Palestine Polytechnic University

College of Engineering and Technology
Electrical Engineering Department

Graduation Project

Design of Conferences Room model controlled Using CPLD and HDL Technologies

Project Team

Israa Mohammad Deeb

Motaz Lotfi Sabha

Project Supervisor
Eng. Mazen Zallum

Hebron-Palestine

Jan , 2009

جامعة بوليتكنك فلسطين

الخليل – فلسطين

كلية الهندسة والتكنولوجيا

دائرة الهندسة الكهربائية والحاسوب

Design of Conferences Room model controlled Using CPLD & HDL Technology

تصميم موديل للتحكم بقاعة مؤتمرات باستخدام تقنية CPLD و لغة HDL

فريق المشروع:

إسراء محمد ذيب معتز لطفي صبحا

بناء على نظام كلية الهندسة والتكنولوجيا وإشراف ومتابعة المشرف المباشر على المشروع و موافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع إلى دائرة الهندسة الكهربائية والحاسوب وذلك للوفاء بمتطلبات درجة البكالوريوس في هندسة أنظمة الحاسوب.

توقيع المشرف

.....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

.....

Dedication

To our parents who
spent nights and days doing their best
to give us the best...

To all students and who
Wish to look for
the future...

To who love the knowledge and
Looking for the new
in this world...

To who carry candle of science
To light his avenue
of life...

To our beloved country Palestine...

To all of our friends…

Acknowledgments

To our great supervisor, who offered his best for this project to see light through his instructions and advices, Eng .Mazen Zallum with all his kindness and wisdom we thank him.

We would also like to thank every person who offered anything to make this work success; we sincerely believe that this work wouldn't exist without his inspiration. Great thanks to our college for the support and help, and any one who helped us in our project.

So again thanks to:

Our great Palestine Polytechnic University (PPU)

Our College of Engineering and Technology (CET)

Our Electrical and Computer Department (ECD)

Abstract

This project was implemented by using Complex Programmable Logic Device (CPLD) technology as controller and Verilog Hardware Description Language (Verilog HDL) to control different applications in conference room model contains switches, sensors, door, curtain Heating Ventilation Air Conditioning (HVAC) system and fire alarm

The goal was achieved and we were controlling of different applications on model of conference room

تم تمثيل هذا المشروع باستخدام تقنيه CPLD وverilog HDL للتحكم في تطبيقات مختلفة من مجسم لقاعه المؤتمرات بحيث يحتوي على مفاتيح ومجسات وأبواب وستائر ونظام للتدفئة والتبريد ونظام الحماية من الحريق.

الهدف من هذا المشروع تحقق وتم التحكم بمختلف التطبيقات في مجسم قاعه المؤتمرات

Table of contents

List of Tables

List of Figure

Chapter 1

INTRODUCTION

1.1 General Idea

1.2 Project Objectives

1.3 Literature Review

1.4 Project Scheduling

1.5 Project Cost

1.6 Road Map

CHAPTER ONE

Introduction

1.1 General Idea

Complex Programmable Logic Device (CPLD) is One of the new techniques nowadays can use to controlling various system and application and this programmable device programmed by many different Hardware Description Language (HDL) such as Verilog HDL or VHDL .

Our project aim to design and implement a model of conference room based on Complex Programmable Logic Device CPLD as a controller and programming it by using Verilog Hardware Description Language (Verilog HDL).

The project implemented model for conference room , this model contains CPLD as controller, sensors and switches as input, doors, lighting, curtains, security system, Heating Ventilation and Air Conditioning (HVAC) system and microphones as output.

1.2 Project Objectives

- Study using CPLD technology as a controller.
- Design and implementation a control board using CPLD and Verilog HDL to control a conference room (model) .
- Actual control of different application as opening doors, open Curtains, controll of HVAC and fire alarm well be implemented (model).

1.3 Literature Review

There are many projects found using CPLD and FPGA as controller.

1.3.1 Control Traffic Light Using CPLD & HDL Technology

This project implemented of a Traffic Light system. The system is built to control traffic lights at 4 way cross roads with pedestrian passage ways and warning signals for both vehicles and pedestrians.

The project will be implemented with Complex Programmable Logic Device (CPLD) -Altera EPM 7064 SLC 44-10 – using Verilog Hardware Description Language (Verilog HDL).[1]

1.3.2 Embedded FPGA Security Devices in Remote Metering System

Embedded FPGA Security Devices in Remote Metering System; this project uses ways for security by using some encryption algorithms such as DES (Data Encryption Stander), RSA (Rivest, Shamir, and Adleman), Hashing algorithms, also design procedures using FPGA/CPLD and related mobile agent security solutions and protocols.[2]

1.4 Project Scheduling

The project activities here depend on each other, so the task durations and dependencies are the following:-

Table (1.1): The tasks duration for first semester

No.	Activities (Tasks)	Tasks name	Duration(week)
1	Collecting information and theoretical issues	T1	Four week
2	Project requirements analysis	T2	Five week
3	Study the Verilog HDL and CPLD Technology.	T3	Six week
4	Design block diagram	T4	Four week
5	Project implementation.	T5	Six week
6	Project testing	T6	Seven week
7	Project Documentation	T7	Twenty week
8	Final project		Thirty two week

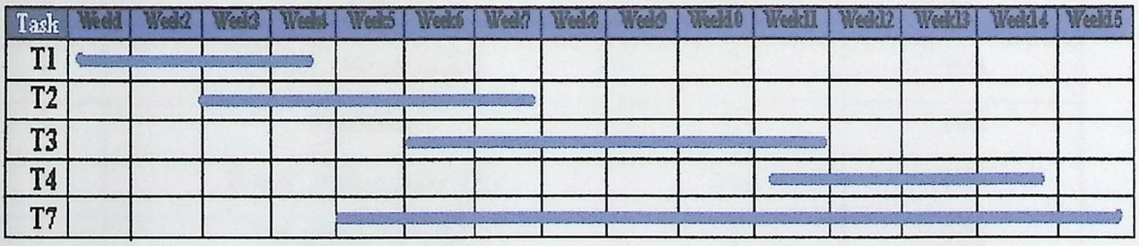


Figure (1.1): Project activity &time plane (1)

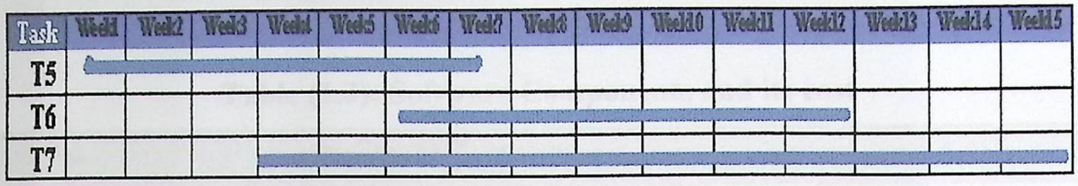


Figure (1.2): Project activity &time plane (2)

1.5 Project Cost

The cost of designing and implementing the project can be divided into three parts:

1.5.1 The Hardware Components

The table below shows the components and their cost.

Table (1.2): Hardware components and its cost

No.	Component	Quantity	Total Cost
1	Resistors	60	4$
2	Capacitors	10	4$
3	LEDs	50	5$
4	CPLD and its socket	1	40$
5	Switches	15	10$
6	Wires, cables and plugs	Set	10$
7	Complete Model	1	50$
8	DC motor	2	40$
9	Relay	2	5$
10	Three sensors (photo ,fire , LM35)	3	45$
Total			210$

1.5.2 Software programs

Table (1.3): Software Components and its cost

No.	Software Program	Cost
1	Veriloger HDL programmer	$300
2	Quartus II programmer	$500
3	Windows XP version	$500
4	Orcad Family Release 9.2	$400
5	SmartDraw 6	$300
Total		**$2000**

1.5.3 Human cost

The team of the project consists of two students, work in 30 week, 4 hour a day and the work's hour costs 10$. So, the human cost equal (4 * 30 * 2 * 5 * 10 = $12000).

1.5.4 Total cost

The total cost contains the hardware equipments and software programs, the total cost reach about approximately ($14200).

1.6 Road Map

Report consists of seven chapters; the following is a brief description of the topics that are covered in each chapter.

Chapter 2: Theoretical Background

This chapter talks in more details about the basic project theoretical background.

Chapter 3: Conceptual Design

This chapter details the design concepts, introduces project objectives, shows the general block diagram of the system and explains how system works.

Chapter 4: Hardware System Design

This chapter presents a formal procedure for designing, discusses design options and justifies those that will be used in this project.

Chapter 5: Software System Design

This chapter handles the software related to the intended system, depicts flowcharts about system operations.

Chapter Six: System Implementation and Testing

This chapter includes the implementation phase with the testing of this phase. General hardware and software component tested are shown in this chapter.

Chapter Seven: Conclusion and Future Work

This chapter provides the conclusions that will be concluded after working the system, and suggestion for future work.

Chapter 2

THEORETICAL BACKGROUND

2.1 Introduction

2.2 Theoretical Background of the project

2.3 Hardware and software Components.

2.4 Project Integrity

The design of combinational circuits with PLA and PAL are presented in the next two sections. [3]

2.2.2 Sequential programmable devices

Digital systems are designed using flip-flops and gates. Since the combinational PLD consists of only gates, it is necessary to include external flip-flops when they are used in the design. Sequential programmable devices include both gates and flip-flops. In this way, the device can be programmed to perform a variety of sequential-circuit functions. There are several types of sequential programmable devices available commercially and each device has vendor-specific variant within each type. The internal logic of these devices is too complex to be shown here. Therefore, we will describe three major types without going into their detailed construction:

1. Sequential (or simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field programmable gate array (FPGA)

CHAPTER TWO
Theoretical Background

2.1 Introduction

This chapter focuses on theoretical subjects related to the main idea of the project, and information about the components used in the project.

2.2 Theoretical Background of the project

2.2.1 Combinational PLDs.

The PROM is combinational programmable logic device (PLD). A combinational PLD is an integrated circuit with programmable gates divided into an AND array and an OR array to provide an AND-OR sum of product implementation. There are three major types of combinational PLDs and they differ in the placement of the programmable connections in the AND-OR array. Figure 2.2 shows the configuration of the three PLDs. The programmable read-only memory (PROM) has a fixed AND array constructed as a decoder and programmable OR array. The programmable OR gates implement the Boolean functions in sum of minterms. The programmable array logic (PAL) has a programmable AND array and a fixed OR array. The AND gates are programmed to provide the product terms for the Boolean functions, which are logically summed in each OR gate. The most flexible PLD is the programmable logic array (PLA), where both the AND and OR arrays can be programmed. The product terms in the AND array may be shared by any OR gate to provide the required sum of products implementation. The names PAL and PLA emerged from different vendors during the development of programmable logic devices. The implementation of combinational circuits with PROM was demonstrated before.

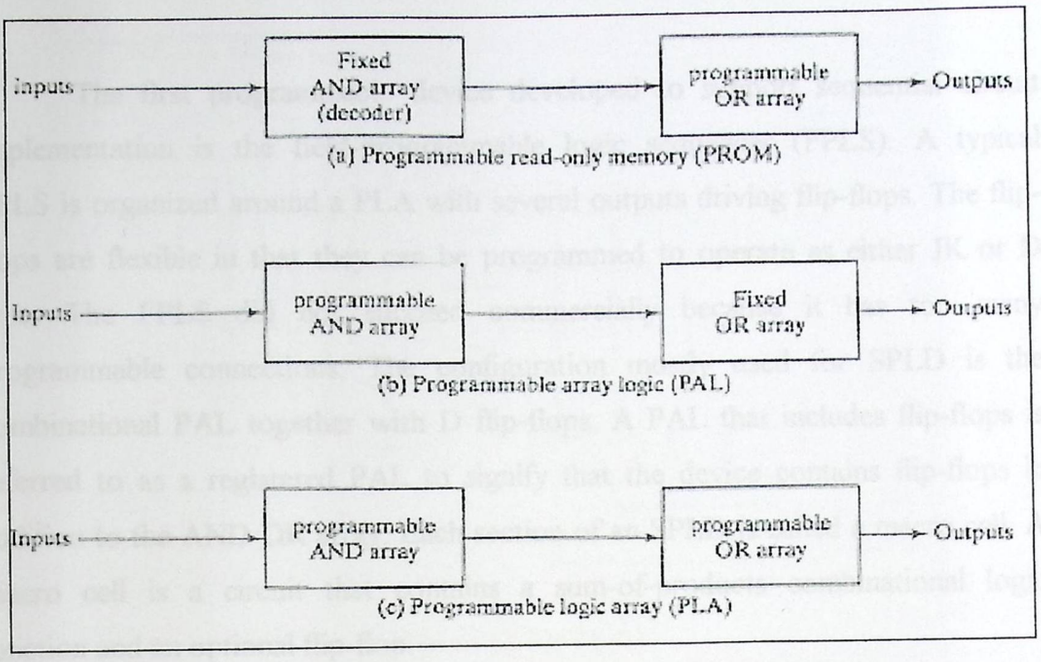


Figure (2.1): Basic configurations of three PLDs

2.2.3 SPLD

The sequential PLD is sometimes referred to as a simple PLD to differentiate it from the complex PLD. SPLD includes flip-flops within the integrated circuit chip in addition to the AND-OR array. The result is a sequential circuit as shown in Figure 2.2. A PAL or PLA is modified by including a number of flip-flops connected to form a register. The circuit outputs can be taken from the OR gates or from the outputs of the flip-flops. Additional programmable connections are available to include the flip-flop outputs in the product terms formed with the AND array. The flip-flops may be of the D or the JK type.

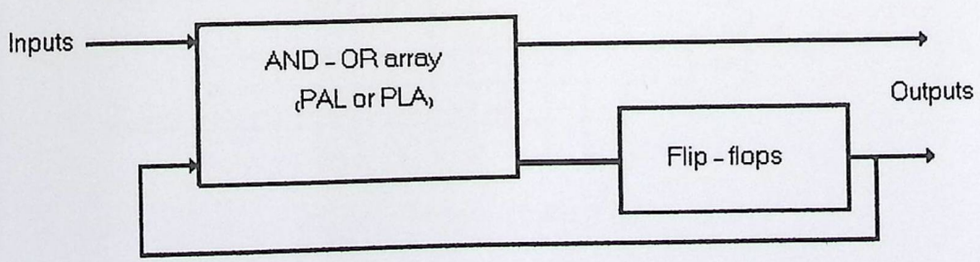


Figure (2.2): SPLD [4]

The first programmable device developed to support sequential circuit implementation is the field-programmable logic sequencer (FPLS). A typical FPLS is organized around a PLA with several outputs driving flip-flops. The flip-flops are flexible in that they can be programmed to operate as either JK or D type. The FPLS did not succeed commercially because it has too many programmable connections. The configuration mostly used for SPLD is the combinational PAL together with D flip-flops. A PAL that includes flip-flops is referred to as a registered PAL to signify that the device contains flip-flops in addition to the AND-OR array. Each section of an SPLD is called a macro cell. A macro cell is a circuit that contains a sum-of-products combinational logic function and an optional flip-flop.

2.2.4 PLA

A Programmable Logic Array (PLA) as shown on figure 2.3 is a relatively small programmable logic devices (PLDs); that contains two levels of logic, an AND-plane and an OR-plane, where both levels are programmable (note: although PLA structures are sometimes embedded into full-custom chips, we refer here only to those PLAs that are provided as separate integrated circuits and are user-programmable).

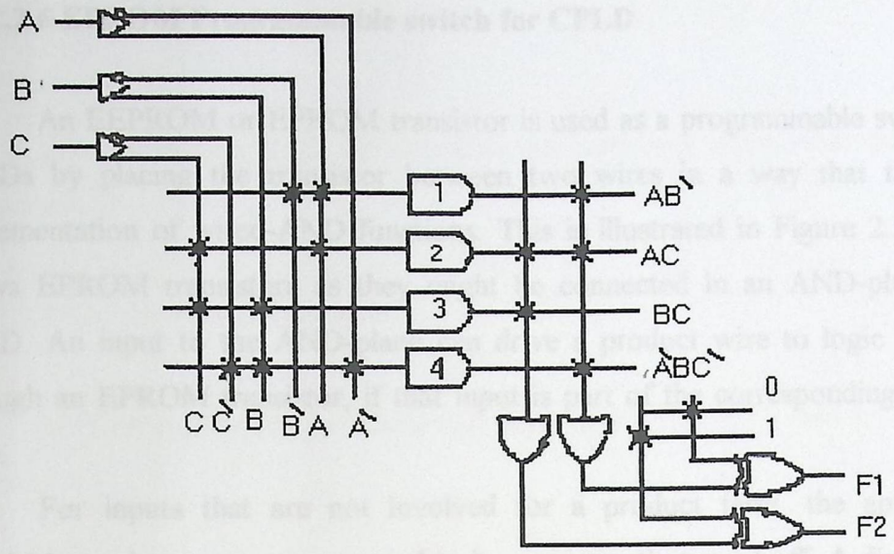


Figure (2.3): PLA

2.2.5 PAL

A Programmable Array Logic (PAL) is a relatively small (PLDs) that has a programmable AND-plane followed by a fixed OR-plane.[5]

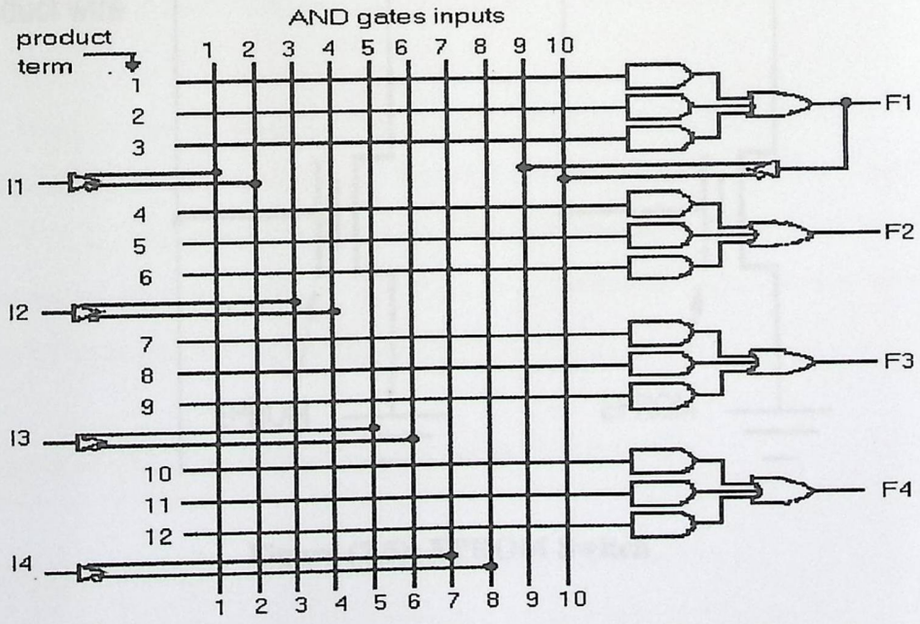


Figure (2.4): PAL

2.2.6 EPROM Programmable switch for CPLD

An EEPROM or EPROM transistor is used as a programmable switch for CPLDs by placing the transistor between two wires in a way that facilitates implementation of wired-AND functions. This is illustrated in Figure 2.5, which shows EPROM transistors as they might be connected in an AND-plane of a CPLD. An input to the AND-plane can drive a product wire to logic level '0' through an EPROM transistor, if that input is part of the corresponding product term.

For inputs that are not involved for a product term, the appropriate EPROM transistors are programmed to be permanently turned off. A diagram for an EEPROM based device would look similar.[6]

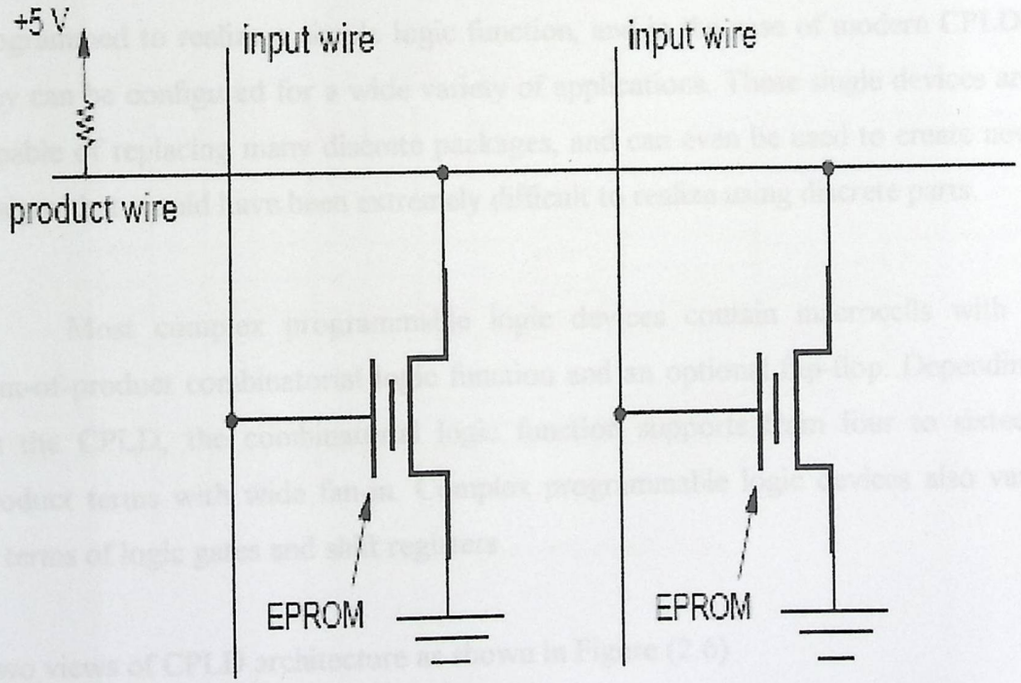


Figure (2.5): EPROM Switch

2.3 Components

There are two kinds of components hardware component and software component.

2.3.1 Hardware Components

In This section will mentions and explains the hardware that are used in this project.

2.3.1.1 Complex Programmable Logic Devices (CPLD)

CPLDs (Complex Programmable Logic Devices) are chips that can be programmed to realize a simple logic function, and in the case of modern CPLDs they can be configured for a wide variety of applications. These single devices are capable of replacing many discrete packages, and can even be used to create new designs that would have been extremely difficult to realize using discrete parts.

Most complex programmable logic devices contain macrocells with a sum-of-product combinatorial logic function and an optional flip-flop. Depending on the CPLD, the combinatorial logic function supports from four to sixteen product terms with wide fan-in. Complex programmable logic devices also vary in terms of logic gates and shift registers

Two views of CPLD architecture as shown in Figure (2.6)

1- A CPLD is basically a huge interconnect array between SPLDs
2- A SPLD block is basically Programmable Array Logic (PAL) or Programmable Logic Arrays (PLA) and Macrocells. PAL and PLA are the predecessors of CPLD.

