window will then close, revealing the waveform diagram compete with input and output nodes.as illustrated by figure

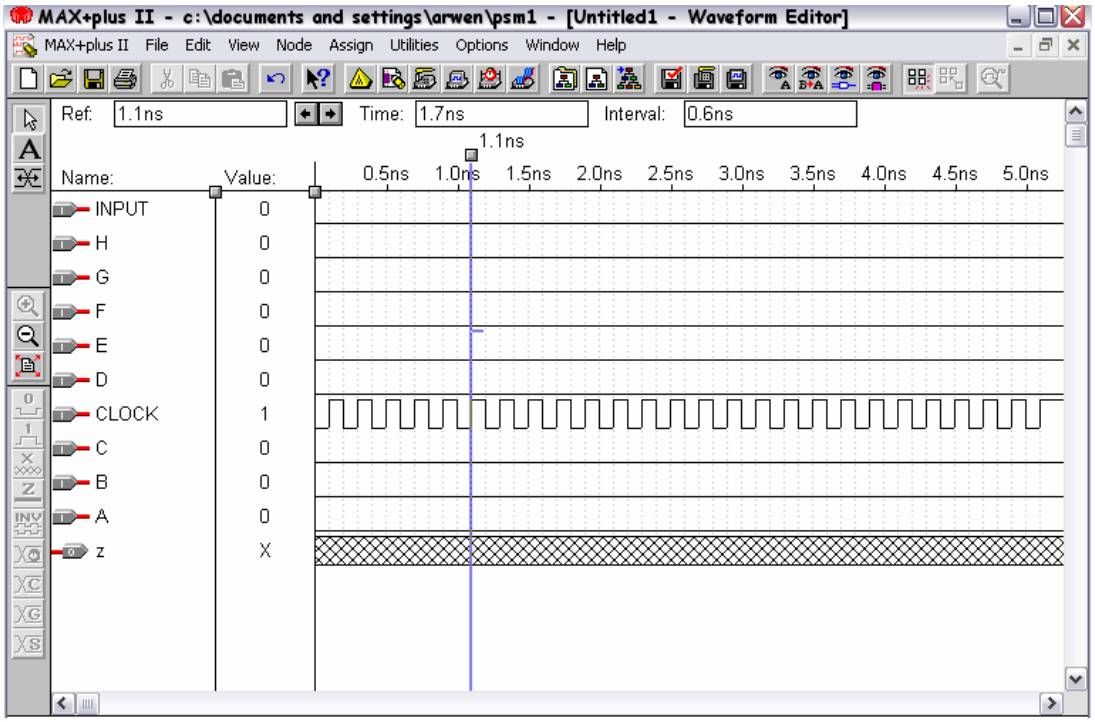
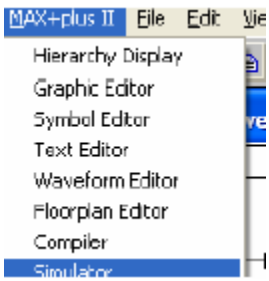
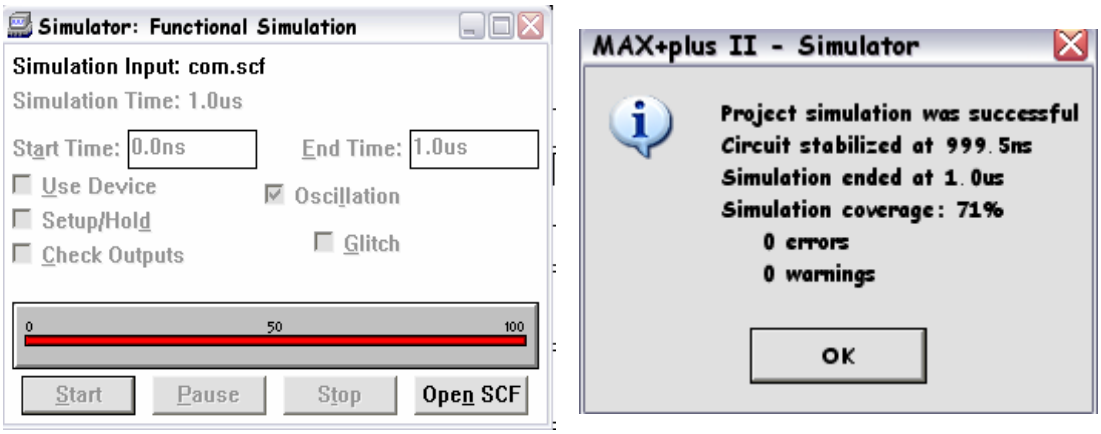