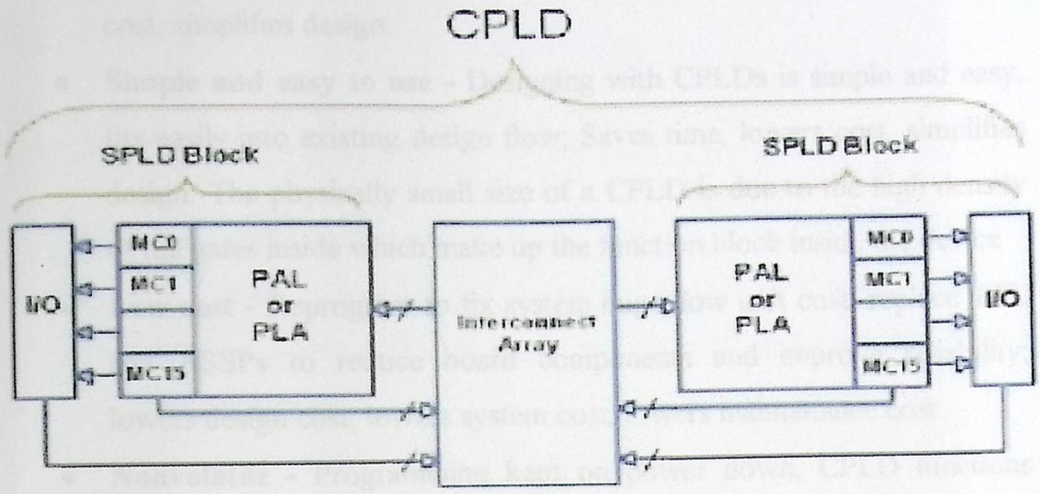


Figure (2.6): CPLD architecture[7]

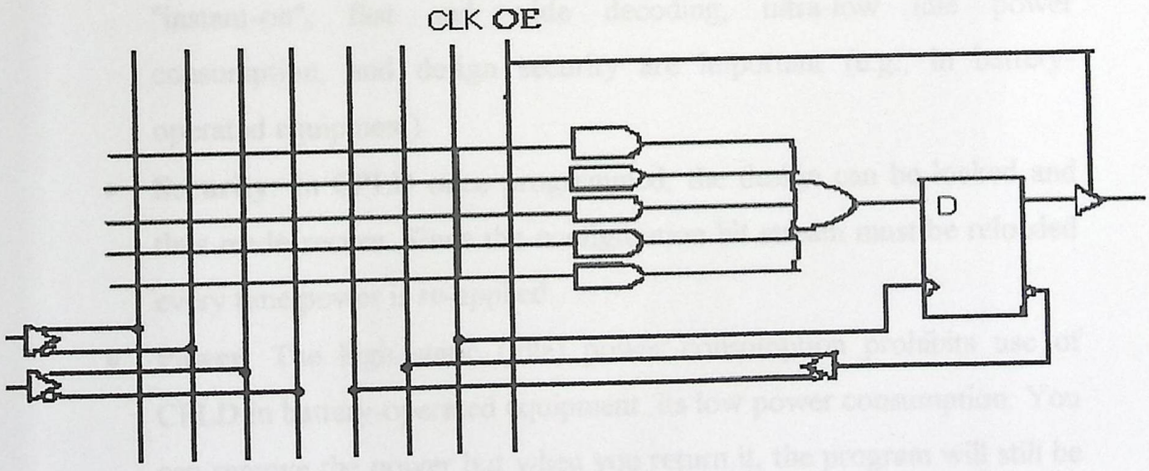


Figure (2.7) Basic Macrocell Logic

Features of CPLD

A CPLD is greatly praised for its nonvolatile characteristics :

- **Reprogrammable** - Change your design instantly for no cost as many times as you like, build reconfigurable systems, fix ASIC bugs,

upgrade system functions anytime from anywhere; Saves time, lowers cost, simplifies design.

- **Simple and easy to use** - Designing with CPLDs is simple and easy, fits easily into existing design flow; Saves time, lowers cost, simplifies design. The physically small size of a CPLD is due to the high density of the gates inside which make up the function block inside the device
- **Low cost** - Reprogram to fix system bugs, low unit cost, replace TTL and ASSPs to reduce board components and improve reliability; lowers design cost, lowers system cost, lowers maintenance cost.
- **Nonvolatile** - Programming kept on power down, CPLD functions available instantly on system power up, almost impossible to steal stored design; Improves security, simplifies design.
- **Speed**: CPLDs offer a single-chip solution with fast pin-to-pin delays, even for wide input functions. Use CPLDs for small designs, where "instant-on", fast and wide decoding, ultra-low idle power consumption, and design security are important (e.g., in battery-operated equipment).
- **Security**: In CPLD once programmed, the design can be locked and thus made secure. Since the configuration bit stream must be reloaded every time power is re-applied.
- **Power**: The high static (idle) power consumption prohibits use of CPLD in battery-operated equipment. its low power consumption. You can remove the power but when you return it, the program will still be there.[8]

Applications of CPLDs

Because they offer high speeds and a range of capacities, CPLDs are useful for a very wide assortment of applications, from implementing random glue logic to prototyping small gate arrays. One of the most common uses in industry at this time, and a strong reason for the large growth of the CPLD

market, is the conversion of designs that consist of multiple SPLDs into a smaller number of CPLDs.

2.3.1.2 DC Motor

A DC motor works by converting electric power into mechanical work. This is accomplished by forcing current through a coil and producing a magnetic field that spins the motor. The simplest DC motor is a single coil apparatus, used here to discuss the DC motor theory.

A typical DC motor consists of two parts:

1. An outside stationary stator having coils supplied with DC current to produce a rotating magnetic field.
2. An inside rotor attached to the output shaft that is given a torque by the constant field and sequential changing of polarity of magnetic field.[9]

Figure (2.8): DC Motor

2.3.1.3 Fire Detector

A fire alarm system is an active fire protection system that detects fire or the effects of fire, and as a result provides one or more of the following: notifies

the occupants, notifies persons in the surrounding area, summons the fire service, and controls all the fire alarm components in a building. Fire alarm systems can include alarm initiating devices, alarm notification appliances, control units, fire safety control devices, annunciates, power supplies, and wiring.[10]

Figure (2.9): Fire Detector

2.3.1.4 Temperature Sensor

Semiconductor temperature sensors are produced in the form of ICs. Their fundamental design results from the fact that semiconductor diodes have temperature-sensitive voltage vs. current characteristics.

The use of IC temperature sensors is limited to applications where the temperature is within a –55° to 150°C range. The measurement range of IC temperature sensors may be small compared to that of thermocouples and RTDs, but they have several advantages: they are small, accurate, and inexpensive, and are easy to interface with other devices such as amplifiers, regulators, DSPs, and microcontrollers.

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.[11]

Features LM35

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 μA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±1⁄4°C typical

- Low impedance output, 0.1 W for 1 mA load

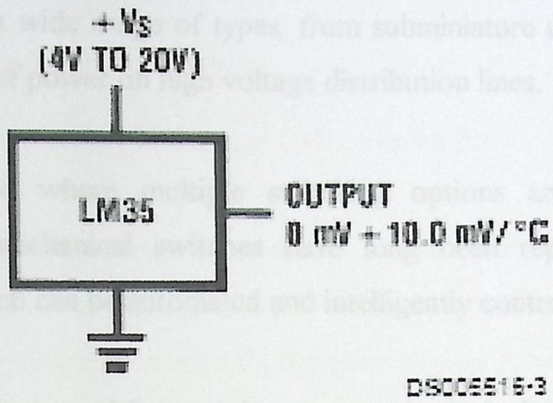


Figure (2.10): Temperature Sensor LM35

2.3.1.5 Photo Resistor

A photoresistor or Light Dependent Resistor or CdS Cell is a resistor whose resistance decreases with increasing incident lightest. It can also be referred to as a photoconductor.

A photoresistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electron (and its hole partner) conduct electricity, thereby lowering resistance. [12]

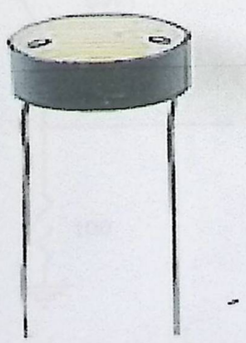


Figure (2.11): Photoresistor

2.3.1.7 Switch

A switch is a mechanical device used to connect and disconnect a circuit at will. Switches cover a wide range of types, from subminiature up to industrial plant switching megawatts of power on high voltage distribution lines.

In applications where multiple switching options are required (e.g., a telephone service), mechanical switches have long been replaced by electronic switching devices which can be automated and intelligently controlled.

The prototypical model is perhaps a mechanical device (for example a railroad switch) which can be disconnected from one course and connected to another.

The switch is referred to as a "gate" when abstracted to mathematical form. In the philosophy of logic, operational arguments are represented as logic gates. The use of electronic gates to function as a system of logical gates is the fundamental basis for the computer—i.e. a computer is a system of electronic switches which function as logical gates.

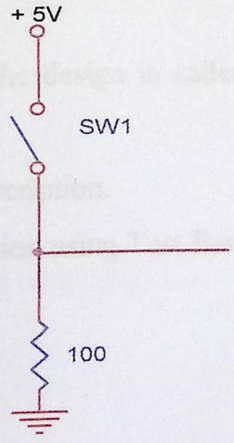


Figure (2.12): Switch

2.3.2 Software component

In this section we describe the software component that can be used in project

2.3.2.1 Hardware Description Language (HDL) software

HDL stands for Hardware Description Language used to describe digital systems in textual form, it is a programming language oriented to the structure and behavior of digital systems and is used to represent digital system in a form that can be read by humans and computers. There are two applications of HDL processing:

- Logic Simulation.
- Logic Synthesis.

2.3.2.1.1 Logic Simulation

The representation of the structure and behavior of digital system using the computer, the simulator interrupts the HDL description and produce readable output (i.e. timing diagram). This process help the designers in predicting how digital systems will behave before it is fabricated. This process allows designer to detect errors in design without physically create it.

The stimulus that tests the design is called Test Bench which is written in HDL to simulate a digital system:

- Writing the HDL description.
- Verifying the simulation using Test Bench.

2.3.2.1.2 Logic Synthesis

Driving a list of components and its interconnection (net list) from a module described in HDL, it is used to fabricate an integrated circuits (IC's) or printed circuit

board, same as conventional high level languages, the difference it produces a database with instruction to fabricate the physical piece of digital system.

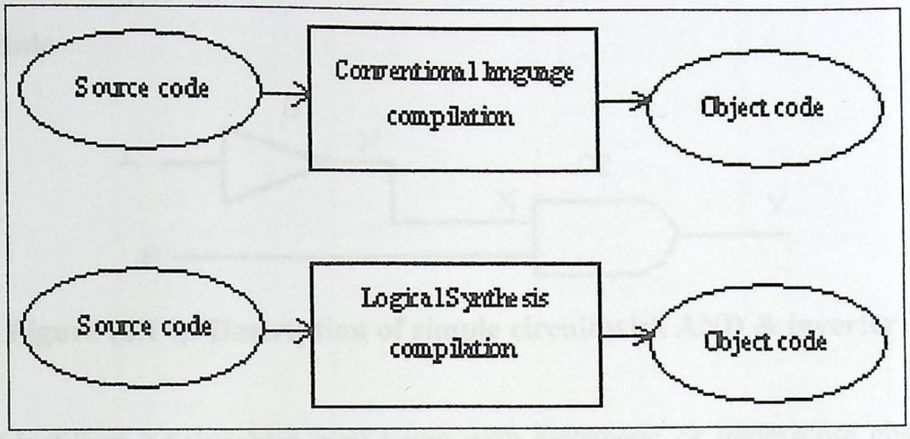


Figure (2.13): Compilation

There are two main standards HDL that are supported by IEEE (Institute of Electrical and Electronic Engineer): VHDL which was developed by Department of Defense and used for defense constructors but now are used commercially and in research universities, and Verilog HDL to discuss below.[13]

2.3.2.2 Verilog HDL Programming

It was introduced by Cadence Data System (late 1990), Verilog HDL is easier than VHDL and it is like C Programming language.[14]

Verilog HDL description

We will describe a circuit with AND and inverter gates shown in Figure 2.15

```
module and_gate(A,B,y);
        input A,B;          //Defining input ports
        output y;           //Defining output port
```

```
    wire x;              //Defining internal connection
    not G1(x,A);
    and G2(y,x,B);
end module
```

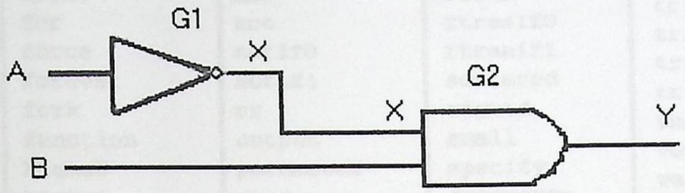


Figure (2.14): Description of simple circuit with AND & inverter

Identifiers or variables may begin with alphabetic or underscore characters, and may contain (0-9, $, _).

Verilog has four logic values: - 0, 1, z or Z (high impedance), x or X (unknown).

The general format for number is a Size, base, and value where:

Size (optional) represents the number of bits in number, default is 32bit.

Base (optional) represents the base of number, default base is integer

Table 2.1 Numbers Format

Binary	B or b	0,1,X,x,Z,z
Octal	O or o	0-7,X,x,Z,z
Decimal	D or d	0-9
Hexadecimal	H or h	0-9,A-F,a-f,X,x,Z,z

The keywords in Verilog are about 100 keywords that are used by Verilog HDL; each keyword must be in lowercase. Table 2.2 lists the keywords used in Verilog HDL language:

Table 2.2 Verilog HDL keywords [15]

always	endmodule	large	reg	
and	endprimitive	macromodule	release	tranif0
assign	endspecify	nand	repeat	tranif1
attribute	endtable	negedge	rnmos	tri
begin	endtask	nmos	rpmos	tri0
buf	event	nor	rtran	tri1
bufif0	for	not	rtranif0	triand
bufif1	force	notif0	rtranif1	trior
case	forever	notif1	scalared	trireg
casex	fork	or	signed	unsigned
casez	function	output	small	vectored
cmos	highz0	parameter	specify	wait
deassign	highz1	pmos	specparam	wand
default	if	posedge	strength	weak0
defparam	ifnone	primitive	strong0	weak1
disable	initial	pull0	strong1	while
edge	inout	pull1	supply0	wire
else	input	pulldown	supply1	wor
end	integer	pullup	table	xnor
endattribute	join	rcmos	task	xor
endcase	medium	real	time	
endfunction	module	realtime	tran	

Operators

Arithmetic Operators: +, -, *, /, %

Bit-wise Operators: ~ (Inverting), & (ANDing), | (ORing), ^ (XOR)

Logical Operators: ! (Is not true), && (m&&n mean are both true), ||, = =,! =

Relational Operators: >, <, >=, <=

Logical shift Operators: << (shift left), >> (shift right)

- **Gate-Level modeling**: - using **instantiation** of primitive gates and user-defined module.
- **Dataflow modeling**:- Using continues assignment statements with keyword **assign**
- **Behavioral modeling** :- Using procedural statements with the keyword always

Gate-Level Modeling

- Describes the circuit by specifying the gates & their interconnection.
- Provide a textual description of a schematic diagram.
- Verilog recognize 12 basic gates

Boolean Expressions are specified in HDL with the keyword assign followed by the Boolean Expressions (BE).

Verilog uses the following symbols for BE

& for AND | for OR

~ for NOT ^ for XOR

The BE y= A.B +C' is described in Verilog HDL:-

assign y = (A&B) | ~C

Dataflow modeling

Gate-level modeling works well for small circuits:

- Because number of gate is limited.
- Well suited to designer with basic knowledge of digital logic design.

In complex design, # of gates is very large, and it is more effective to use higher modeling (dataflow).

Table 2.3 Verilog HDL operators

Operator Type	Symbol	Operation Performed
Arithmetic	+	addition
	-	subtraction
	*	multiplication
	/	division
	%	modulus
Logic (bit-wise or reduction)	~	negation (complement)
	&	AND
	|	OR
	^	Exclusive-OR (XOR)
Logical	!	negation
	&&	AND
	||	OR
Shift	>>	shift right
	<<	shift left
	{ , }	concatenation
Relational	>	greater than
	<	less than
	==	equality
	!=	inequality
	>=	greater than or equal
	<=	less than or equal

Other Operators:

Table 2.4 Other Verilog HDL operators

Operator	Example	Description
?:	X?y:z	If x true select y if x false select z
{ }	{m,n}	Concatenate m to n

HDL for combinational circuits

A **module** is described in any one of the following modeling techniques:

Dataflow provides a powerful way to implement a design (especially when # of gates very large). Verilog allows a circuit to be designed in terms of data flow between registers (rather than instantiation)

A process of creating gate-level circuits from dataflow design, dataflow is a popular and sophisticated approach as a logic synthesis tools. Dataflow modeling uses a number of operators that act operands to produce the desired result. Verilog provides 23 operators type . It is important to distinguish between arithmetic and logic operators (+ addition, while & is logic ADD). Dataflow modeling continuous assignment start with the keyword assign. Continuous assignment is a statement that assigns a value to a net (physical connection between circuit elements). A net defines a gate O/P declared by an output or wire statement.

Behavioral Modeling

Behavioral Modeling is the process of representing the digital circuits at a functional & algorithm level. Verilog allows us to describe design functionality in an algorithm manner (i.e. describe the behavior ...).

Design using behavioral modeling resembles C more than resembles digital circuit design. Behavioral Modeling mostly used for sequential. Behavioral Modeling Allows designer to evaluate the trade-offs of various architecture & algorithms, then choosing the optimum to implement hardware. Verilog is rich in behavioral constructs that provide the design with a great amount of flexibility. Two structured procedure statements in Verilog: - **always & initial,** they are the most basic statements in behavioral modeling. All other behavioral statements appear only inside these statements.

Behavioral, descriptions use the keyword always followed by a list of procedural (behavior) statements. An always statements constitute an always block.

The always statement always starts at time 0 and execute statement in the block continuously (looping).

Target output; The target output of a procedural statement must be of **reg** data type (not a **wire**), **reg** data type remains unchanged until a new value is assigned by procedural assignment. **wire** data type may continuously updated with assignment statements. [16]

2.4 Project integrity

This system does not cause any problem either for the human or for the environment, it has humanistic helpful.

Chapter 3

PROJECT CONCEPTUAL DESIGN

3.1 Introduction

3.2 Detailed Project Objectives

3.3 Design Options

3.4 Design Realization Approach

3.5 Project Design Block Diagram

3.6 Project Interaction with the Surrounding Environment

CHAPTER THREE
Project Conceptual Design

3.1 Introduction

This chapter shows the design concepts and the block diagram of the system.

3.2 Project Objectives

- Study using CPLD technology as a controller, so the design will using CPLD to control of the following applications :

  a- Lighting: the CPLD will control of open or close the lighting manually by input the value through switches to the CPLD or automatically by photo sensor send signal to CPLD, the CPLD will accept this value to do the right action.

  b- Door: the CPLD will control this application by input the value through switches to the CPLD then the CPLD will send signal to the DC motor to open or close the door.

  c- Curtains:at this application the CPLD will control it by input the value through switches to the CPLD then the CPLD will send signal to the DC motor to open or close the Curtains.

  d- Fire alarm : the CPLD will control of fire alarm when receive a signal from fire sensor then the CPLD will send a signal to open the door curtain and alarm on.

e- HVAC: to open air conditions or heater the temperature sensor will sense the temperature in the surrounding area and send the value of temperature to the CPLD to do the right action.

- Design and implementation a control board using CPLD and Verilog HDL to control a conference room (model).
  using a model of conference room contains LED's instead of light , fire sensor for fire alarm , a DC motor to open or close a small door and curtains , and temperature sensor for HVAC .

- Actual control of different application as opening doors, open Curtains, HVAC and fire alarm will be implemented (model). So after implantation ,simulation and modeling the project , the user will be able to control of each application

3.3 Design Options

Depending on the requirements of the system there are various options for the design as the following:

a) To control of the different application on conference room there are many technique can be use:
   1. CPLD.
   2. FPGA.
   3. Programmable interface controller (PIC).

This project will use CPLD as controller and the reason of choosing this option refers to:

- The CPLD can be reprogrammable by change the design instantly for no cost as many times as you like
- Designing with CPLDs is simple and easy, fits easily into existing design flow; Saves time, lowers cost, simplifies design. The physically small size of a CPLD is due to the high density of the gates inside which make up the function block inside the device
- Nonvolatile so can kept Programming on power down, CPLD functions available instantly on system power up.
- The Security in CPLD, so once programmed in CPLD, the design can be locked and thus made secure. Since the configuration bitstream must be reloaded every time power is re-applied.
- The Speed of the CPLDs offer a single-chip solution with fast pin-to-pin delays, even for wide input functions. Use CPLDs for small designs, where "instant-on", fast and wide decoding, ultra-low idle power consumption.
- The high static (idle) power consumption prohibits use of CPLD in battery-operated equipment. its low power consumption. so You can remove the power but when you return it, the program will still be there

b) For using CPLD technology there are many options can be use:

1. dealing with CPLD as IC .
2. dealing with kit of CPLD .

This project will dealing with kit of CPLD and the reason of choosing this option refers to:

- mush easier to dealing of pins
- the pins of programmer is already connect
- pins I/O is connected insanely

- has a clock

c) The CPLD can programmable by many hardware language such as:

1. Verilog HDL
2. VHDL

This project will use Verilog HDL as hardware language and the reason of choosing this option refers to:

- Verilog HDL is easier than VHDL.
- Its syntax similar to the C programming language.

d) This design must be control the door opening operation by an order come from the CPLD through interfacing unit between CPLD and the door ; there are many techniques can be used to make this such that:

1. DC motor
2. AC motor

This project will use DC motor for open the door and the reason of choosing this option refers to:

- using DC motors is the wide availability of AC power
- DC motors generally cost less than AC motors
- DC motors work well in constant-speed applications, because its speed is determined by the frequency of the DC voltage applied to the motor terminals.

3.4 Design Realization approach

After studying the previous design options we have decided to make the project in several stages and to examine each stage separately to ensure readiness for work and then the project was put in a special case, easy-to-use box and applying the design realization approach by making the implementation, modeling and simulation.

3.5 General Block Diagram

As shown in the Figure 3.1, This system composed of seven units:

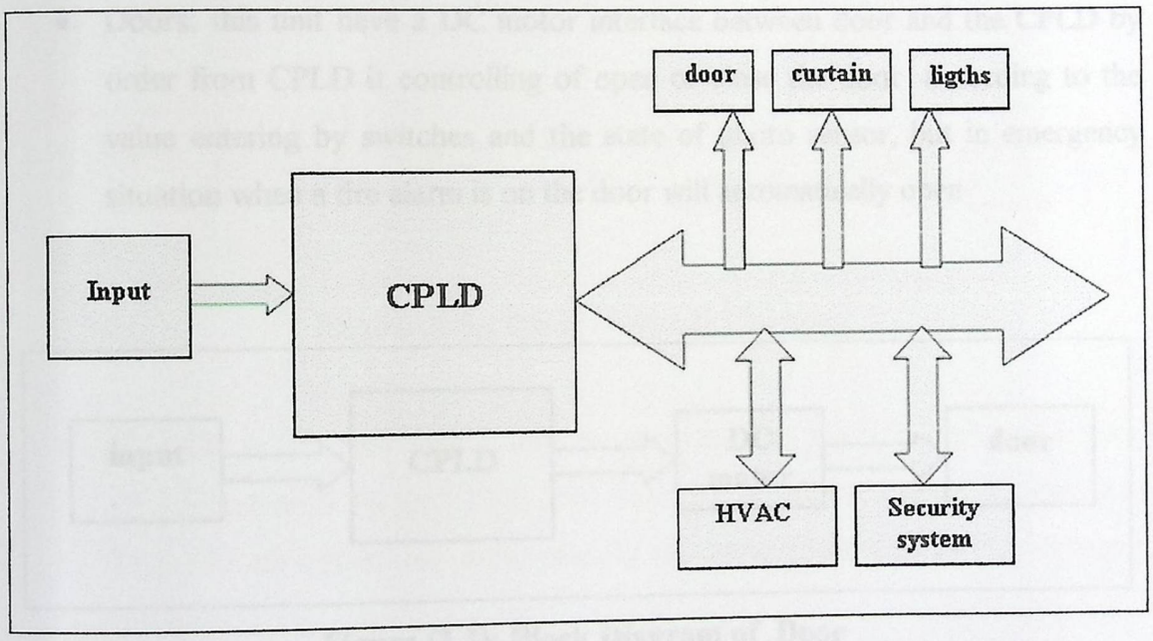


Figure (3.1): General Block Diagram

- CPLD: this unit is a basic unit that controlling of all system by takes the value from switches and appears the result on other units.
- Input : this unit use switches to input value to the CPLD
- Fire alarm : These units have fire detector sense the smoke and send a digital signal to the CPLD to open the door and switch on an alarm.