Figure 4.1 : The waveform diagram with input and output nodes added

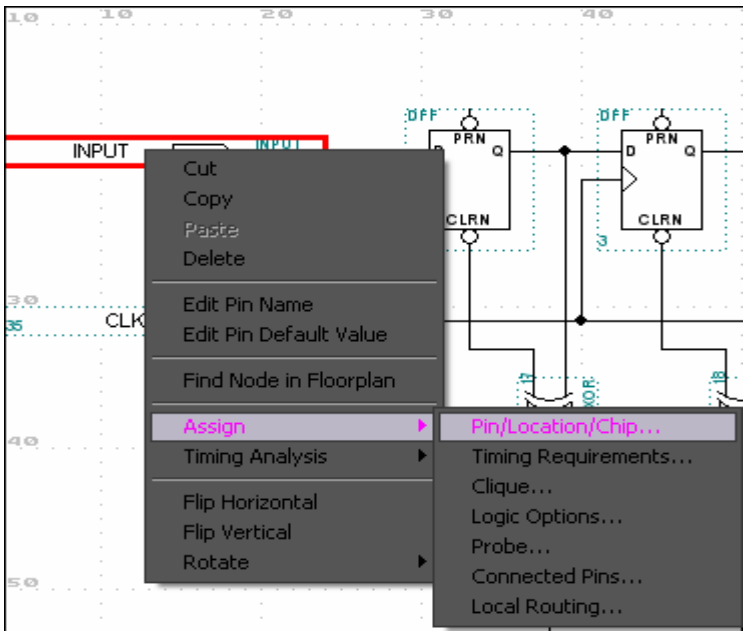
Start the simulator by choosing the “MAX+PLUS II”, “Simulator” from the menu bar, or by clicking the simulator icon.