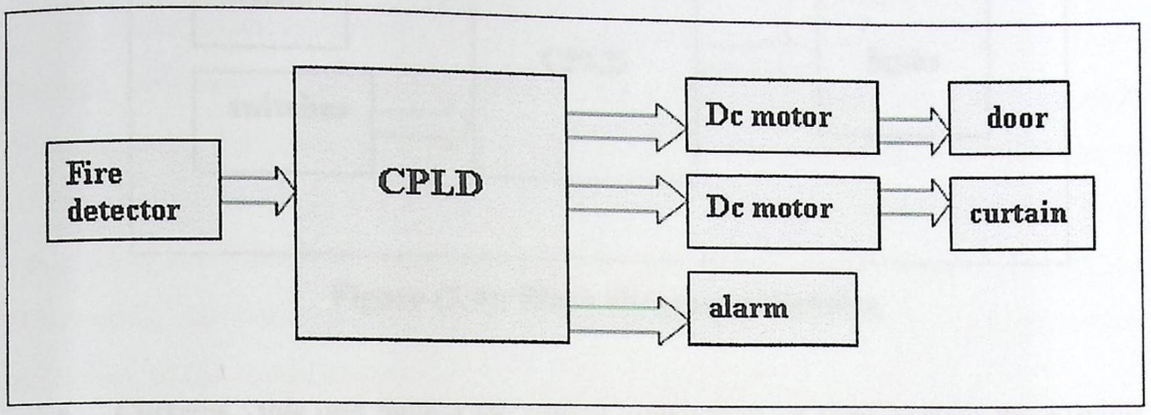


Figure (3.2): Block Diagram of fire alarm

- Doors: this unit have a DC motor interface between door and the CPLD by order from CPLD it controlling of open or close the door according to the value entering by switches and the state of photo sensor, but in emergency situation when a fire alarm is on the door will automatically open

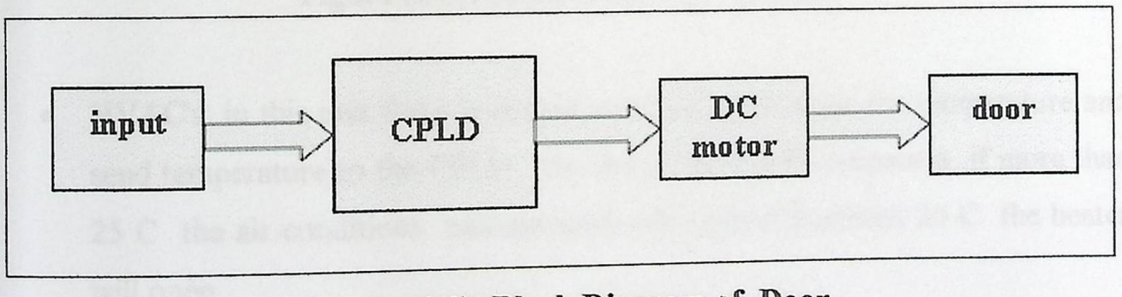


Figure (3.3): Block Diagram of Door

- Lighting: in this unit contains can controlling of LED's automatically according of photo sensor or manually by switches.

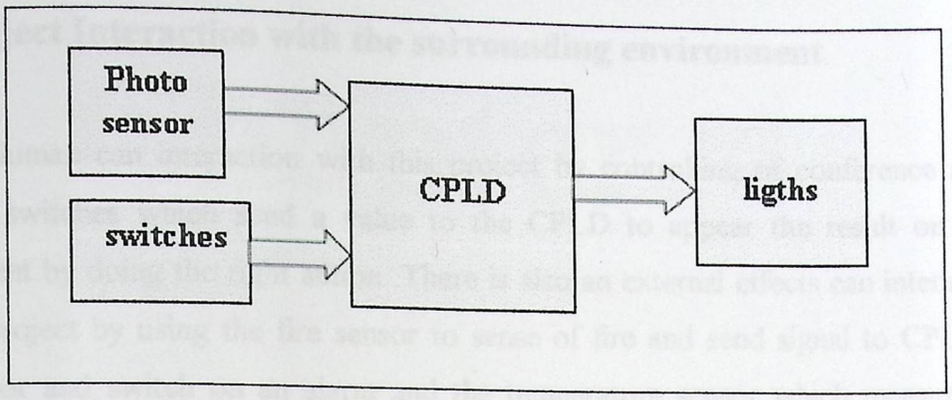


Figure (3.4): Block Diagram of lighting

- Curtains : this unit have a DC motor controlling of open or close the curtains according to the value entering by switches

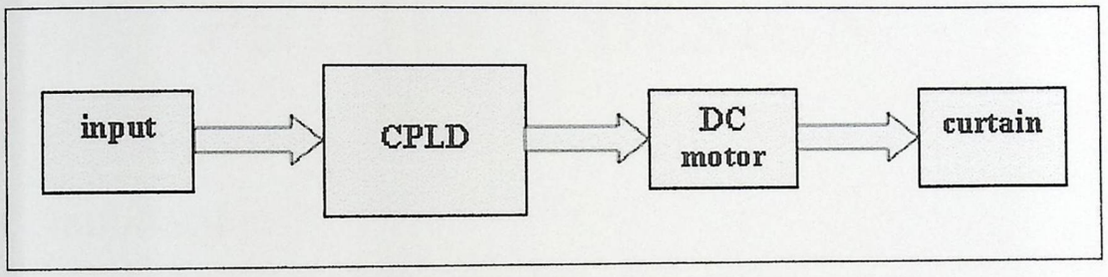


Figure (3.5): Block Diagram Curtains

- HVAC's: in this unit there is temperature sensor to sense the temperature and send temperature to the CPLD , the then CPLD will compare it ,if more than 25 C the air conditions automatically will open if less than 20 C the heater will open

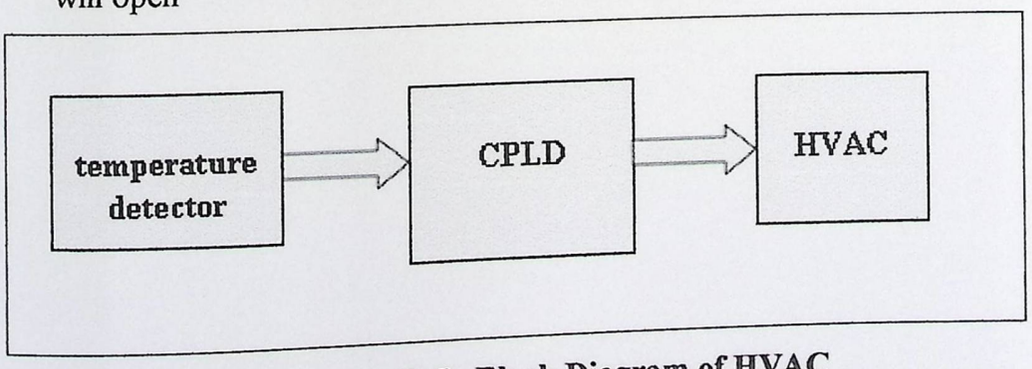


Figure (3.6): Block Diagram of HVAC

3.6 Project Interaction with the surrounding environment

Human can interaction with this project by controlling of conference room through switches which send a value to the CPLD to appear the result on each component by doing the right action .There is also an external effects can interaction of this project by using the fire sensor to sense of fire and send signal to CPLD to open door and switch on an alarm and the temperature sensor which measure the surrounding temperature then send the value to the CPLD to do the right action according to the temperature.

Chapter 4

DETAILED TECHNICAL PROJECT DESIGN

4.1 Introduction

4.2 Detailed Description of the Program Phases

4.3 Subsystem Detailed Design

4.4 overall system designs

4.5 User System interface

CHAPTER FOUR
Detailed Technical Project Design

4.1 Introductions

This chapter outlines formal procedure for design and the overall system design

4.2 Detailed Description of the Program Phases

The system goes through three main phases: the input, the processing and the output. These phases are explained as the following:

4.2.1 The Input Phase

In this phase there are two different inputs can input the signal to the CPLD:

a) On/Off Switch : users will controlling to open and close the door, curtains, microphones and lights by on/off switches these switch will input value to the CPLD to process it .

b) Sensors : for automatic control of security we used fire sensor to sense the smoke, the temperature sensor to sense the temperature and the photo sensor to sense the light on the door, these sensors input value to the CPLD to process it.

4.2.2 The Processing Phase

In this phase the CPLD take the value from input phase and process it according of program that written by Verilog HDL language.

4.2.3 The Output Phase

This phase shows the result of processing phase, The output must achieve the following:

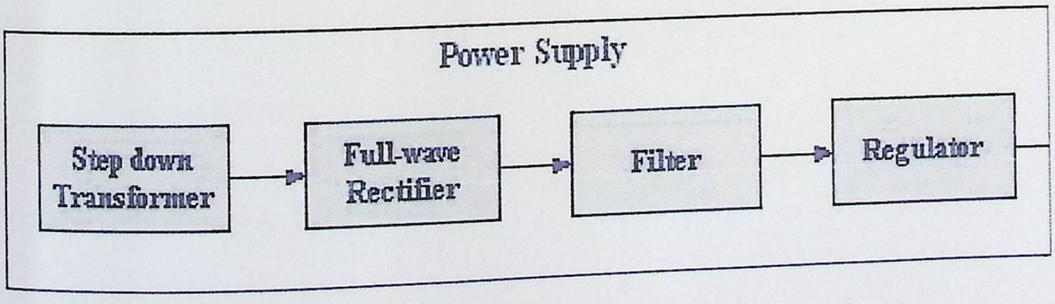
- The door, curtains and lights open or close according of switches if on or off.
- In security system if there is a smoke the alarm will be turn on and the door will open
- In HVAC's system when temperature is more than 25 C the air-condition will open and when the temperature is less than 20 C the heater will open.

4.3 Subsystem Detailed Design

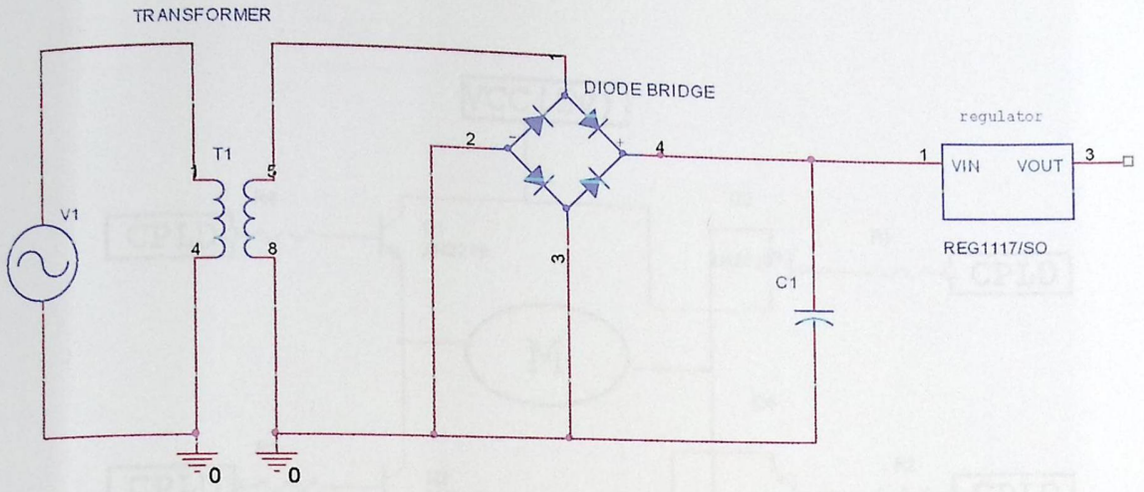
This section contains detailed description of the subsystems.

4.3.1 Power Supply

The AC signal enters the power supply part, goes to the step down transformer to gain the required voltage then goes to the full-wave rectifier and the filter. After that the current enters the regulator .These steps provide us with the needed DC voltage for the relay as shown in the following Figure 4-2.



(a)



(b)

Figure (4.1): (a) Block Diagram of the Power Supply
(b) Circuit Diagram for the Power Supply

4.3.2 H- Bridge Circuit

The circuit consists Four transistors connected with each other , collector connect to the Vcc and emitter connect to the ground motor connect between the four transistor to tern left and tern right ,four resistor connect from the base to output pins in the CPLD this circuit use to open or close the door and curtain by tern on the motor on the left or right if only Q3 and Q2 feed a 5 volt from CPLD then the motor will tern on left to open the door or curtain otherwise if only Q4 and Q5 feed from CPLD the motor will tern on right to close the door or curtain

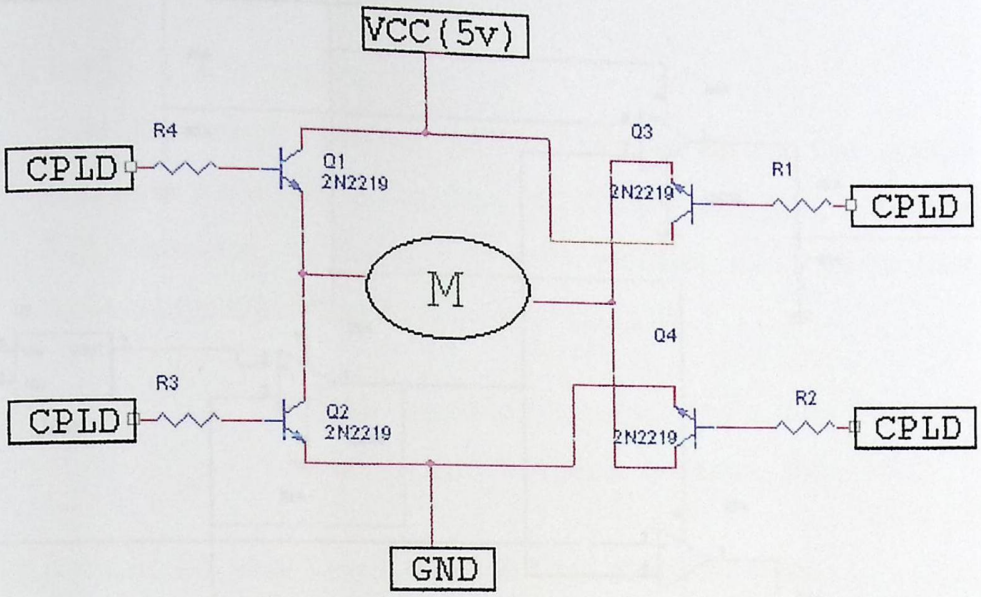


Figure (4.2): H – Bridge Circuit

4.3.3 Fire Detector

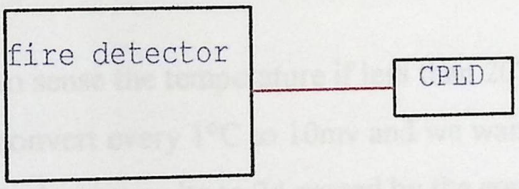


Figure (4.3): Fire Detector

Fire detector work if there is smoke then it will on and send a digital signal to the CPLD, otherwise it will be off and there is no signal.

4.3.4 Temperature Sensor Circuit

The circuit in Figure 4.4 consists:

LM35: convert every 1° C to 10 mV.

Amplifier: voltage gain to 24 passed.

Comparator: to make comparison between 200mV and 250mV.

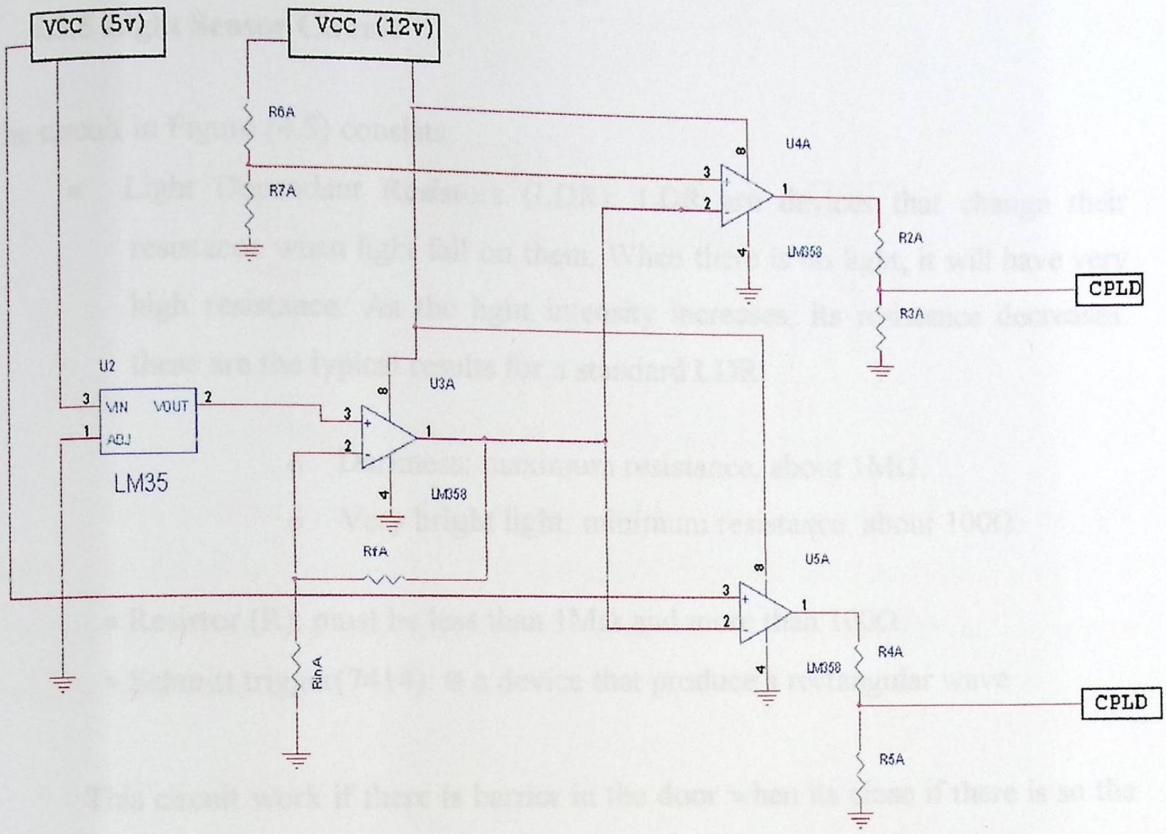


Figure (4.4): Temperature Sensor Circuit

This circuit use to sense the temperature if less than 20° C or more than 25 °C and because the LM35 convert every 1°C to 10mv and we want voltage more than that we put amplifier to enlarging volts to 24 passed by the equation:

$$V_{out}=V_{in}(1+Rs/Rf)$$

So the different between R_s and R_f must be 23 for that:

200mv*24 = 5 volt

250mv*24 = 6 volt

Then we but comparator to make comparison between the voltage that com from amplifier and 12 volt in the first comparator if it is more than 6volt then will out 5 volt from the first comparator if less than 5volt then out 5volt from the second comparator.

4.3.5 Light Sensor Circuit

The circuit in Figure (4.5) consists:

- Light Dependant Resistors (LDR): LDR are devices that change their resistance when light fall on them. When there is no light, it will have very high resistance. As the light intensity increases, its resistance decreases. these are the typical results for a standard LDR:
    - Darkness: maximum resistance, about 1MΩ.
    - Very bright light: minimum resistance, about 100Ω.
- Resistor (R): must be less than 1MΩ and more than 100Ω.
- Schmitt trigger(7414): is a device that produce a rectangular wave

This circuit work if there is barrier in the door when its close if there is so the resistance of LDR must be more than the resistance of resistor and because it connect with Vcc the current will go through it and the Schmitt trigger convert sine wave into crisp square-shaped signal to input this signal to the CPLD ,otherwise the resistance of resistor is more and because it connect with ground there is no signal will appear

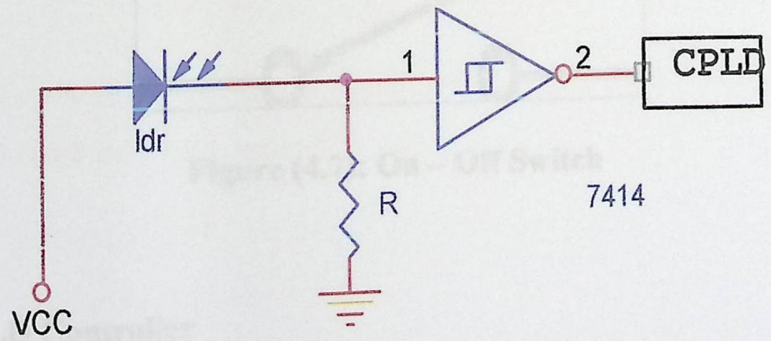


Figure (4.5): Light Sensor Circuit

4.3.7 Light-Emitting Diode

LEDs emit light when an electric current passes through them.

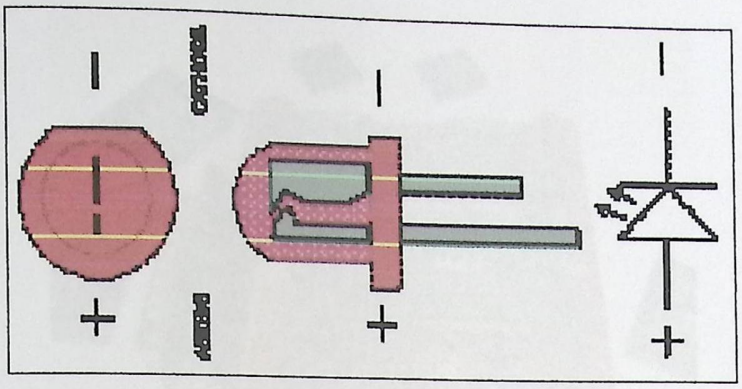


Figure (4.6): LED Polarity and Symbol

LEDs must be connected the correct way round a or + for anode and k or - for cathode as shown in Figure 4.6 . The cathode is the short lead and there may be a slight flat on the body of round LEDs. If you can see inside the LED the cathode is the larger electrode.

4.3.8 ON-OFF Switch

a simple on-off switch. This type can be used to switch the power supply to a circuit. When used with mains electricity this type of switch must be in the live wire.

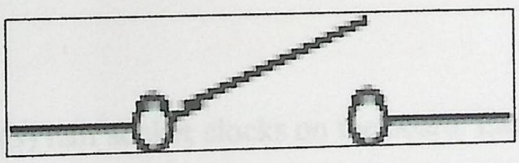


Figure (4.7): On – Off Switch

4.3.9 CPLD Controller

In our project we use aboard of CPLD because its better than usein single ship of CPLD. The board that used in this project is a multi - CPLD design board.