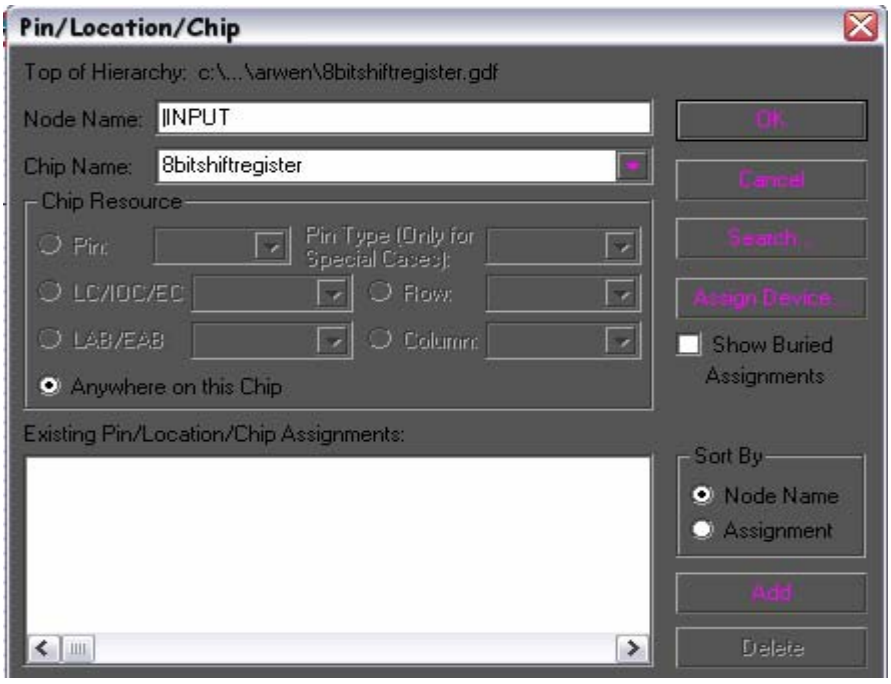




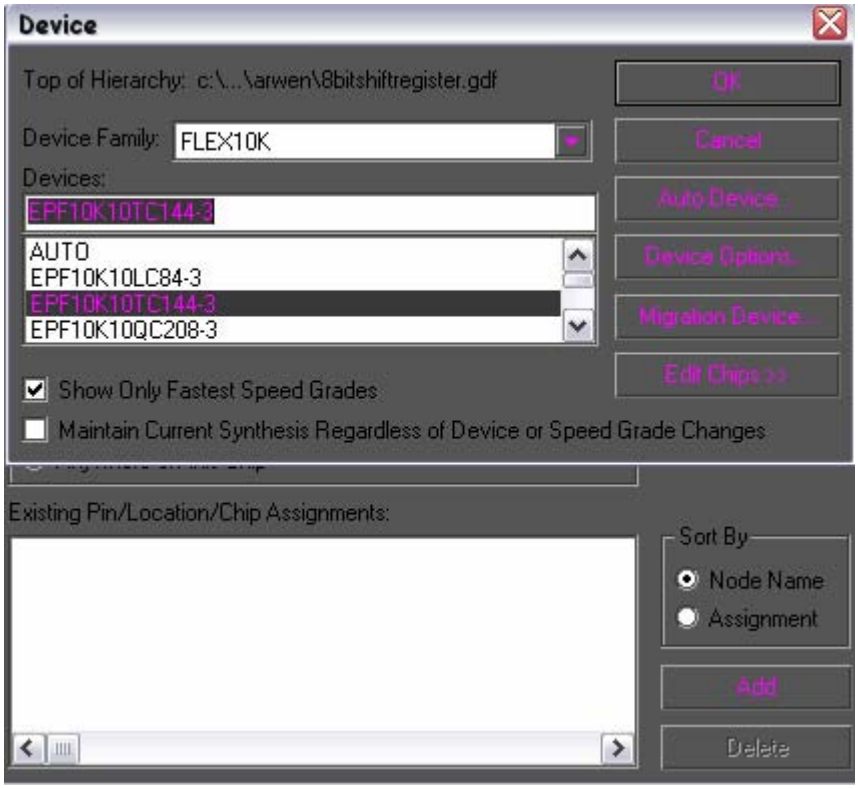
3.7.2.3 Assign Pin

To assign pin to each input and output port. Right-click the mouse button and select “Assign”, “Pin/Location/Chip..”.



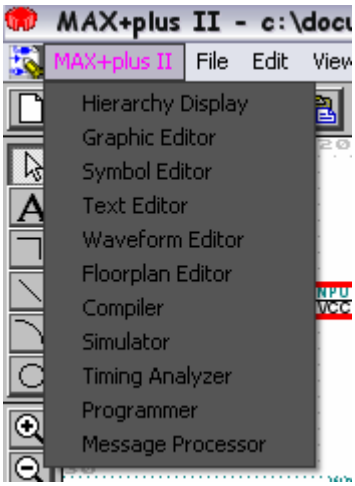


Then, choose “Assign Device”, make sure the device is set to “Flex 10K”, “EPF10K10TC144-4”