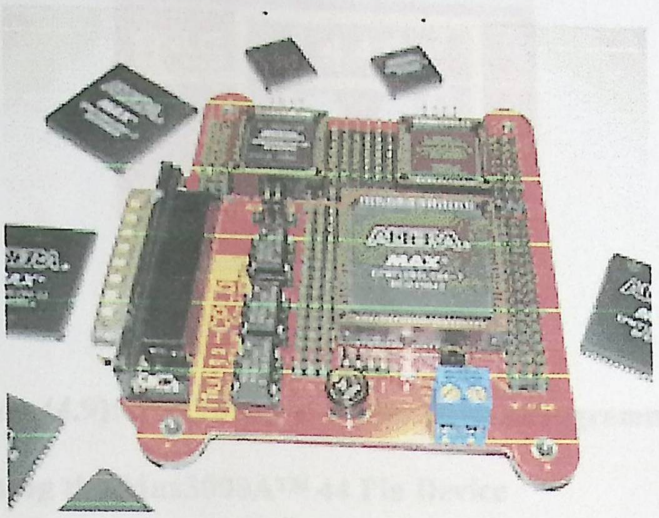


Figure (4.8): CPLD Controller

This Controller consists of three main chips:

- Max 7000S 44
- Max 3000A 44
- Max 7000S 84

Other parts are as followed:

→Clocks

There are three (3) half socket clocks on the board. Each CPLD socket has its own dedicated clock as shown below:

→Programming

- **Programming the Max7000S™ 44 Pin Device**

To program the Max7000S™ 44 pin socket, move the two supplied programming jumpers to the positions as shown below:

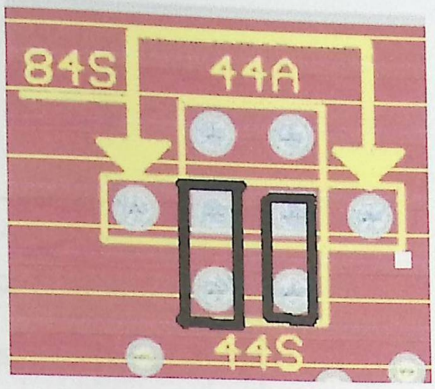


Figure (4.9): Max7000S™ 44 Pin Device Programming

- **Programming the Max3000A™ 44 Pin Device**

To program the Max3000A™ 44 pin socket, move the two supplied programming jumpers to the positions as shown below:

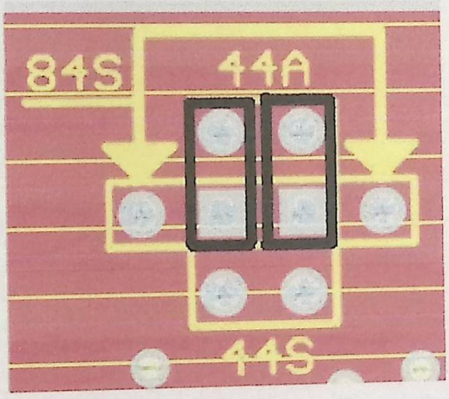


Figure (4.10): Max3000S™ 44 Pin Device Programming

- **Programming the Max7000S™ 84 Pin Device**

To program the Max7000S™ 84 pin socket, move the two supplied programming jumpers to the positions as shown below:

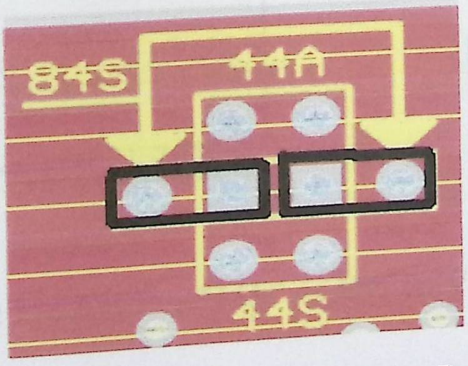


Figure (4.11): Max7000S™ 84 Pin Device Programming

→I/O Pin Locations

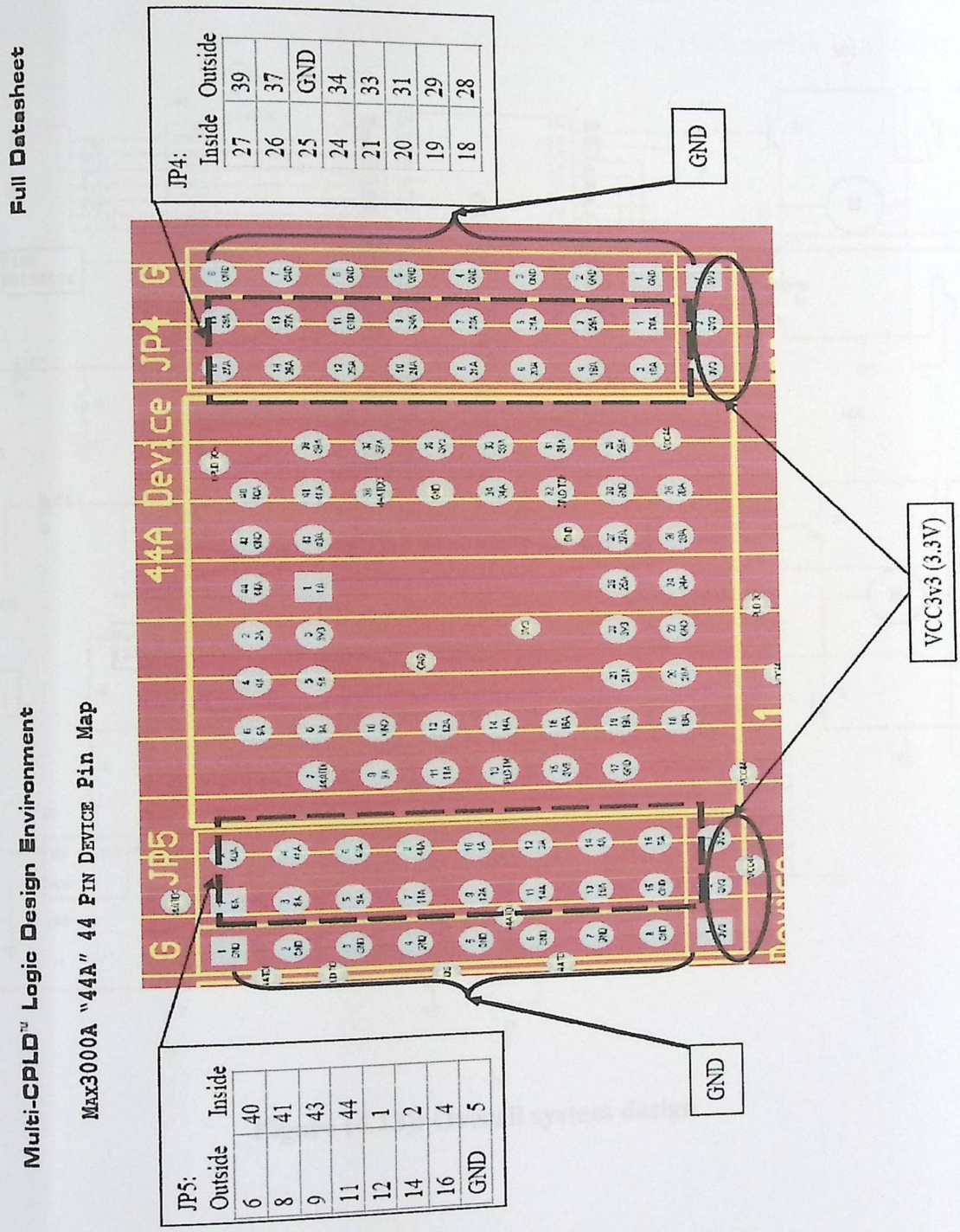


Figure (4.12): MAX3000A "44A" 44 PIN DEVICE Pin Map

For further information return to Appendix A .

4.4 Overall system design

The system design is shown in the following figure:

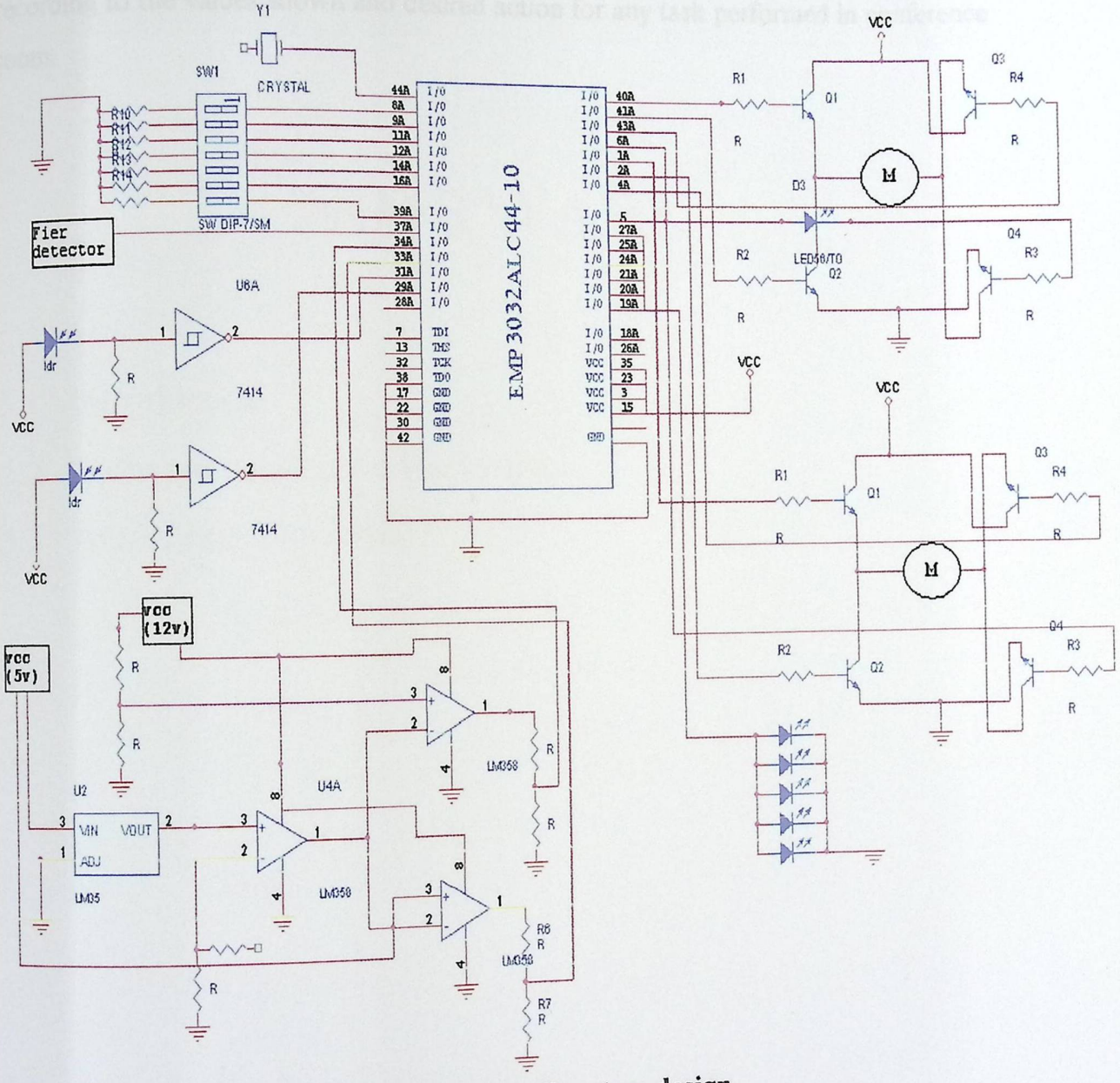


Figure (4.13): Overall system design

4.5 User System Interface

User Interface for this system will be a group of switches and declaration lights connected to the controller, such that the user can give the desired input recording to the values shown and desired action for any task performed in conference room.

Chapter 5

SOFTWARE

5.1 Introduction

5.2 Software needed for the Project.

5.3 Algorithm, Flowchart.

CHAPTER FIVE

SOFTWARE

5.1 Introduction

In this chapter, we are going to describe the software system design which includes an explaining of the programming environment, programming tools, and to describe some of the using methods and algorithm design as followed. And it contains the flowchart for all processes in this project from first step to the final step.

The overall software is programming in Verilogger HDL programming language because it is easier and its syntax similar to the C programming language.

5.2 Software needed for the project

After analyzing all software requirements, the following software functions and modules are needed:

5.2.1 Win ME/NT/2000/XP Compatible Software

This system needs one of these types of windows in order to load the software packages needed for programming, downloading and transferring the programs to the CPLD chip.

5.2.2 Orcad Family Release 9.2

We used this software package to draw the circuits of the project and to draw also the printed boards which we used in the hardware.

5.2.3 SmartDraw 6

We used this software package to draw the flowchart of the project.

5.2.4 Verilogger Pro

We need this software package to write the whole HDL program so as to make it ready and tested by simulating it at the same package and check the timing diagrams to prepare it for the other software package which delivers it to the CPLD chip.

The steps of writing the program, simulating it and checking timing diagrams are shown in appendix B.

5.2.5 ALTERA QUARTUS II

We need this software package to transfer the HDL program that we have already written and tested by the Verilogger Pro program which delivers it to the CPLD chip.

5.3 Algorithms and Flow Charts

In this section we will explain the algorithms and methods that will be needed to design our project. And show the flow chart if it needed.

5.3.1 Algorithms

There are many modeling technique can be use to describe a module:

- **Gate-Level modeling**: using **instantiation** of primitive gates and user-defined module.
- **Dataflow modeling**: sing continues assignment statements with keyword **assign**
- **Behavioral modeling**: Using procedural statements with the keyword always

behavioral modeling : is the best modeling can describe a module which Behavioral Modeling mostly used for sequential circuit (CPLD)

Target output: The target output of a procedural statement must be of **reg** data type, **reg** data type remains unchanged until a new value is assigned by procedural assignment so the output must define as reg.

always keyword : Behavioral, descriptions use the keyword always followed by a list of procedural (behavior) statements. An always statements constitute an always block. The always statement always starts at time 0 and execute statement in the block continuously (looping).

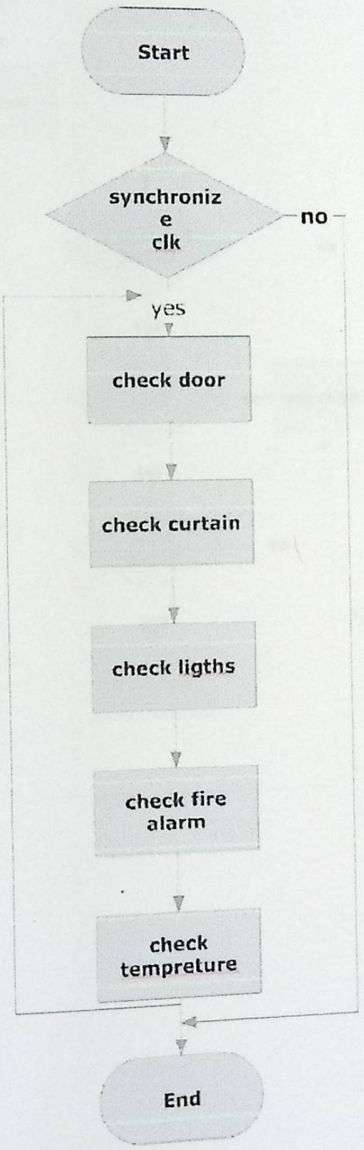
Inside always block there are many operations executed:

- read the value from switches to execute the operation according of the value:
  - If sw0=1 and sw1=0 then check if the door not open (r1=0) if not open the door must open(p= 4'b0011) if it was opened (r1=1) then the door stop moving (p= 4'b0000) .
  - If sw0=0 and sw1=1 then check if door not close (r2=0) if not then check the state of photodiode (phd) if phd=0 then close the door (p= 4'b1100)If phd=1 then the door stop move(p= 4'b0000). If the door was closed (r2=1) then stop moving door (p= 4'b0000) .
  - If sw2=1 and sw3=0 then check if the curtain not open (r3=0) if not the door must open(n= 4'b0011) if it was opened (r3=1) then curtain stop moving (n= 4'b0000)
  - If sw2=0 and sw3=1 then check if curtain not close (r4=0) if not close then close the curtain (n= 4'b1100) If the curtain was closed (r4=1) then the curtain stop move (n= 4'b0000).
  - If sw4=0 then the lights will open and close according of phd if phd = 0 then the low light will be open (L1=1) else if phd=1 high lights will open (L2=1, L1=1)
  - If sw4=1 then lights will open and close manually according of there switches.
- read the value from fire sensor if this value equal 1 (fs=1) then the door and the curtain must open(p= 4'b0011, n= 4'b0011) and the alarm must tern on (alarm=1) .

- read the value from temperature sensor if this value less than 20 then open the heat (heat=1) else if this value more than 25 then the air-conditions must on (cool =1)