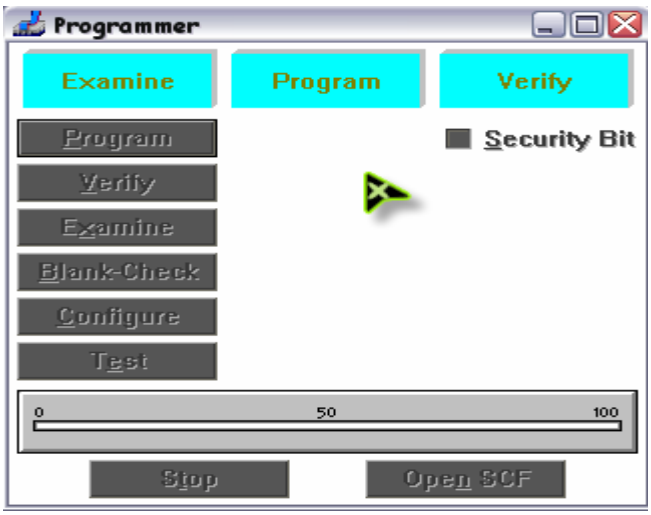


3.7.2.4 Download and Test

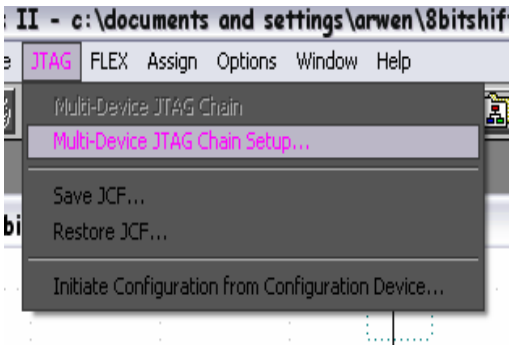
The final step is to download the compiled circuit to the EPF10K10TC144-4 programmable logic device in the FPGA.
Start the programmer by choosing the “MAX+PLSU II”, “Programmer” from the menu bar, or by clicking the programmer icon.



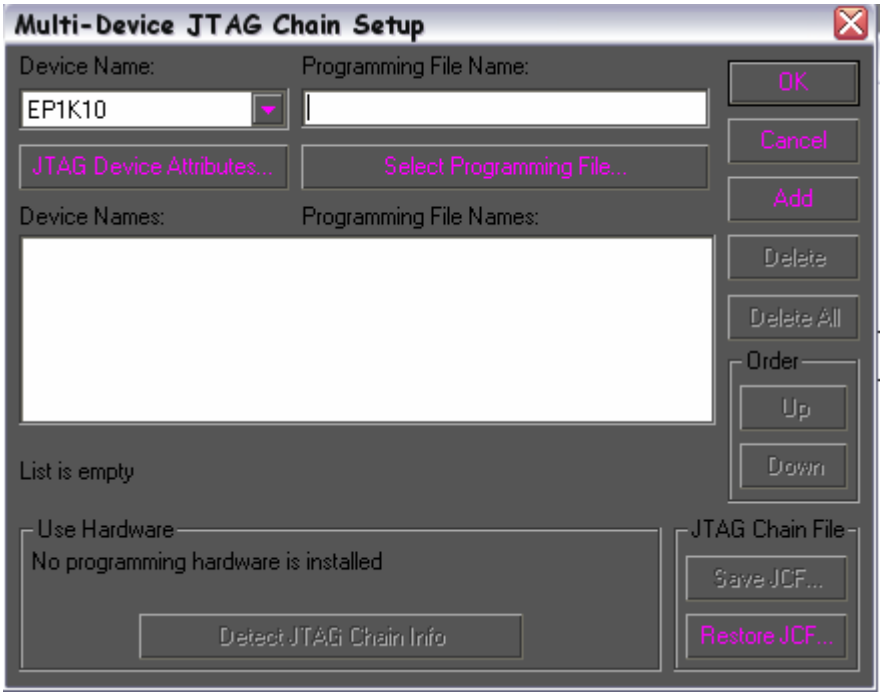
When the programmer window open, check the device is set to EPF10k10TC144-4 and that the file to be downloaded is “ filename.sof”. Click the Configure” button to start download.



To select the file for download, choose " JTAG", "Multi-Device JTAG Chain Setup.."



Choose "Select Programming File.." to select the file that want to download. The file must be with the ".sof" extension format. Then click the "Add" to add to the programmer list. Then click "OK".



3.7.3 ASYNCHRONOUS SERIAL COMPARATOR

Computer data can be sent from one device to another, carried by a wire cable without the use of a network interface card. This data can be sent using the serial port that sent each bits at a time over wires in the cable. Asynchronous serial devices can communicate using 7 or 8 data bits, and 1, 1½, or 2 stop bits. To further complicate matters, devices can also employ a parity bit instead of an eighth data bit to check for errors. Even parity systems transmit a one when the sum of the seven bits is an even number, while odd parity systems transmit a one when the sum is odd. Still more exotic systems may specify "mark" or "space" parity, where the parity bit is always a one or zero, respectively.

A PC DB9 port is an inexpensive and yet powerful platform for implementing projects dealing with the control of real world peripherals

To improve noise immunity and support long cable lengths, MAX-23 devices convert TTL/CMOS-level signals (0V=logic zero, +5V or +3.3V=logic one) to higher voltage bipolar signals. For the TD and RD signal lines, MAX-233 devices use a voltage between -3V and -25V to transmit a one and a voltage between +3V and +25V to transmit a zero.