5.3.2 Flow chart

this section talks about the flowchart of the system phases



Figure(5.1): general flow chart

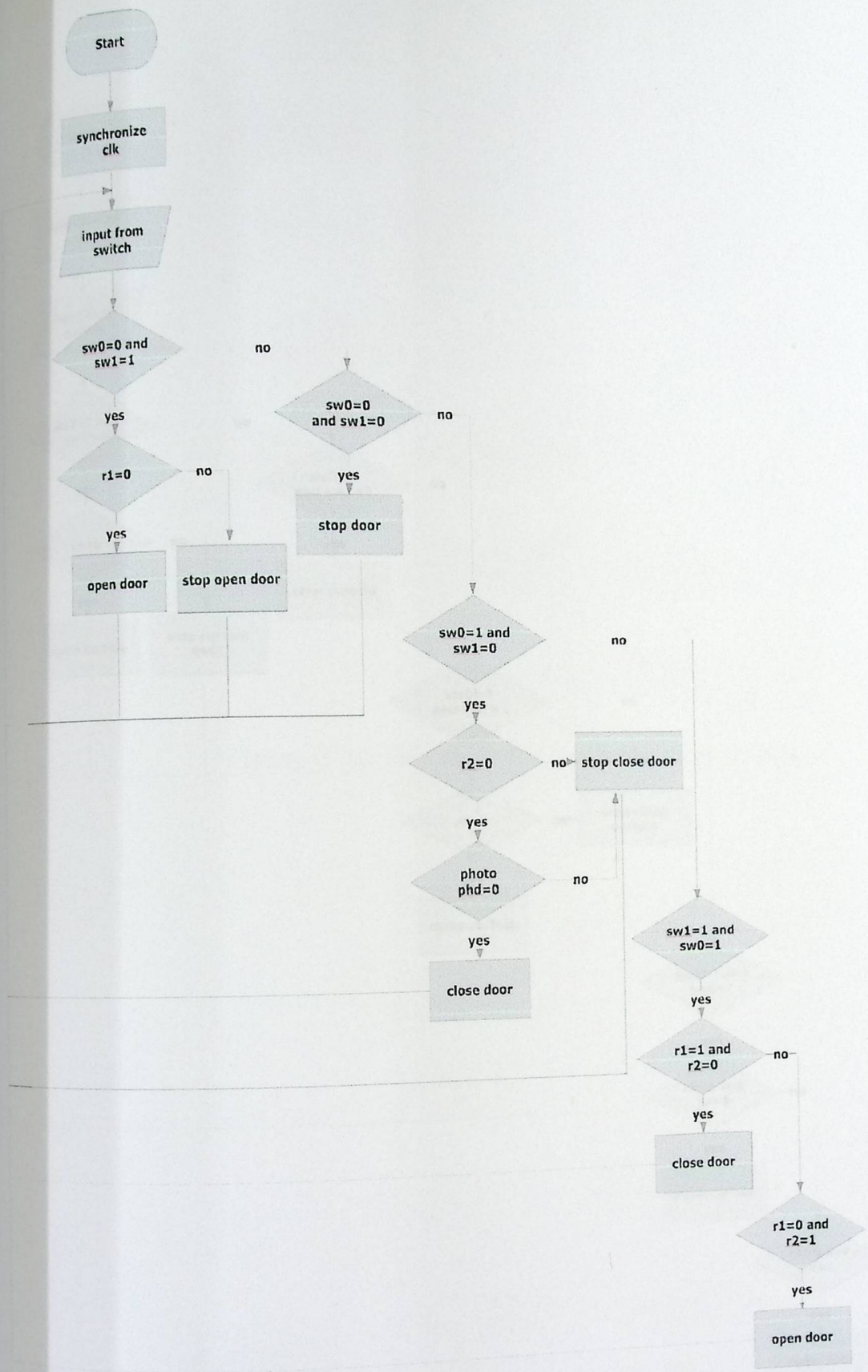


Figure (5.2): flow chart to open or close door

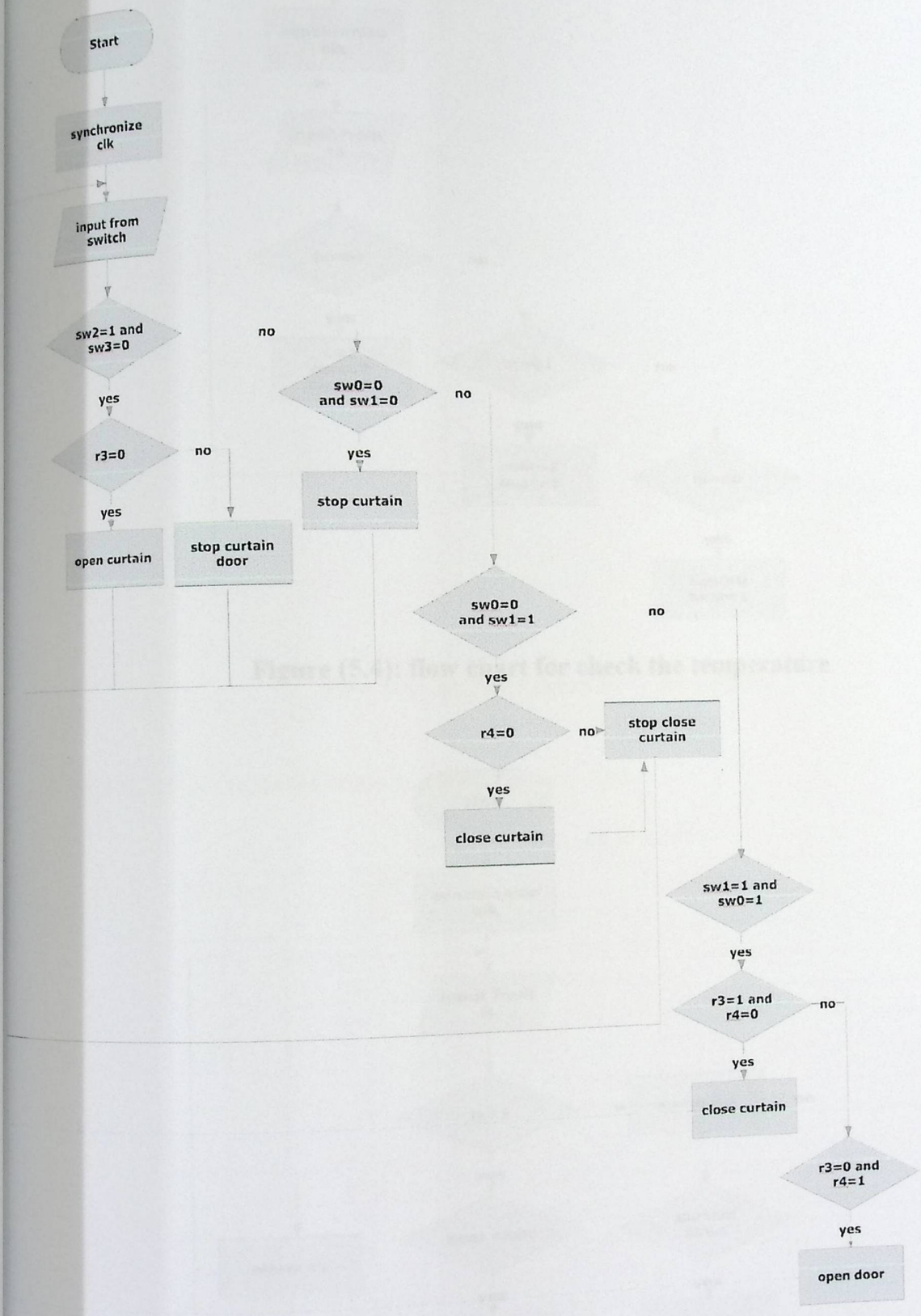


Figure (5.3): flow chart to open or close curtain

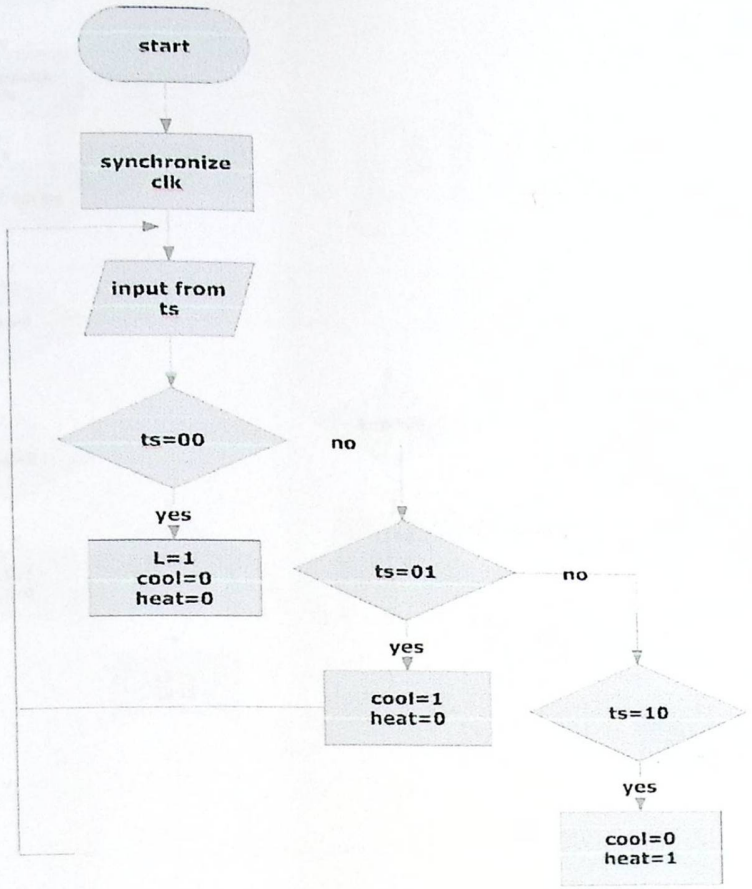


Figure (5.4): flow chart for check the temperature

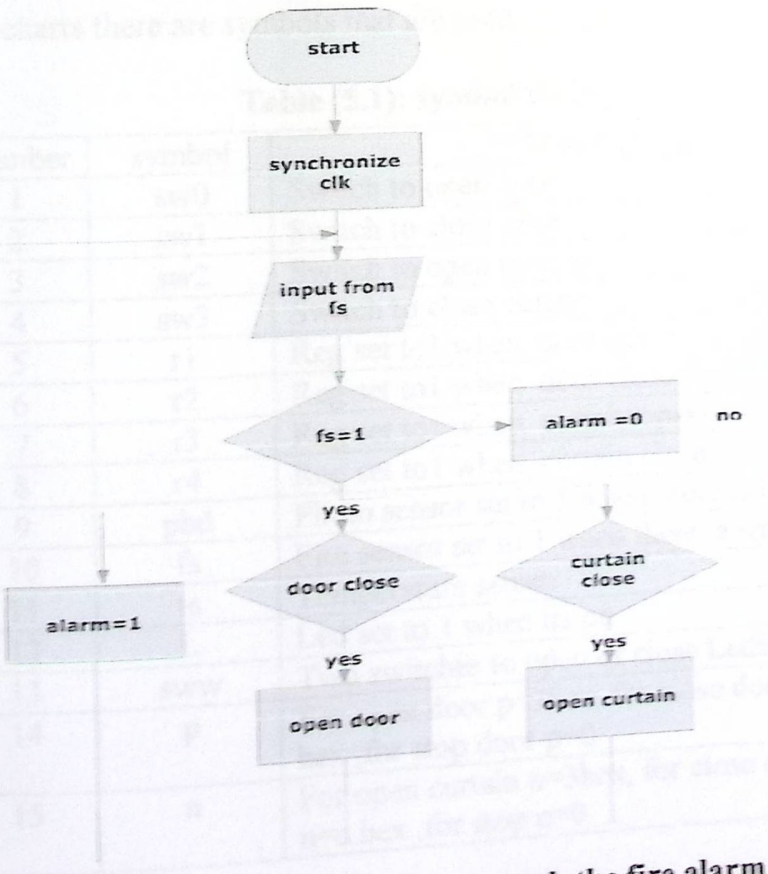


Figure (5.5): flow chart for check the fire alarm

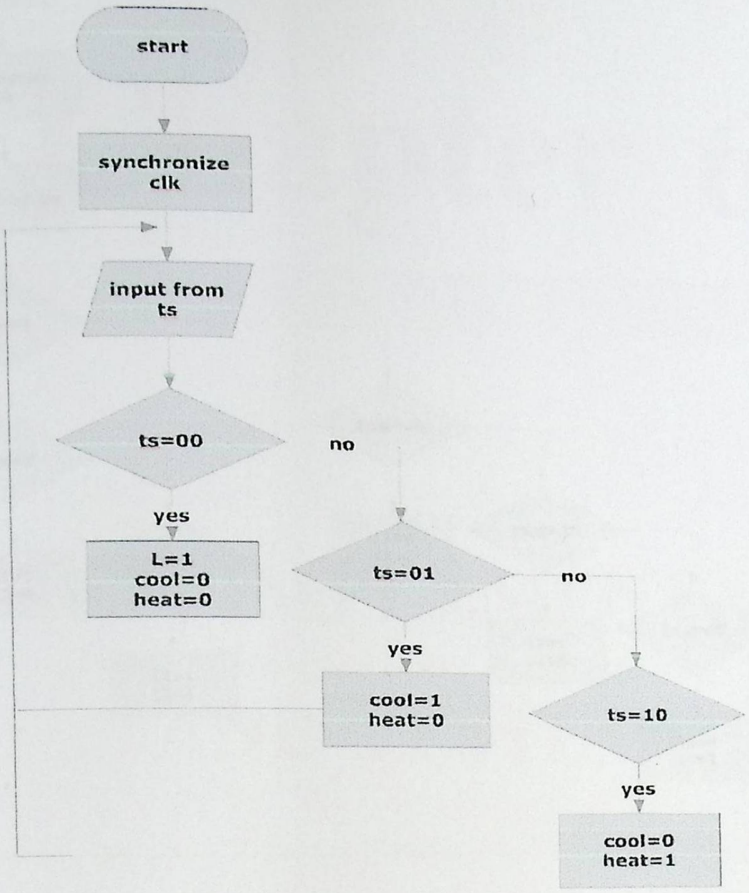


Figure (5.4): flow chart for check the temperature

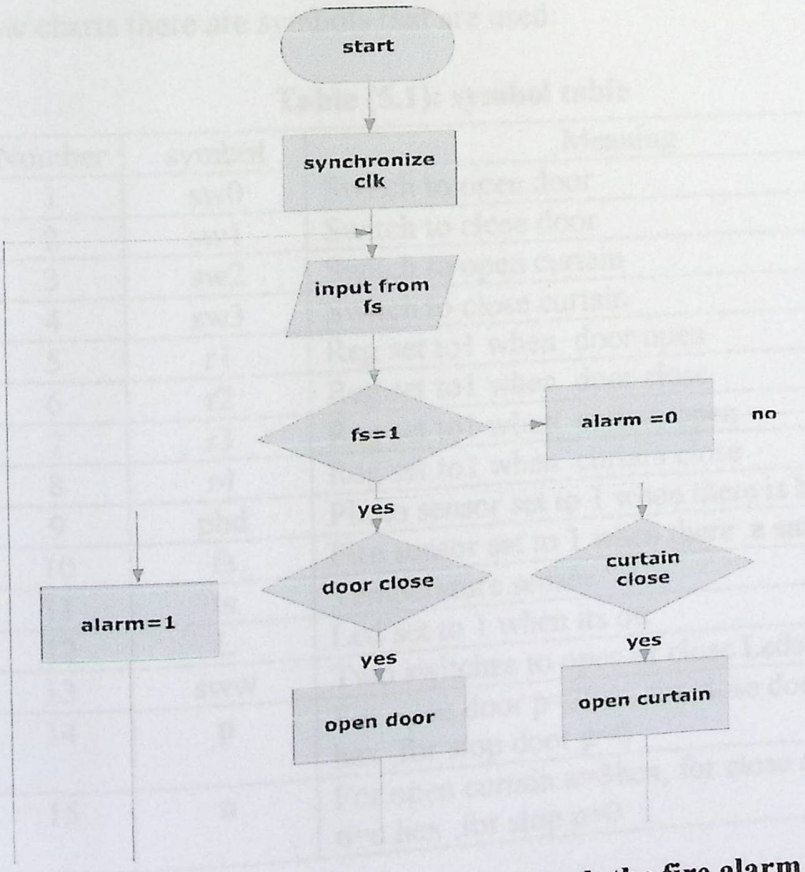


Figure (5.5): flow chart for check the fire alarm

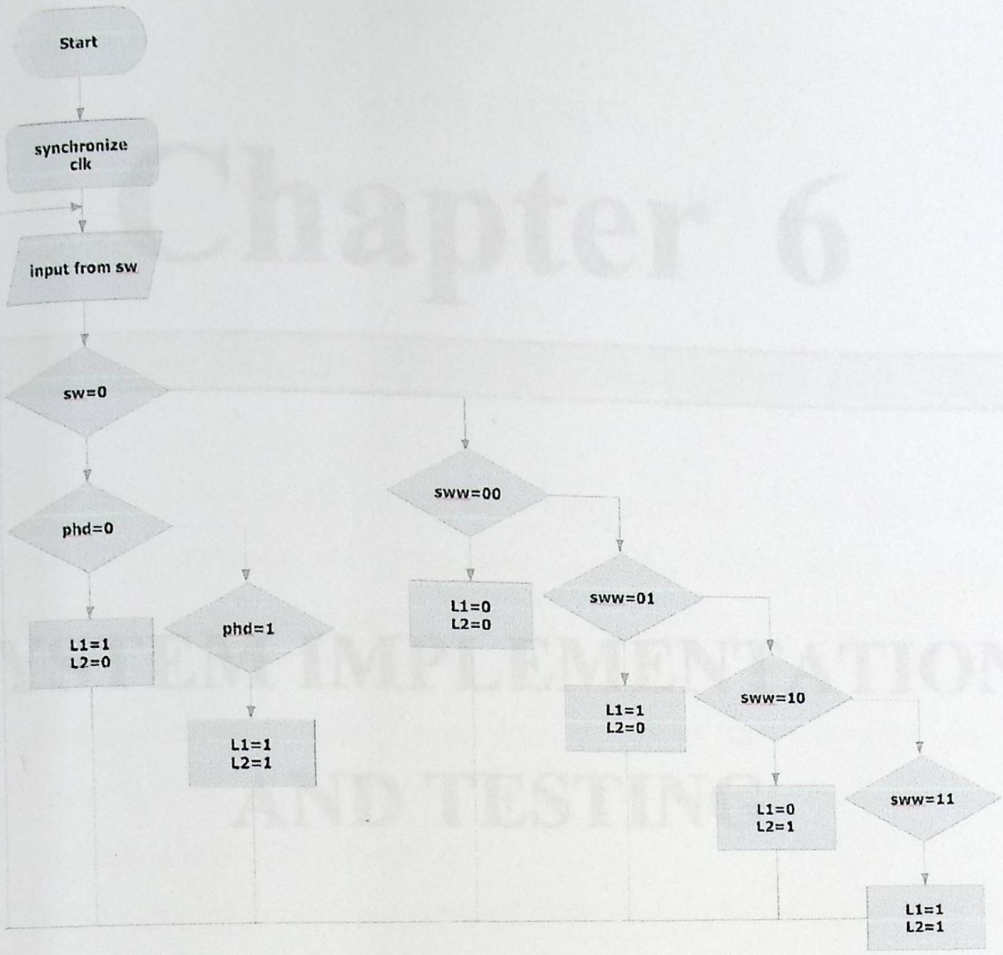


Figure (5.6): flow chart for check lights

For the flow charts there are symbols that are used:

Table (5.1): symbol table

Number	symbol	Meaning
1	sw0	Switch to open door
2	sw1	Switch to close door
3	sw2	Switch to open curtain
4	sw3	Switch to close curtain
5	r1	Reg set to1 when door open
6	r2	Reg set to1 when door close
7	r3	Reg set to1 when curtain open
8	r4	Reg set to1 when curtain close
9	phd	Photo sensor set to 1 when there is barrier
10	fs	Fire sensor set to 1 when there a smoke
11	ts	Temperature sensor
12	L	Led set to 1 when its on
13	sww	Two switches to open or close Leds
14	p	For open door p=3hex, for close door p=c hex ,for stop door p=0
15	n	For open curtain n=3hex, for close curtain n=c hex ,for stop n=0

Chapter 6

SYSTEM IMPLEMENTATION AND TESTING

6.1 Introduction

6.2 Implementation

6.3 Testing

CHAPTER SIX
Implementation and Testing

6.1 Introduction

This chapter shows the implementation and testing processes for our application, and demonstrates the methods and procedures used for testing and examining the application operation and behavior.

The implementation and testing was done by using the following tools and components:

- Connectors with different colors.
- All the components that are depicted in the design chapter (see chapter 4).
- A wire grabber and a wire cutter.
- Soldering Iron.
- A digital Millimeter, Power Supply, and Oscilloscope for testing.

This system has more than one issue to be tested. Some testing parts reflect software and hardware.

6.2 Implementation

The implementation process is done synchronized with the testing operation, because each implementation phase will take many tests to ensure that there are no errors.

6.3 testing

In this section we will demonstrate the testing of each subsystem separately.

6.3.1 Testing Software by Simulation Program

After writing the project program using Verilog pro Programming Package we made compilation and simulation to check that the program is working well and make sure to see the timing diagrams as shown in Figures 6.1 to 6.5 below:

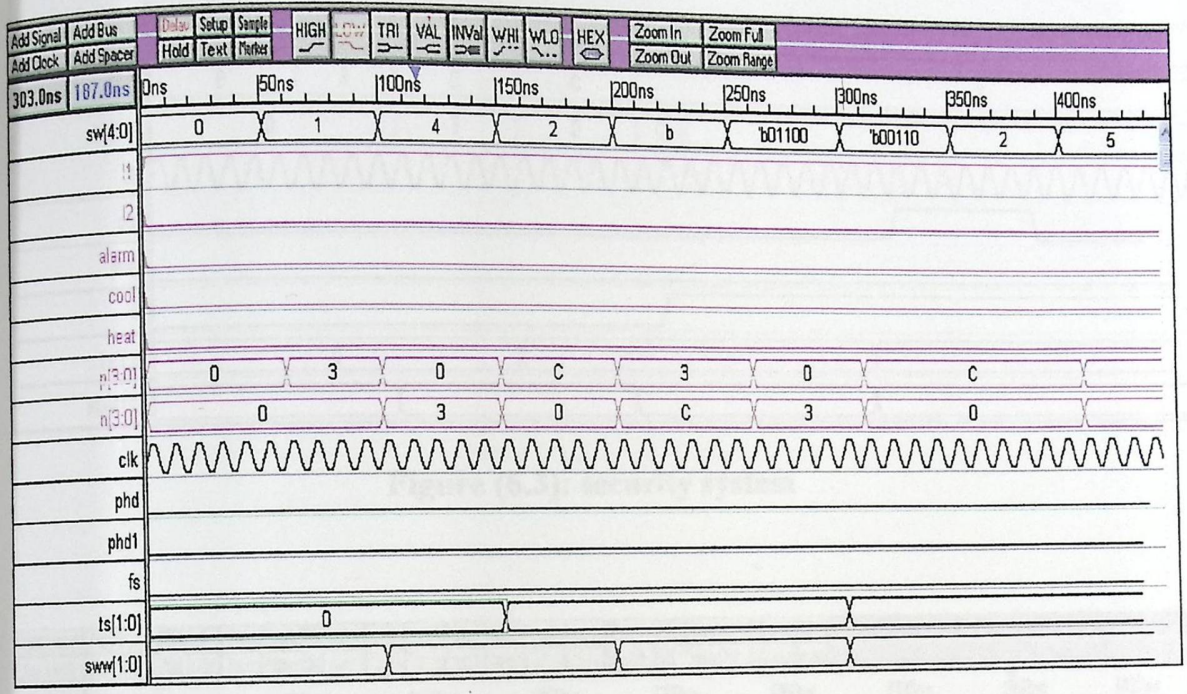


Figure (6.1): Timing Diagram for open or close door and curtain

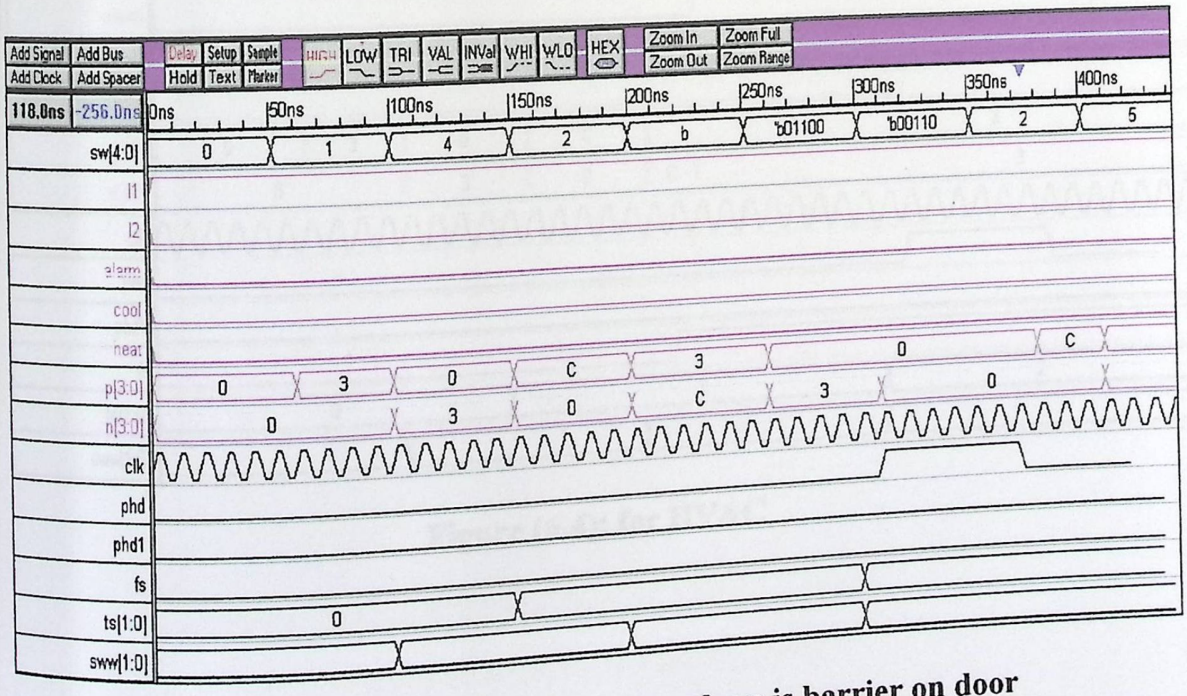


Figure (6.2): Timing Diagram when there is barrier on door

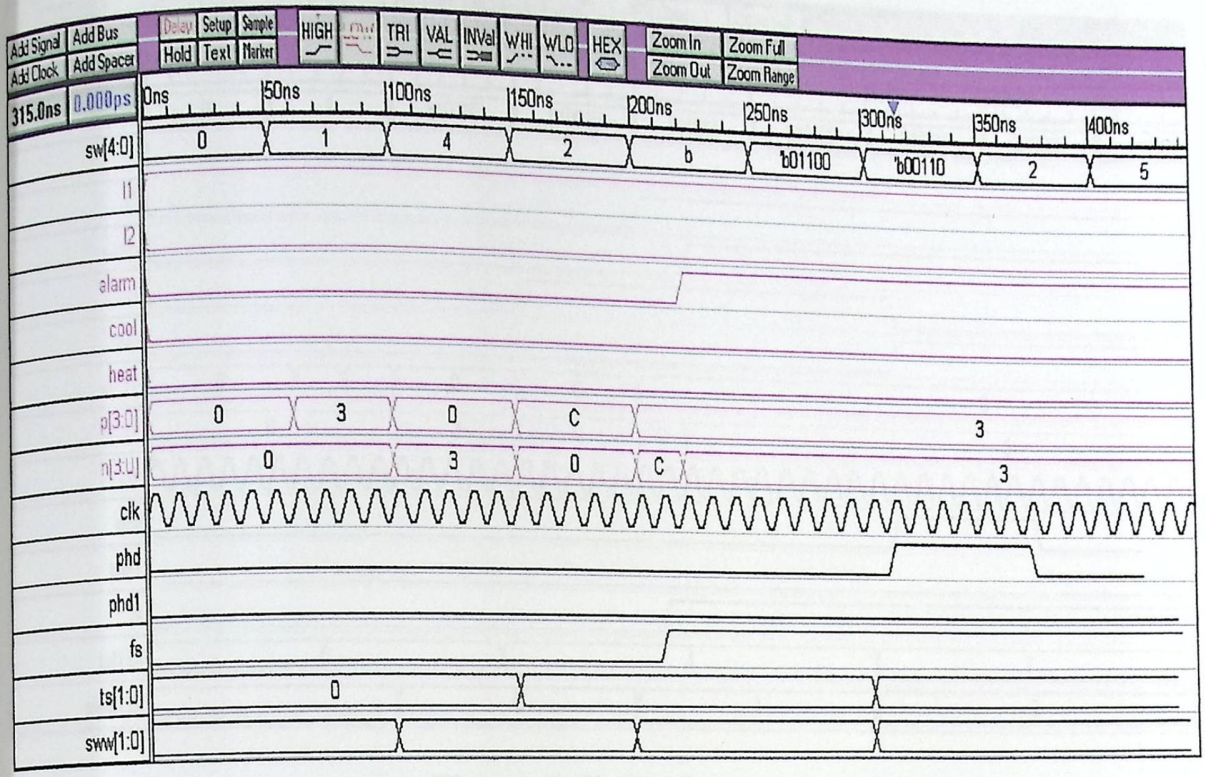


Figure (6.3): security system

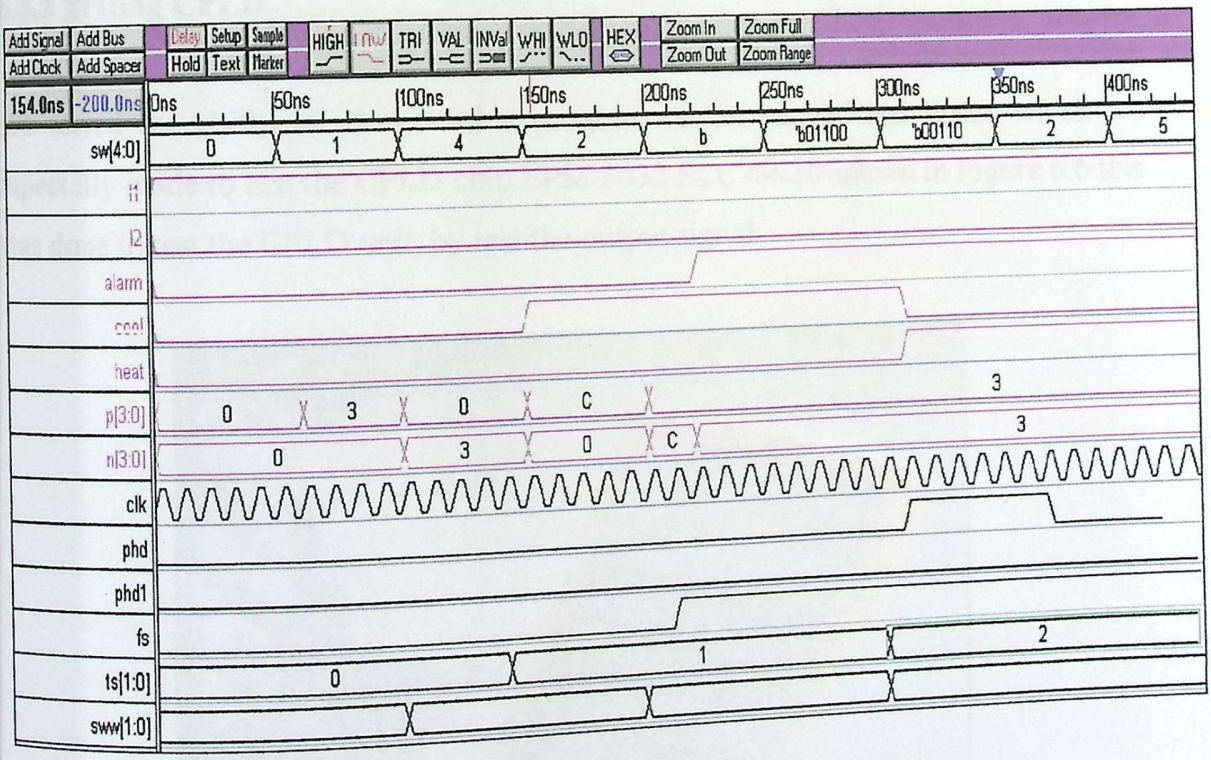


Figure (6.4): for HVAC

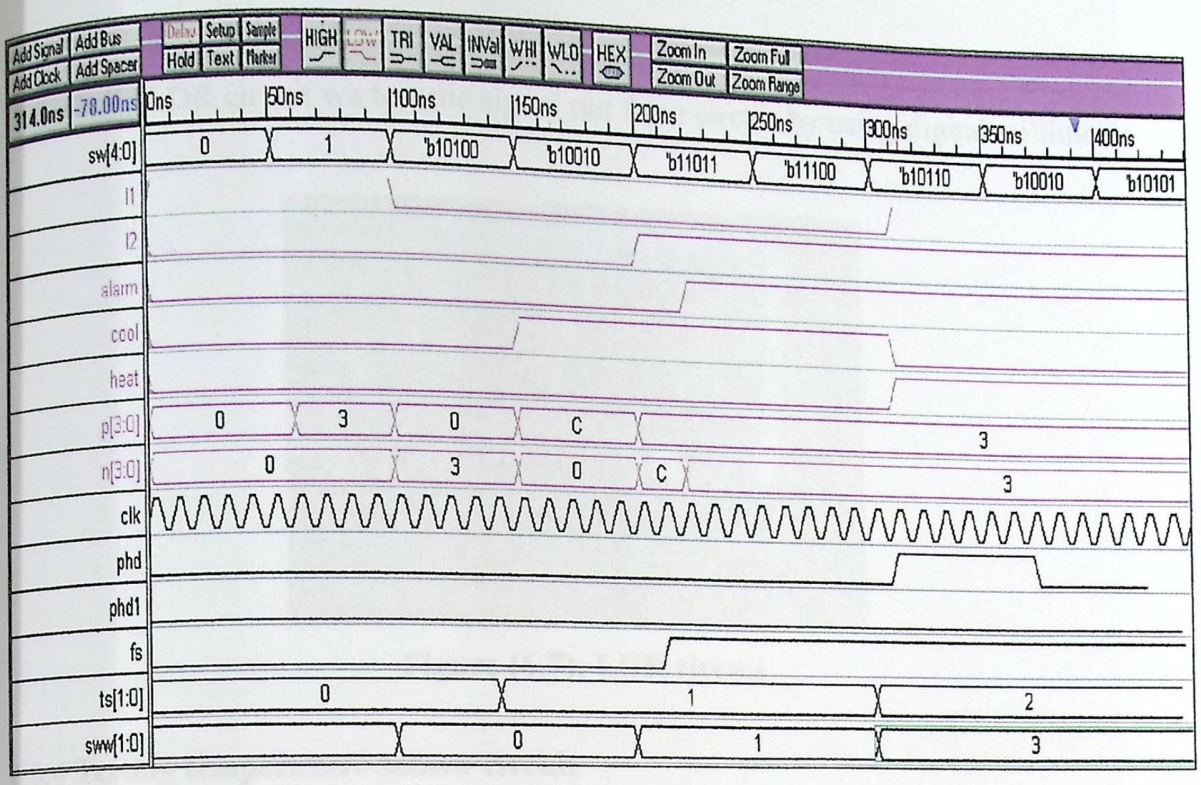


Figure (6.5): for light

6.3.2 Testing CPLD

We tested the ability to download a program on an educational board which we especially made to test the CPLD chip EPM 3032 SLC 44-10 shown in Figure 6.6 this was done to test the CPLD unit by test the output signal.

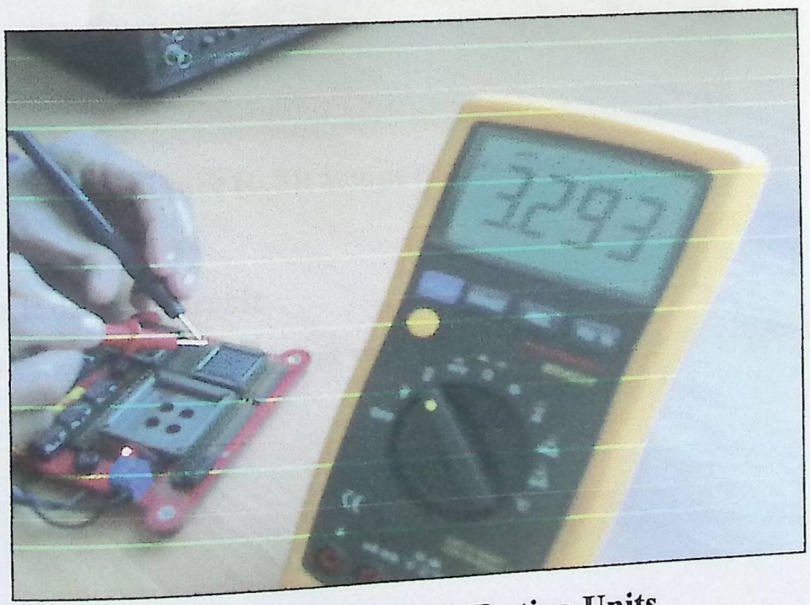


Figure (6.6): CPLD Testing Units

6.3.3 Testing LDR circuit

After build LDR circuit we test the signal out from circuit by using digital millimeter.

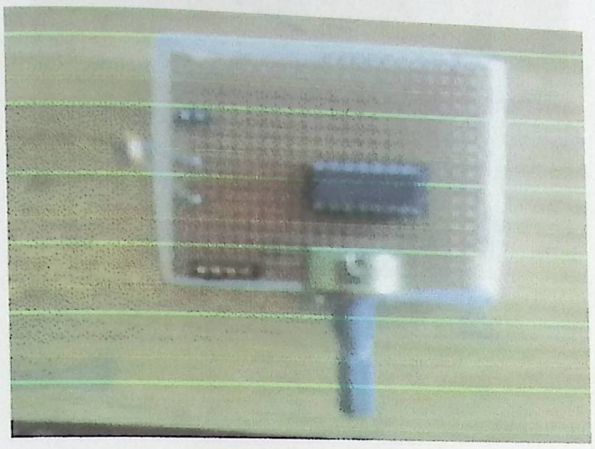


Figure (6.7): LDR circuit

6.3.4 Testing temperature sensor circuit

After build temperature sensor circuit as shown in figure 6.8 we test the signal out from circuit by using digital millimeter

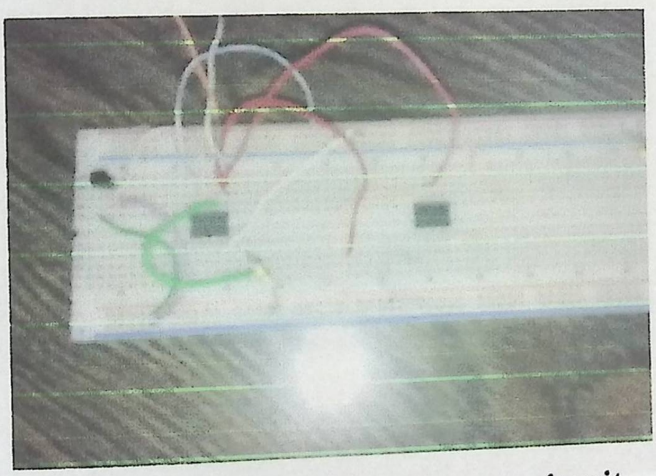


Figure (6.8): temperature sensor circuit

6.3.5 Testing H-Bridge circuit

In figure 6.9 shown H-Bridge circuit we test the motor when turn right or turn left.

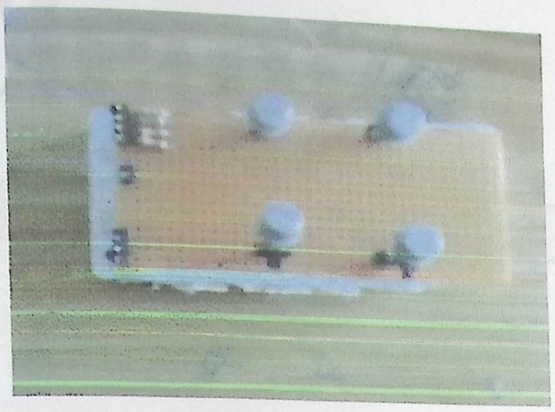


Figure (6.9): H-Bridge circuit

6.3.6 Model implementation

Model of conference room implemented to be easy to use and clear as shown in figure 6.9.

Figure(6.10): conference room model

Chapter 7

CONCLUSIONS AND FUTURE WORK

7.1 Introduction

7.2 Conclusions

7.3 Problems

7.4 Suggestions and Future Provision

Chapter Seven

Conclusion and Future Work

7.1 Introduction

This chapter represents the conclusions extracted during designing and implementing and illustrates the system implementation achievements and output.

7.2 Conclusions

After the team has finished the design and implementation of the project and integrated overall system. The team concludes that :

- There is an ability to control different application on conference room by using a new technology called CPLD and Verilog HDL language.
- using CPLD technology for control is more efficient because its low power consumption it has high speed with fast pin-to-pin delays its nonvolatile and the program is stile save until if the power off .

7.3 Problems

Here are some problems faced the project team during the system implementation:

7.3.1 Hardware Problems

- One of the problems that we faced during the project is whether the model which containing phases of the project wood or plastic to be clearly shown and easy-to-use and for maintenance when needed, so we settled on the wooden model with glass for its light weight and easy to deal with .

- Another problem we faced is to find the board of CPLD in our country because it is not available in our shops, first want to have board with EMP7128SLC84-7 CPLD but we not found it we replaced by board with EMP3032ALC44-10 and it performed the task.

- There were other problems in terms of electrical connections where errors discovered and develop appropriate solution in a timely manner.

After a lot of work the trouble was overcame and all the problems have been settled down, the final project worked in an excellent manner and was tested successfully.

7.3.2 Software Problems

The main problem we faced that the software was new for us so it took us a lot of time to study this software and making the program several time, improving it until we settled on the final program which we used. As mentioned earlier we had to use several packages to program the chip we faced a lot of troubles when we dealt with these programs, but we passed these problem successfully.

7.4 Suggestions and Future Provision

After the completion of this project and the successful work we have seen that there is a possibility to develop and modify it in several ways to be a proposal for other projects that possible to implement using the same technology, including:

1-Using wireless to connect application with CPLD to avoid any problems that might happen from wire.
2- Control of more application in conference room like controlling audio or video.
3- Using CPLD technique in industry filed since it is save time, power and cost
4- Installation this project and benefit from it at a real conference room on city.

References

[1]: Ismail Mohd Ibsais , Waddah I. Sultan , Musa Zugaier , Designing Traffic Lights Using CPLD & HDL Technology, Palestine Polytechnic University, 2007.

[2]: http:// www.gwe.edu /~researech/accele.htm.

[3]: M. Morris Mano, DIGITAL DESIGN, Prentice Hall, New Jersey, USA, 2002, http://www.sysf.physto.se/kurser/digsyst/lekt2.pdf

[4]:Zainalabedin Navabi, Digital Design and Implementation with Field Programmable Devices, Kluwer Academic Publishers, Massachusetts, USA, 2005.

[5]: M. Morris Mano, DIGITAL DESIGN, Prentice Hall, New Jersey, USA, 2002, http://www.eng.auburn.edu/~strouce/class/elec4200/PALsandPLDs.pdf

[6]: http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.pdf

[7]:http://www.ece.unm.edu/vhdl/Labs2006/spring06/lab05/lecture_notes_lab05_v71sp3.pdf]

[8]: http://www.altera.com/products/devices/max7k/features/m7k-features.html

[9]: http://en.wikipedia.org/wiki/DC_motor

[10]: http://en.wikipedia.org/wiki/Smoke_detector

[11]: http://www.national.com/ds/LM/LM35.pdf

[12]: http://en.wikipedia.org/wiki/Photoresistor

[13]: http://mazsola.iit.uni-miskolc.hu/cae/docs/xabel.html

[14]: Samir Palnitkar, Verilog HDL A guide to Digital Design and Synthesis, Sun Soft Press, 1996.

[15]: http://www.cl.cam.ac.uk/~mjcg/Verilog/index.html

[16]: http://www.sutherland-hdl.com/online_verilog_ref_guide/vlog_ref_top.html

APPENDECES

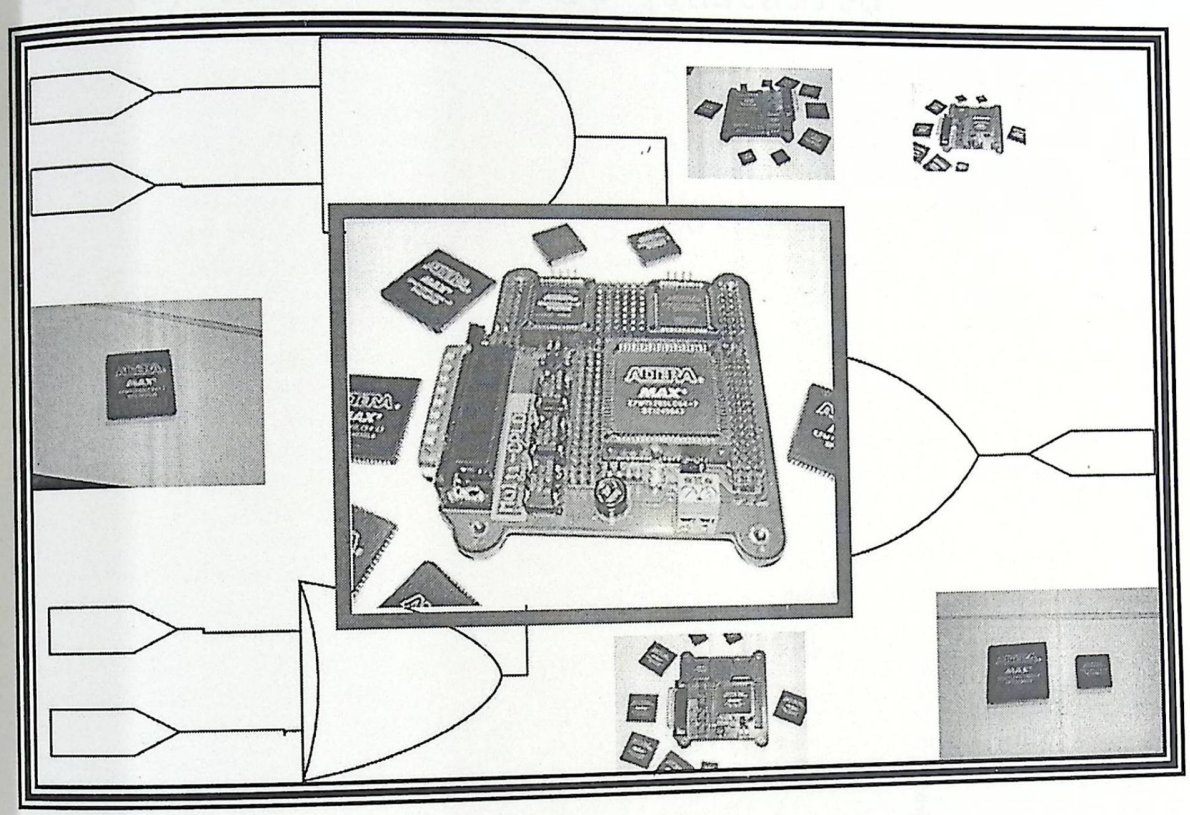
APPENDIX A

DATA SHEET

Maximum Digital Designs, LLC
Presents:

Multi-CPLD™

Logic Design Environment



Supports ALTERA MAX7000S and MAX3000A PLD Devices

(+ other pin compatible industry standard CPLDs/SPLDs)

1 Standard Package Contents:

1. (1) ***Multi-CPLD***™ Design Board
2. (3) 25.175 MHz Clocks (Installed)
3. (1) EPM3032ALC44-10 (Installed)

1 Standard Package Contents:

1. (1) ***Multi-CPLD***™ Design Board
2. (3) 25.175 MHz Clocks (Installed)
3. (1) EPM3032ALC44-10 (Installed)

2 Disclaimer:

Maximum Digital Designs, LLC reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

Maximum Digital Designs, LLC provides the information and data included in this document for your benefit. However, it is not possible to entirely verify and test all the information, in all circumstances, particularly information relating to products manufactured by those other than ***Maximum Digital Designs, LLC***. ***Maximum Digital Designs, LLC*** makes neither warranty nor representation relating to the quality, content, or adequacy of this information. Although every effort has been made to ensure the accuracy of this manual, *Maximum Digital Designs, LLC* shall not be liable for any errors, incidental or consequential damages in connection with the furnishing, performance, or use of ***Maximum Digital Designs, LLC*** products. ***Maximum Digital Designs, LLC*** assumes no responsibility for damage or loss resulting from the use of this manual, for loss or claims by third parties, which may arise through the use of the ***Multi-CPLD™***, or for any damage or loss caused by malfunction.

Customers assume all responsibility in using the ***Multi-CPLD™***. ***Maximum Digital Designs, LLC*** provides a thirty (30) day money back satisfaction guarantee less shipping if product is in resellable condition (to be determined by ***Maximum Digital Designs, LLC)***. All ***Maximum Digital Designs, LLC*** products carry a ninety day (90) parts and labor warranty on defective parts and assembly (excluding logic devices). ***Maximum Digital Designs, LLC*** is not responsible for damaged products that are the result of improper handling, installation, or operation. The customer must handle, install, and operate the device within the guidelines outlined within the operating manual.

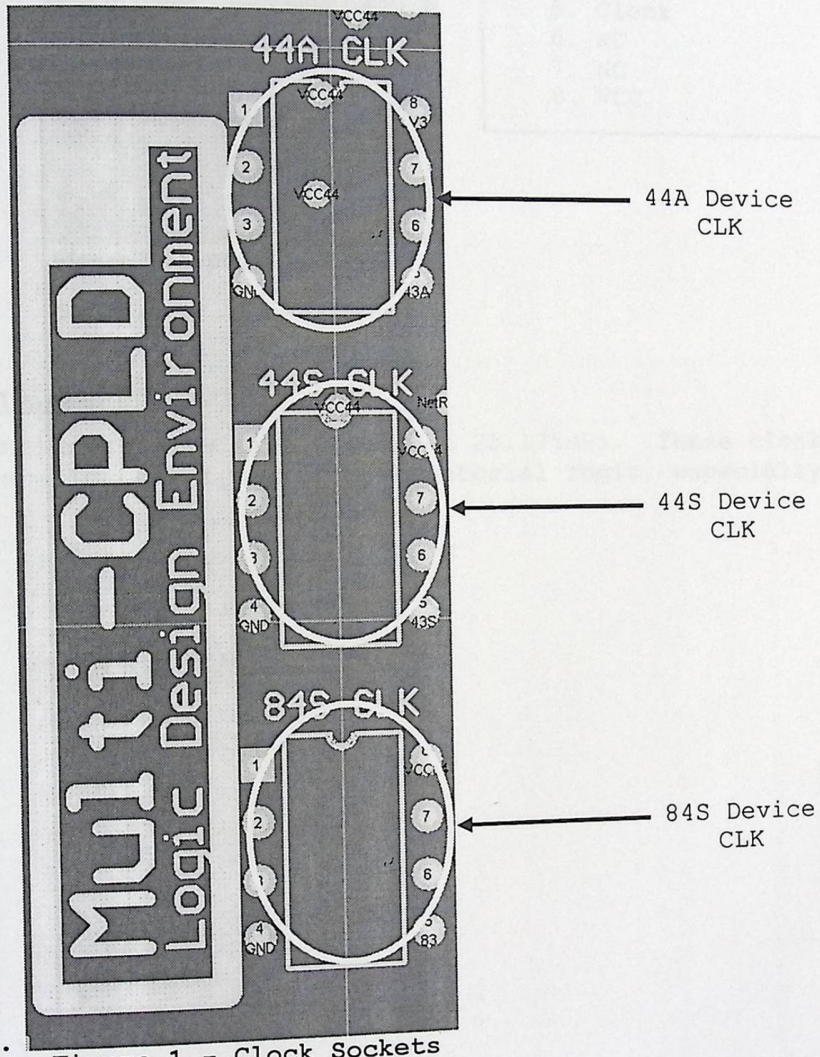
3 Intended Use:

This product is intended for non-commercial applications and is not INTENDED FOR COMMERCIAL APPLICATIONS. ALL PARTIES USING THIS PRODUCT FOR NON-EDUCATIONAL APPLICATIONS ASSUME ALL RESPONSIBILITES. THIS PRODUCT IS INTENTED FOR EDUCATIONAL USE ONLY WITH NO VIABLE COMMERCIAL USE.

4 Clocks

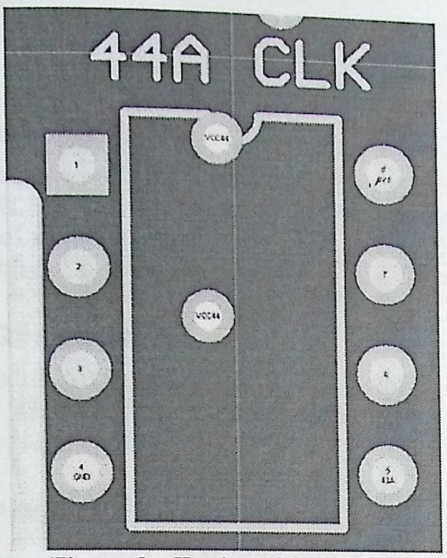
4.1 ½ Size Socket

There are three (3) half socket clocks on the board. Each CPLD socket has its own dedicated clock as shown below:



a. Figure 1 - Clock Sockets

b.



Typical Clock Pinout:

1. No Connect (NC)
2. NC
3. NC
4. GND
5. Clock
6. NC
7. NC
8. VCC

Figure 2 - Typical Clock Pinout

4.2 Included Clocks

The *Multi-CPLD*™ includes (3) ½ size clocks at 25.175MHz. These clocks are ideal for generating all types of combinatorial logic, especially video signals.

5 Grounds

5.1 Ground Planes

The *Multi-CPLD*™ features full ground plane copper floods on both side of the two layer PCB. The copper floods provide close reference planes for adjacent signals. Further, as the signals are closely coupled to a reference plane, they provide jitter free timing edges.