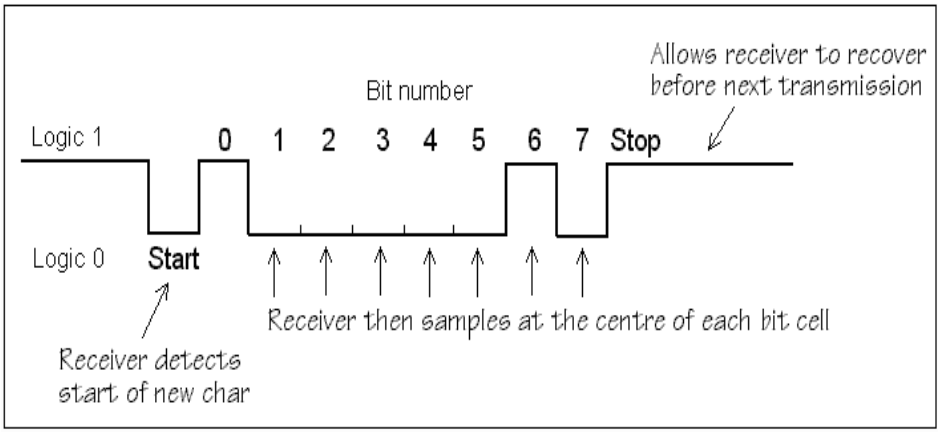


Figure 4.2 Bit Number Waveform

Figure 4.3: DB9

Signal Name		Pin Number		Direction
		25-pin	9-pin	
transmitted data	TD	2	3	DTE ▶ DCE
received data	RD	3	2	DTE ◀ DCE
request to send	RTS	4	7	DTE ▶ DCE
clear to send	CTS	5	8	DTE ◀ DCE
data terminal ready	DTR	20	4	DTE ▶ DCE
data set ready	DSR	6	6	DTE ◀ DCE
data carrier detect	DCD	8	1	DTE ◀ DCE
ring indicator	RI	22	9	DTE ◀ DCE
signal ground	GND	7	5	

Figure 4.4: The DB9 connector pinouts

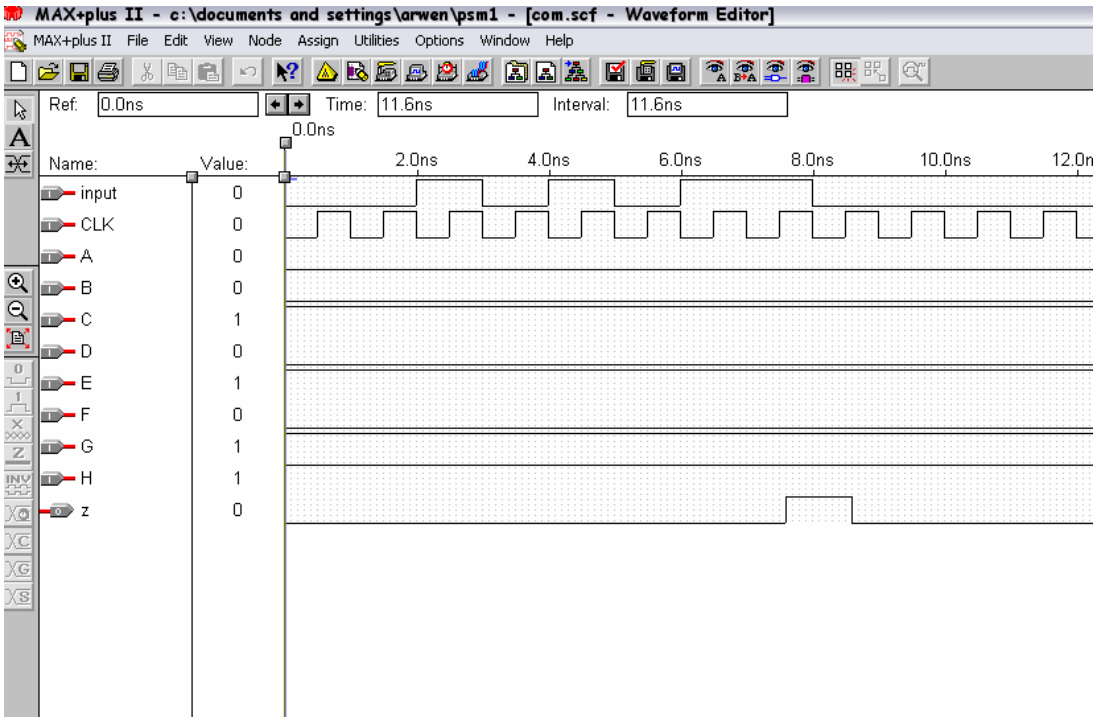
CHAPTER IV

RESULT

4.2 FPGA SIMULATION RESULT

Start process

Figure below shows the simulation result on the start of the FGPA of the 8 bit shift register and the comparator circuit. To start, asynchronous reset signal must be ‘1’. Input clock cycle of high (1) and low (0) is specified. The simulation of the circuit will produce high at the 8 time of bit meaning the data of ID code and ID number that send from the PC to the FPGA board is matches. This will produce LED on and the buzzer on.



CHAPTER 5

CONCLUSION

Referring to the objectives in chapter one, the purpose of this project is to design a program for FPGA as an external hardware to communicate with the MAX233 and PC by doing the simulation of the program of ALTERA to transfer the data. Besides that the communication between DB 9 and the FPGA by sending in asynchronous serial data comparator directly communicate each others

This project also shows that the used of FPGA is a great tool for learning purposes. The vast application and usage of FPGA in the field of engineering is not to be missed and further exposure in this field will be an extra attributes in enhancing engineering knowledge.

Problems encountered

Hardware implementation is more time consuming than the software implementation. It is because in real time hardware implementation, all the noise, environment, machine and human factors must be considered. For example , to develop the board MAX233 and DB9 cnnection and the application of the FPGA board it is a sensitive part and need to be careful when doing and apply it.This will may the board will not working properly or may damage.Rather when being implemented in the softwarewhere the conditions are considered ideal

5.1 RECOMMENDATION

This project is simple part of the sending the serial data by using the FPGA .In the future , the futures of the asynchronous serial data comparator can be further enhanced by introducing the more advance system such as the appear of the name of the ID code and ID number by applying in system of security. Besides that, maybe try to send by using the infra red to transfer the data.

REFERENCES

i. Steve Yessa, Student,Bradley University Peoria, IL 61606,(2003) http://cegt201.bradley.edu/projects/proj2003/dataman/

ii. Key Ringer. www.keyringer.com

iii. www.latticesemiconductor.com,

iv. Sharper Image. *www.sharperimage.com*,

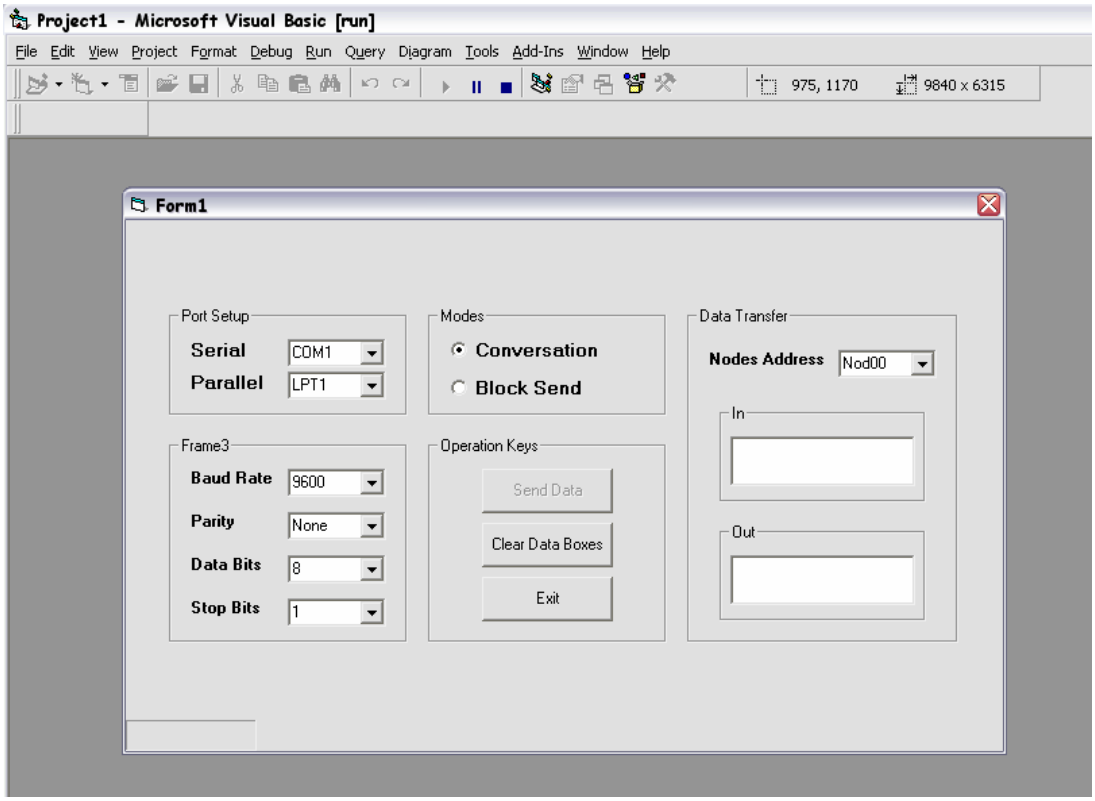
v. www.quatech.com

vi. www.farnell.com

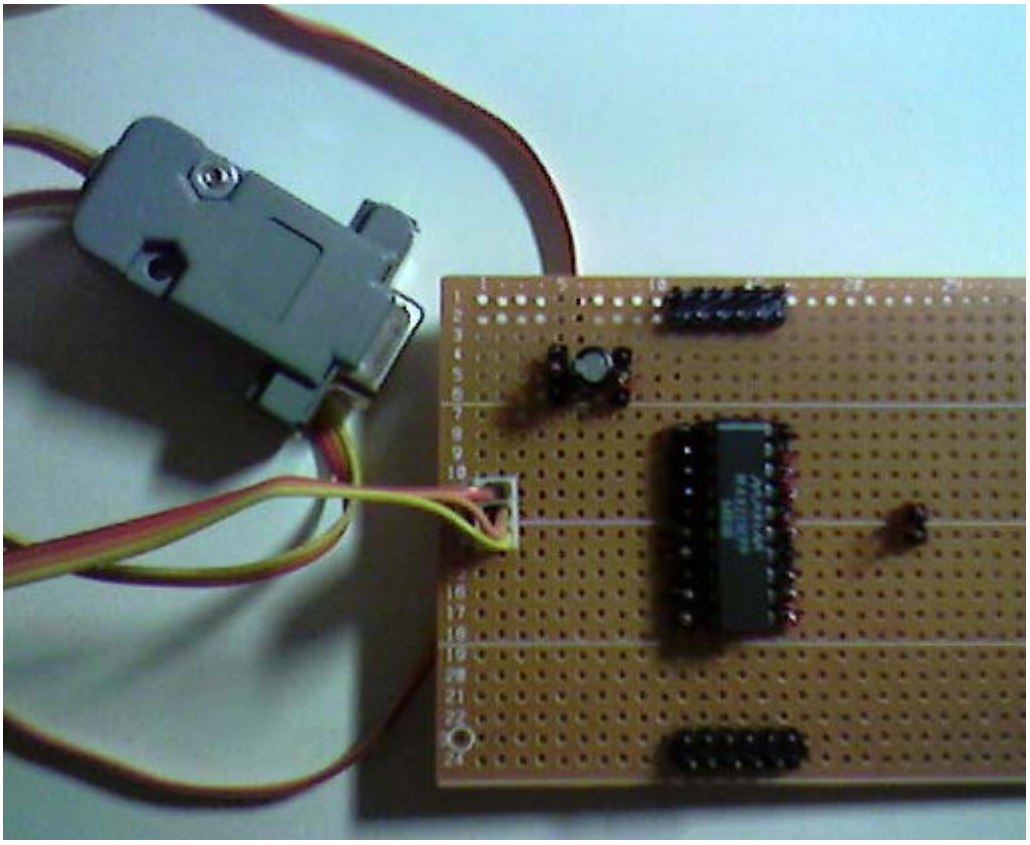
vii. www.altera.com

viii. JohnV. Oldfield and Richard C. Dorf (1995).*Field Programmable Gate Arrays:*A Wiley –Interscience Publication

APPENDIX



Visual Basic Program



Circuit of the MAX233 and DB9