5.2 Available Grounds

All signal pins have adjacent rows of ground pins. These are provided for probe hook-up for various signal devices including oscilloscopes and logic state analyzers.

5.2.1 Markings

Ground signal columns are marked with a 'G' header (circled, red dashes; see figure below). In addition, the four stand-offs and the DB-25 mounting holes are also all tied to the ground plane.

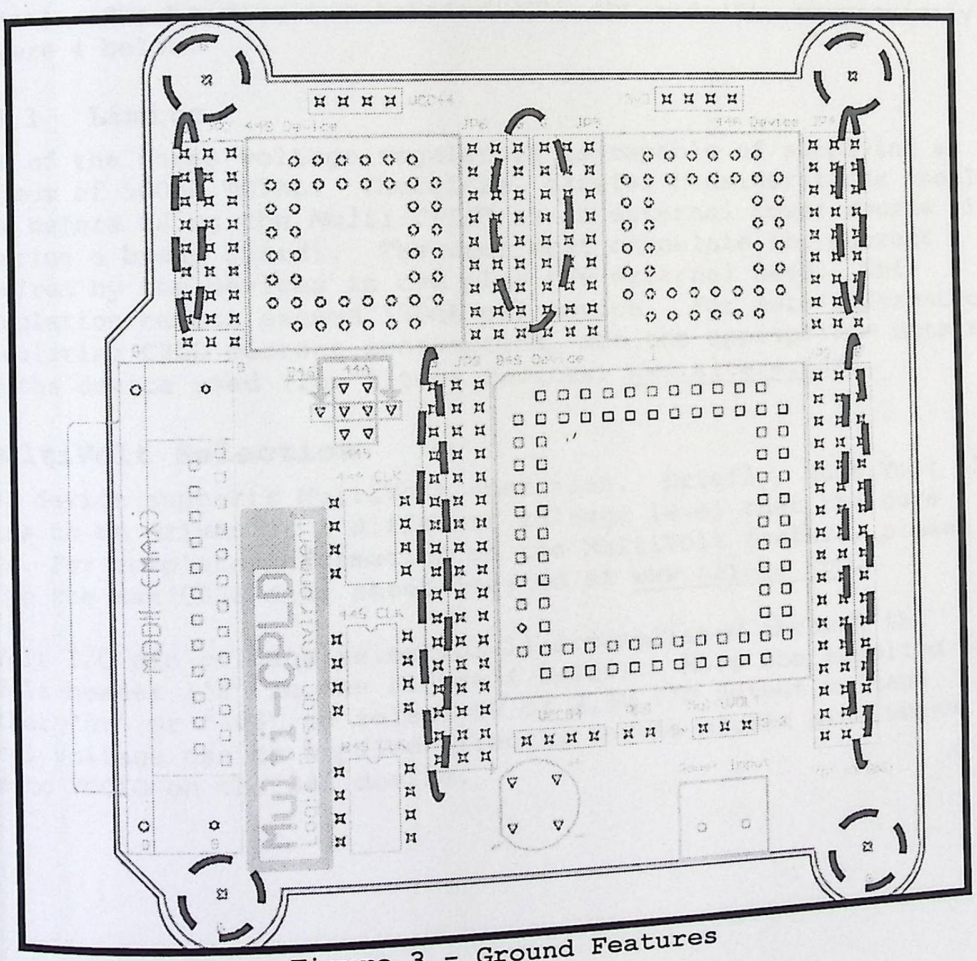


Figure 3 - Ground Features

6 Power

6.1 Board requirements

The *Multi-CPLD*™ can accept any standard DC wall transformer that supplies at least 500mA per socket that is used. Further, 1500mA is strongly recommended to power all three sockets simultaneously. Recommended input voltage should be between 7V and 15V (20V absolute maximum). Power should be supplied through the blue terminal block marked "Power Input" ('A' in the Figure 4 below).

6.2 Rectification

The Multi-CPLD™ power input is fully rectified for convenience. Therefore, polarity need not be observed ('B' in the Figure 4 below).

6.3 Take Off Power

Headers supply power taken from the three voltage regulators. The headers are labeled 'VCC44' (supplying 5V taken from the 44S device voltage regulator), '3v3' (supplying 3.3V from the 44A device voltage regulator) and 'VCC84' (supplying 5V taken from the 84S device voltage regulator). The headers are labeled 'C', 'D' and 'E', respectively, in the Figure 4 below.

6.3.1 Limits

Each of the three voltage regulators is capable of supplying a maximum of 500mA TOTAL. Therefore, careful considerations should be made before using the *Multi-CPLD*™ as an external power source (i.e. powering a bread board). The user must calculate the current required by the devices in use plus the external need. This calculation cannot exceed 500mA per device. For more information on calculating CPLD current consumption, see the appropriate data sheet for the device used (for Altera devices, www.altera.com).

6.4 MultiVolt Selection

The 84S device supports MultiVolt operation. Briefly, MultiVolt allows I/O pins to be driven at a different voltage level that the core voltage. For complete information on the MultiVolt feature, please refer to the Max7000S data sheet located at www.altera.com.

MultiVolt I/O pin voltage selection is accomplished through the MultiVolt header ('F' in the Figure 4 below). Move the supplied jumper to either '5v' or '3v3' to select 5V or 3.3V I/O output voltage. External voltage can be applied directly to the middle pin, which is routed to VCCIO on the 84S device.

6.5 Available Current per Socket

Each socket (and thus voltage regulator) is capable of supplying 500mA per device. This absolute current rating must include device draw AND take off needs. For more information on calculating CPLD current consumption, see the appropriate data sheet for the device used (for Altera devices, www.altera.com).

6.6 Power LED

A power LED is provided to indicate that a valid power source is connected ('G' in the Figure 4 below).

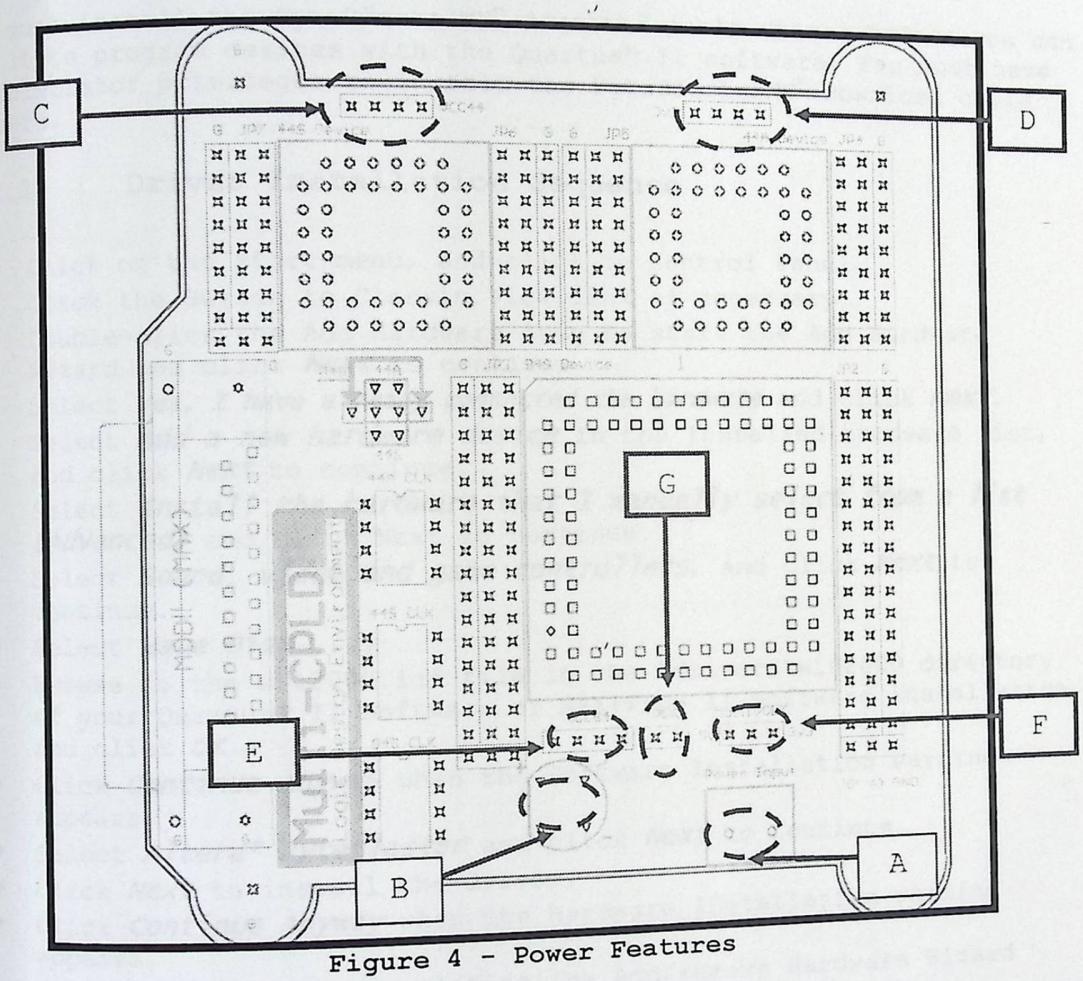


Figure 4 - Power Features

7 Programming

7.1 ByteBlasterMV™

The *Multi-CPLD™* has a native ByteBlasterMV™ on board and ready to use. A standard DB-25 (male/female) parallel cable (straight thru) is needed to interface with most computers. As noted in "System Requirements", the *Multi-CPLD™* requires a computer with an available parallel port. The *Multi-CPLD™* is incompatible with most parallel to USB converters.

7.2 ByteBlasterMV™ Driver Installation

You must install the ByteBlasterMV™ download cable driver before you can use it to program devices with the Quartus™ II software. You must have Administrator privileges to install the ByteBlasterMV™ download cable drivers.

7.2.1 Driver Installation Sequence

- Click on the Start menu, and click on Control Panel.
- Click the Switch to Classic View link if necessary.
- Double-click the Add Hardware icon to start the Add Hardware Wizard and click ***Next*** to continue.
- Select ***Yes, I have already connected the hardware*** and click ***Next***.
- Select ***Add a new hardware device*** in the Installed hardware list, and click ***Next*** to continue.
- Select ***Install the hardware that I manually select from a list (Advanced)*** and click Next to continue.
- Select ***Sound, video and game controllers***, and click ***Next*** to continue.
- Select ***Have Disk ....***
- Browse to the win2000.inf file in the \drivers\win2000 directory of your Quartus™ II software or MAX+PLUS II software installation and click ***OK***.
- Click ***Continue Anyway*** when the Software Installation warning appears.
- Select ***Altera® ByteBlaster*** and click ***Next*** to continue.
- Click ***Next*** to install the driver.
- Click ***Continue Anyway*** when the Hardware Installation warning appears.
- Click ***Finish*** in the Completing the Add/Remove Hardware Wizard window.
- Reboot the computer.
- Complete your installation by setting up programming hardware

7.3 Setting Up Programming Hardware in Quartus™ II Software

For earlier versions of either Quartus™ or Max Plus II, see: http://www.altera.com/support/software/drivers/dri-Quartus.html

- Start the Quartus™ II software.
- Choose ***Programmer*** from the Tools menu. The Programmer window will open.
- Click the ***Hardware Setup...*** button to open the Hardware Setup window.
- The selected programming hardware is identified as Currently Selected Hardware.
- Programming hardware that is already set up appears in the Available hardware items window.
- Click the ***Add Hardware*** button to open the Add Hardware window if the programming hardware you would like to use is not listed in the Available hardware items window.
- Select the appropriate programming cable or programming hardware from the Hardware Type list.
- Select the appropriate port and baud rate if necessary.
- Click ***OK***.
- Select the programming hardware you would like to use by choosing it in the Available hardware items list.
- Click ***Close***.
- Your programming hardware has been set up.

7.4 Discrete Device Programming

To program the *Multi-CPLD™*, the appropriate jumpers must be moved on the Programming header (circled in Figure 5 below).

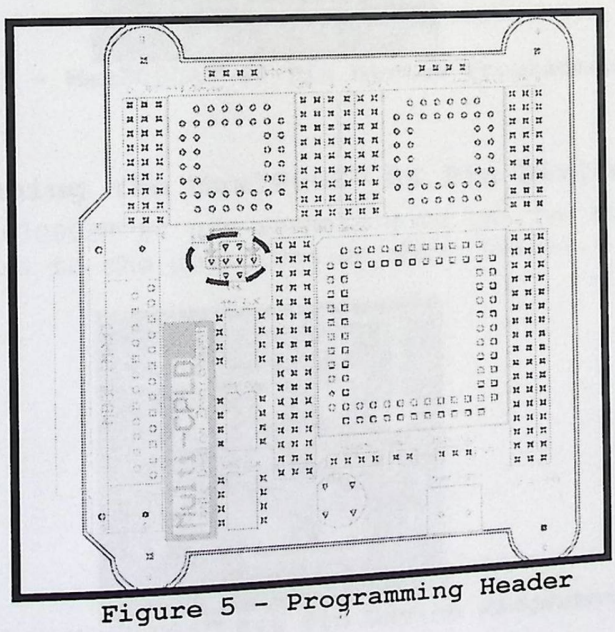


Figure 5 - Programming Header

7.4.1 Programming the Max7000S™ 44 Pin Device

To program the Max7000S™ 44 pin socket, move the two supplied programming jumpers to the positions as shown below:

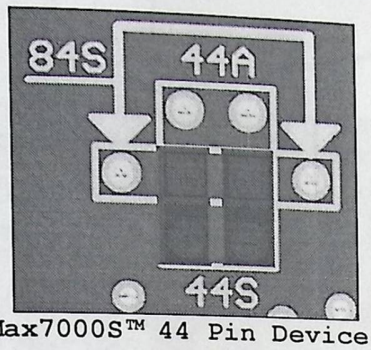


Figure 6 - Max7000S™ 44 Pin Device Programming

7.4.2 Programming the Max3000A™ 44 Pin Device

To program the Max3000A™ 44 pin socket, move the two supplied programming jumpers to the positions as shown below:

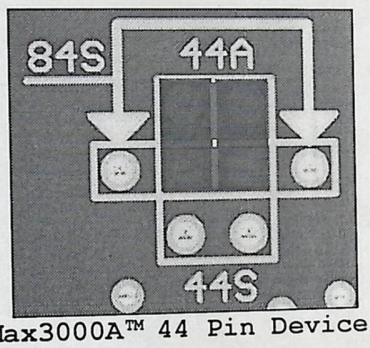


Figure 5 - Max3000A™ 44 Pin Device Programming

7.4.3 Programming the Max7000S™ 84 Pin Device

To program the Max7000S™ 84 pin socket, move the two supplied programming jumpers to the positions as shown below:

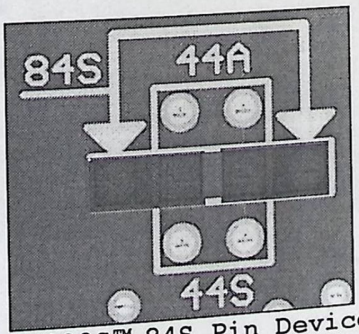


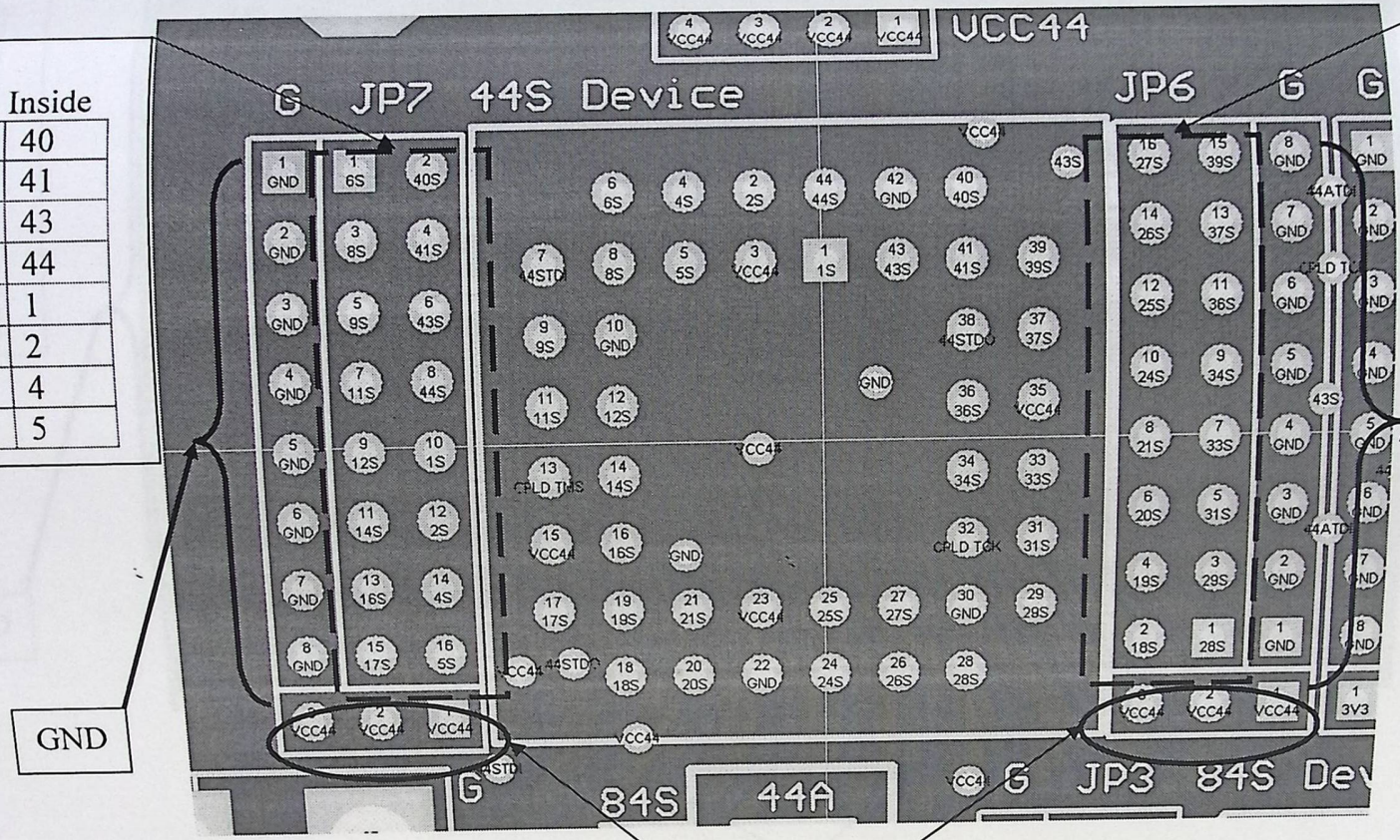
Figure 6 - Max7000S™ 84S Pin Device Programming

8 I/O Pin Locations

8.1 Max7000S™ "44S" 44 Pin Device Pin Map

JP7:

Outside	Inside
6	40
8	41
9	43
11	44
12	1
14	2
16	4
17	5



JP6:

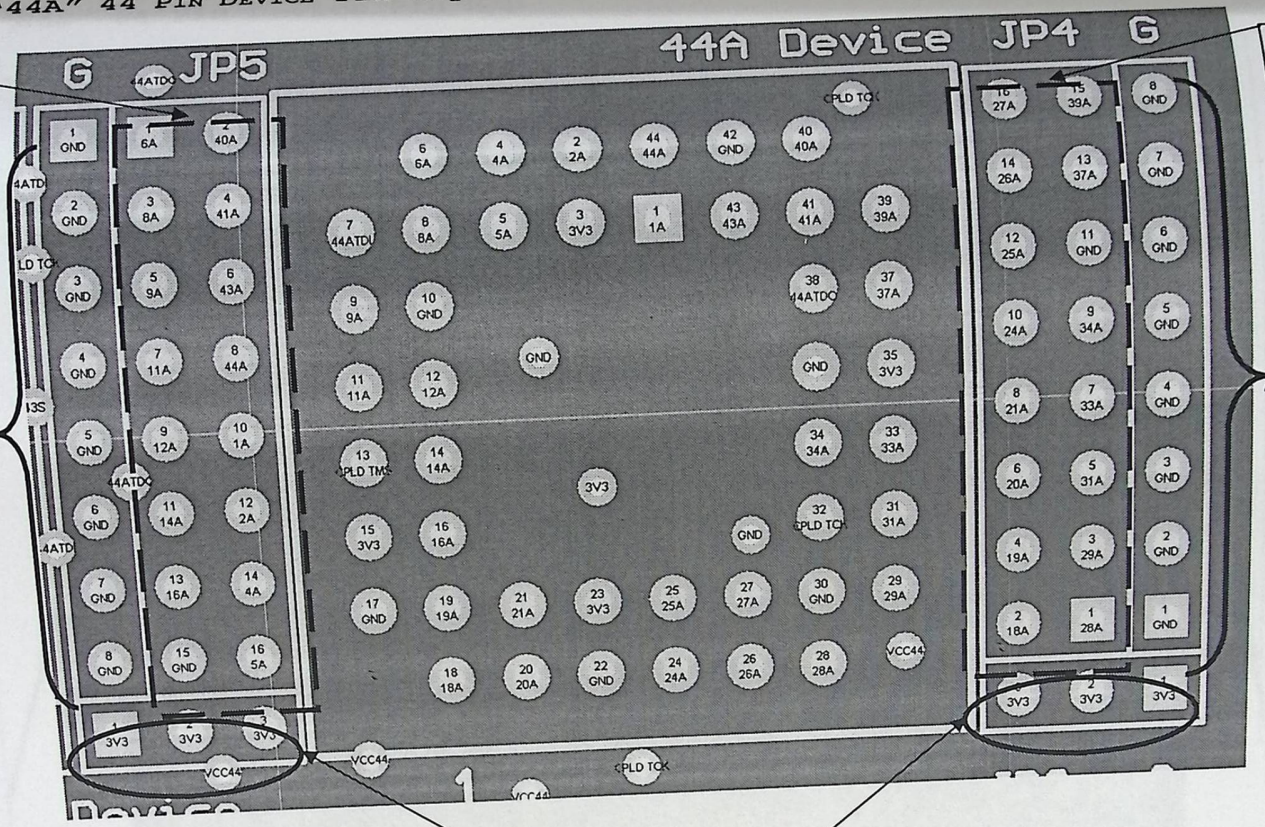
Inside	Outside
27	39
26	37
25	36
24	34
21	33
20	31
19	29
18	28

8.2

MAX3000A "44A" 44 PIN DEVICE Pin Map

JP5:

Outside	Inside
6	40
8	41
9	43
11	44
12	1
14	2
16	4
GND	5



JP4:

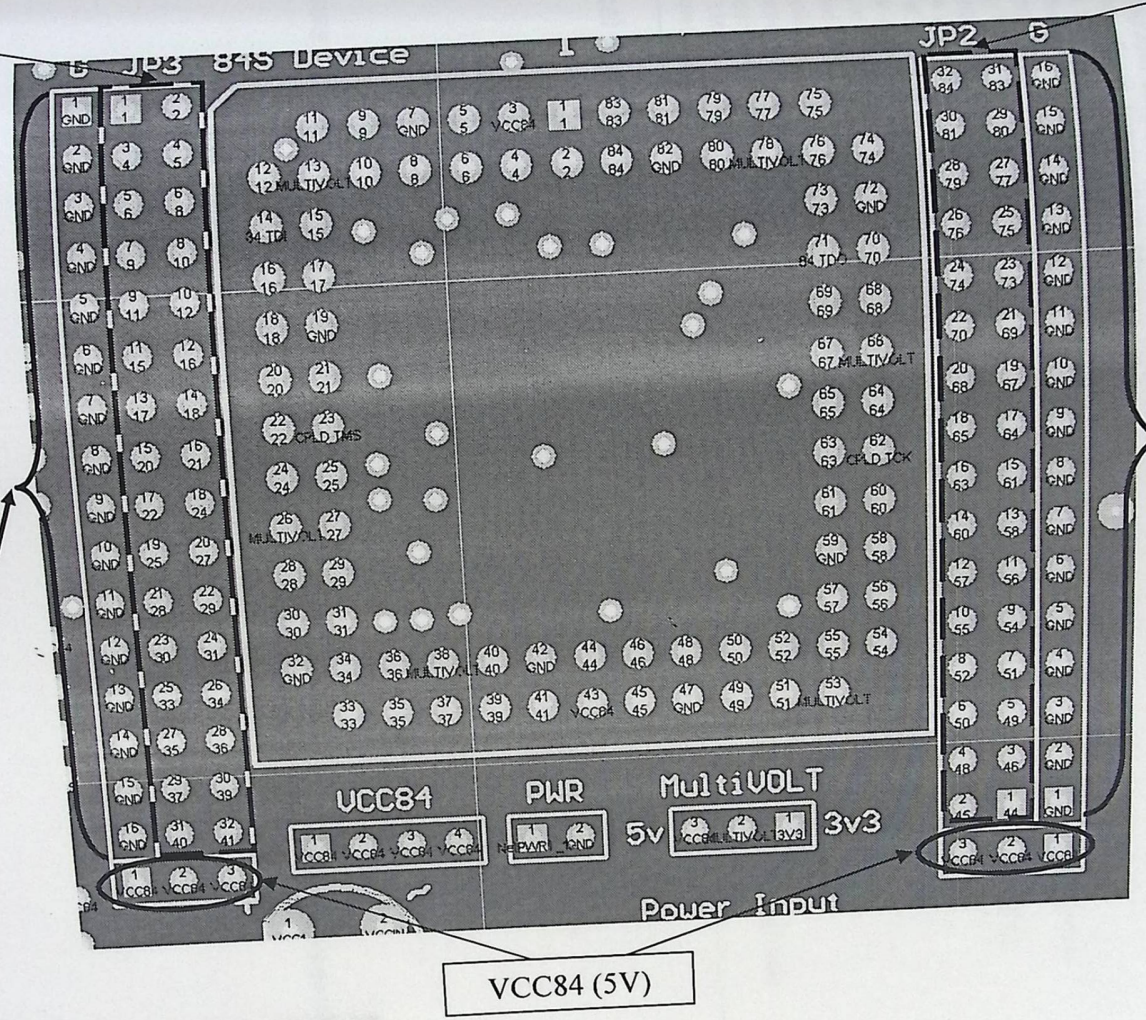
Inside	Outside
27	39
26	37
25	GND
24	34
21	33
20	31
19	29
18	28

8.3 Max7000S "84S" 84 Pin Device Pin Map

JP3:

Outside	Inside
1	2
4	5
6	8
9	10
11	12
15	16
17	18
20	21
22	24
25	27
28	29
30	31
33	34
35	36
37	39
40	41

GND



VCC84 (5V)

JP2:

Inside	Outside
84	83
81	80
79	77
76	75
74	73
70	69
68	67
65	64
63	61
60	58
57	56
55	54
52	51
50	49
48	46
45	44

GND

November 2000

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

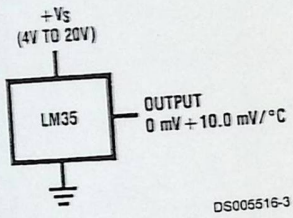
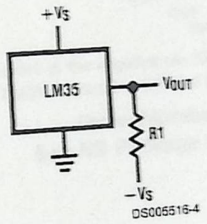


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$

V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

November 2000

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

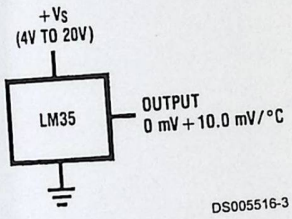
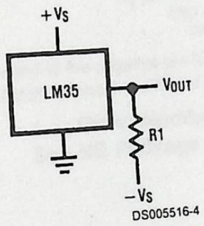


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)

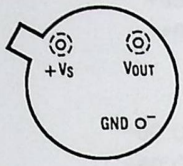


Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

Connection Diagrams

TO-46
Metal Can Package*

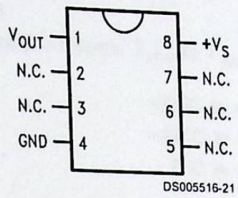


BOTTOM VIEW
DS005516-1

*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

SO-8
Small Outline Molded Package

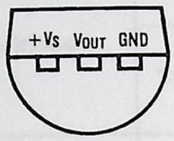


DS005516-21

N.C. = No Connection

Top View
Order Number LM35DM
See NS Package Number M08A

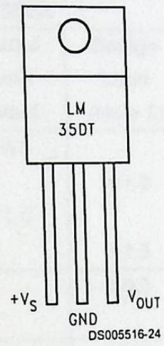
TO-92
Plastic Package



BOTTOM VIEW
DS005516-2

Order Number LM35CZ, LM35CAZ or LM35DZ
See NS Package Number Z03A

TO-220
Plastic Package*



DS005516-24

*Tab is connected to the negative pin (GND).
Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT
See NS Package Number TA03F

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60˚C to +180˚C
TO-92 Package,	−60˚C to +150˚C
SO-8 Package,	−65˚C to +150˚C
TO-220 Package,	−65˚C to +150˚C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300˚C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260˚C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215˚C
Infrared (15 seconds)	220˚C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55˚C to +150˚C
LM35C, LM35CA	−40˚C to +110˚C
LM35D	0˚C to +100˚C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	T_A=+25˚C	±0.2	±0.5		±0.2	±0.5		˚C
	T_A=−10˚C	±0.3			±0.3		±1.0	˚C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		˚C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	˚C
Nonlinearity (Note 8)	T_{MIN}≤T_A≤T_{MAX}	**±0.18**		**±0.35**	**±0.15**		**±0.3**	˚C
Sensor Gain (Average Slope)	T_{MIN}≤T_A≤T_{MAX}	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/˚C
Load Regulation (Note 3) 0≤I_L≤1 mA	T_A=+25˚C	±0.4	±1.0		±0.4	±1.0		mV/mA
	T_{MIN}≤T_A≤T_{MAX}	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation (Note 3)	T_A=+25˚C	±0.01	±0.05		±0.01	±0.05		mV/V
	4V≤V_S≤30V	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25˚C	56	67		56	67		µA
	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25˚C	56.2	68		56.2	68		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	4V≤V_S≤30V, +25˚C	0.2	1.0		0.2	1.0		µA
	4V≤V_S≤30V	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/˚C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	˚C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			˚C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		°C
	T_A=−10°C	±0.5			±0.5		±1.5	°C
	T_A=T_{MAX}	±0.8	±1.5		±0.8		±1.5	°C
	T_A=T_{MIN}	±0.8		±1.5	±0.8		±2.0	°C
Accuracy, LM35D (Note 7)	T_A=+25°C				±0.6	±1.5		°C
	T_A=T_{MAX}				±0.9		±2.0	°C
	T_A=T_{MIN}				±0.9		±2.0	°C
Nonlinearity (Note 8)	T_{MIN}≤T_A≤T_{MAX}	**±0.3**		**±0.5**	**±0.2**		**±0.5**	°C
Sensor Gain (Average Slope)	T_{MIN}≤T_A≤T_{MAX}	**+10.0**	**+9.8, +10.2**		**+10.0**		**+9.8, +10.2**	mV/°C
Load Regulation (Note 3) 0≤I_L≤1 mA	T_A=+25°C	±0.4	±2.0		±0.4	±2.0		mV/mA
	T_{MIN}≤T_A≤T_{MAX}	**±0.5**		**±5.0**	**±0.5**		**±5.0**	mV/mA
Line Regulation (Note 3)	T_A=+25°C	±0.01	±0.1		±0.01	±0.1		mV/V
	4V≤V_S≤30V	**±0.02**		**±0.2**	**±0.02**		**±0.2**	mV/V
Quiescent Current (Note 9)	V_S=+5V, +25°C	56	80		56	80		µA
	V_S=+5V	**105**		**158**	**91**		**138**	µA
	V_S=+30V, +25°C	56.2	82		56.2	82		µA
	V_S=+30V	**105.5**		**161**	**91.5**		**141**	µA
Change of Quiescent Current (Note 3)	4V≤V_S≤30V, +25°C	0.2	2.0		0.2	2.0		µA
	4V≤V_S≤30V	**0.5**		**3.0**	**0.5**		**3.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.7**	**+0.39**		**+0.7**	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			°C

Note 1: Unless otherwise noted, these specifications apply: −55°C≤T_J≤+150°C for the LM35 and LM35A; −40°C≤T_J≤+110°C for the LM35C and LM35CA; and 0°C≤T_J≤+100°C for the LM35D. V_S=+5Vdc and I_{LOAD}=50 µA, in the circuit of *Figure 2*. These specifications also apply from +2°C to T_{MAX} in the circuit of *Figure 1*. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is 400°C/W, junction to ambient, and 24°C/W junction to case. Thermal resistance of the TO-92 package is 180°C/W junction to ambient. Thermal resistance of the small outline molded package is 220°C/W junction to ambient. Thermal resistance of the TO-220 package is 90°C/W junction to ambient. For additional thermal resistance information see table in the Applications section.

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and 10mv/°C times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in °C).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of *Figure 1*.

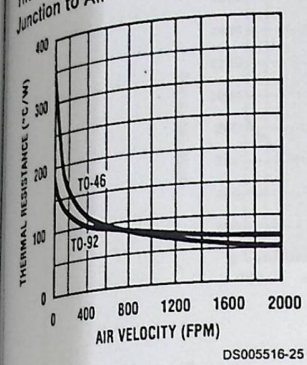
Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

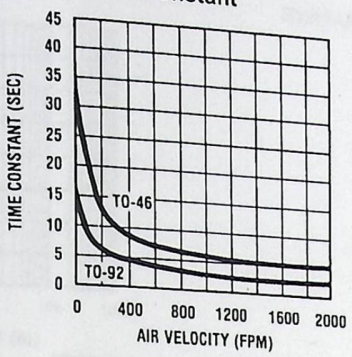
Typical Performance Characteristics

Thermal Resistance Junction to Air



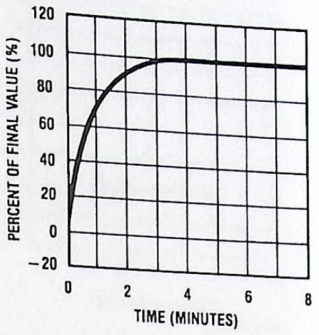
DS005516-25

Thermal Time Constant



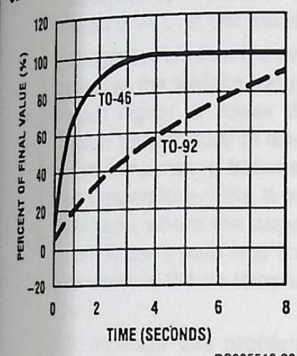
DS005516-26

Thermal Response in Still Air



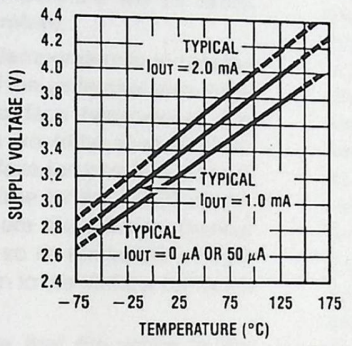
DS005516-27

Thermal Response in Stirred Oil Bath



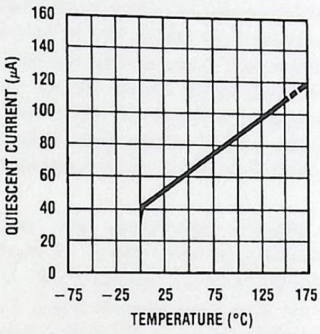
DS005516-28

Minimum Supply Voltage vs. Temperature



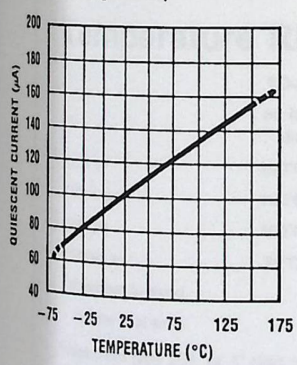
DS005516-29

Quiescent Current vs. Temperature (In Circuit of *Figure 1*.)



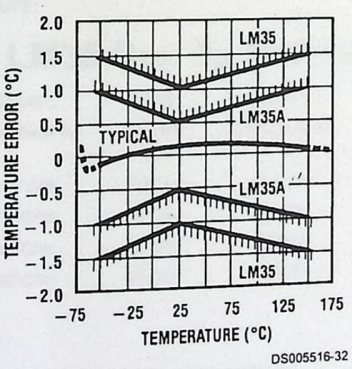
DS005516-30

Quiescent Current vs. Temperature (In Circuit of *Figure 2*.)



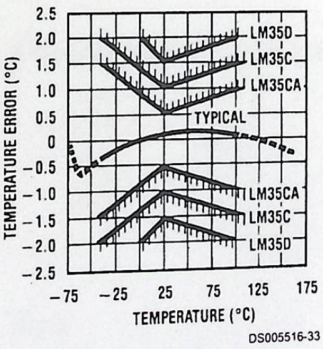
DS005516-31

Accuracy vs. Temperature (Guaranteed)



DS005516-32

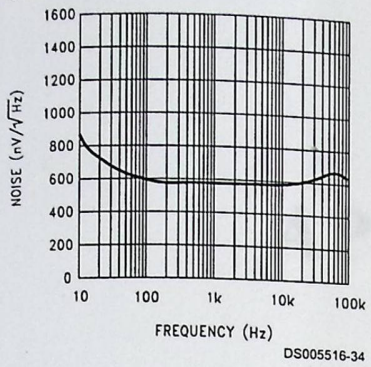
Accuracy vs. Temperature (Guaranteed)



DS005516-33

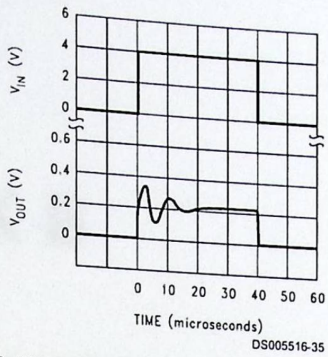
Typical Performance Characteristics (Continued)

Noise Voltage



DS005516-34

Start-Up Response



DS005516-35

Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is expecially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V− terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor, to give the steadiest reading despite small deviations in the air temperature.

Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, θ_{JA})

	TO-46, no heat sink	TO-46*, small heat fin	TO-92, no heat sink	TO-92**, small heat fin	SO-8 no heat sink	SO-8** small heat fin	TO-220 no heat sink
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	90°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	26°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Stirred oil	50°C/W	30°C/W	45°C/W	40°C/W			
(Clamped to metal, Infinite heat sink)	(24°C/W)				(55°C/W)		

*Wakefield type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.
**TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. foil or similar.

APPENDIX B

SOURCE CODE

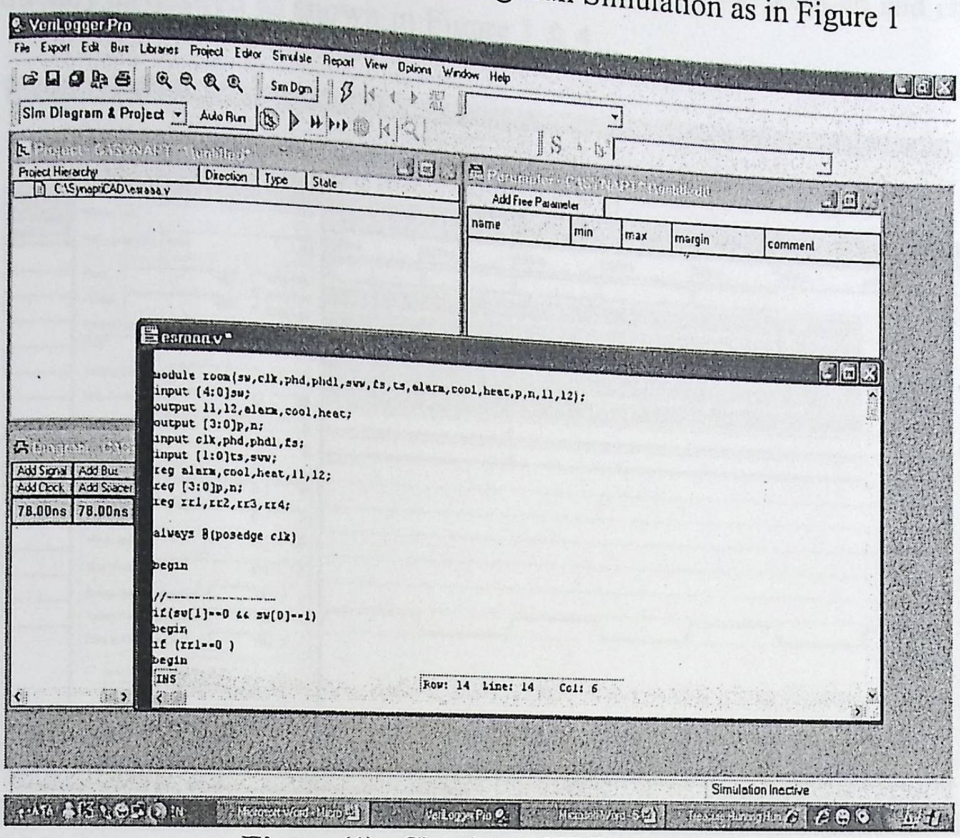
```
room(sw,clk,phd,phd1,sww,fs,ts,alarm,cool,heat,p,n,l1,l2);
[4:0]sw;
l1,l2,alarm,cool,heat;
[3:0]p,n;
clk,phd,phd1,fs;
[1:0]ts,sww;
arm,cool,heat,l1,l2;
3:0]p,n;
rr1,rr2,rr3,rr4;

@(posedge clk)


----------FOR OPEN OR CLOSE DOOR-------------//
w[1]==0 && sw[0]==1)

rr1==0 )

b0011;
1;
0;

if (rr2==1)

b0000;
=0;
=1;


if(sw[1]==0 && sw[0]==0)

'b0000;
=0;
=0;


if(sw[1]==1 && sw[0]==0)

(rr2==0)

phd==1'b0)

'bl100 ;
1=1;
1=0;

if(phd==1'b1)

'b0000 ;
2=0;
1=0;

end
if(rr1==1)

'b0000;
1=0;
2=1;


if (sw[1]==1 && sw[0]==1)

if (rr1==1 && rr2==0)

'bl100 ;
2=1;


if(rr2==1 && rr1==0)

'b0011;
1=1;
```

```
d
end

//----------------FOR OPEN OR CLOSE CURTAIN---------------//

f(sw[3]==0 && sw[2]==1)
egin
f (rr3==0 )
egin
=4'b0011;
r3=1;
r4=0;
nd
lse if (rr4==1)
egin
=4'b0000;
r4=0;
r3=1;
nd
nd

lse if(sw[3]==0 && sw[2]==0)
egin
=4'b0000;
rr3=0;
rr4=0;
nd

lse if(sw[3]==1 && sw[2]==0)
egin
f (rr4==0)
egin
=4'b1100 ;
rr4=1;
rr3=0;
nd
lse if(rr3==1)
egin
=4'b0000;
rr3=0;
rr4=1;
nd
nd

lse if (sw[3]==1 && sw[2]==1)
egin
f (rr3==1 && rr4==0)
egin
=4'b1100 ;
rr4=1;
nd
lse if(rr4==1 && rr3==0)
egin
=4'b0011;
rr3=1;
nd
  end

//----------------FOR LGITH CHCKING-------------//

f(sw[4]==0)
egin
f(phd1==1'b0)
egin
l1=1;
l2=0;
nd
lse if (phd1==1'b1)
egin
 l1=1;
 l2=1;
nd
  end
```

```
se if(sw[4]==1)
gin
se(sww)
b00: begin
=0; l2=0;
d
b01: begin
=0; l2=1;
d
b10: begin
=1; l2=0;
d
b11: begin
=1; l2=1;
d
ndcase
 end

//-------------FOR FIRE CHCHING -------------//
if(fs == 0)
larm=0;
lse if(fs == 1)
egin
larm=1;
=4'b0011;
=4'b0011;
nd


//------------FOR TEMPRETURE CHCKING---------//
if(ts == 2'b00)
egin
cool=0;
heat=0;
end
else
 if(ts == 2'b01)
egin
cool=1;
heat=0;
end
else
 if(ts == 2'b10)
egin
cool=0;
heat=1;
end


  end



endmodule
```

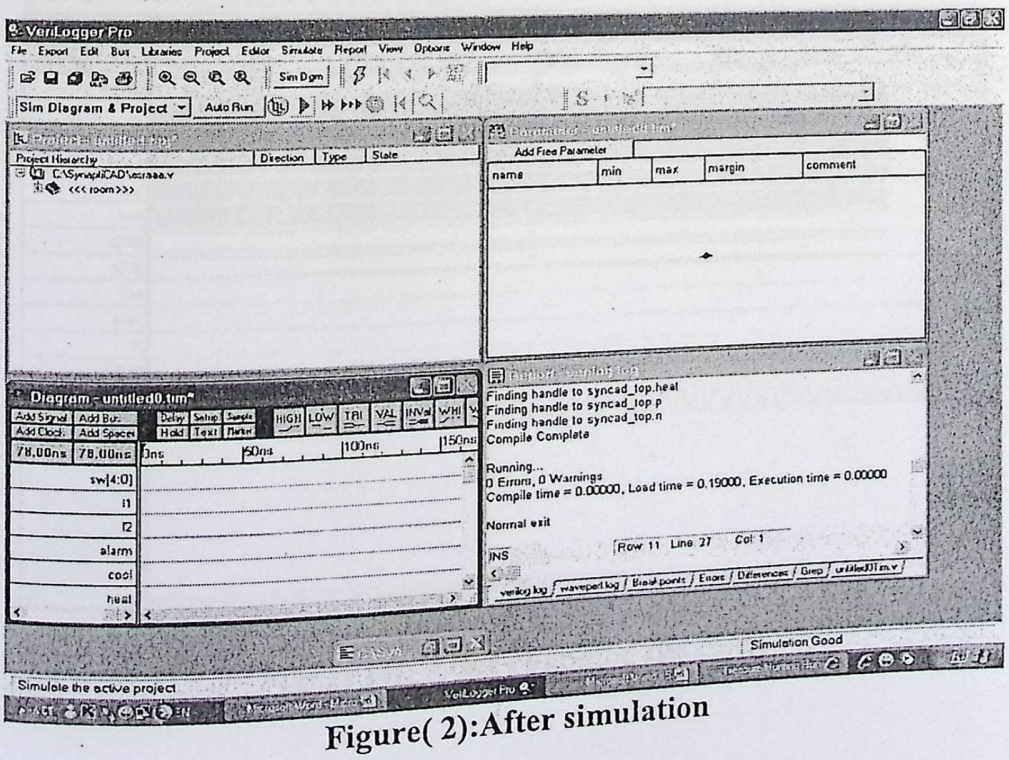
1- Verilogger Pro

We start simulating the program by clicking Run Simulation as in Figure 1



Figure(1): Checking HDL program

Simulation shows if there are errors or simulation is good in Report sub window and Diagram sub window shows the timing diagrams as in Figure 2



Figure(2):After simulation

At the beginning of the program we must determine the clock by clicking on add clock then writing the symbol of the clock as it is used in the program and changing the frequency as desired as shown in Figure 3 & 4

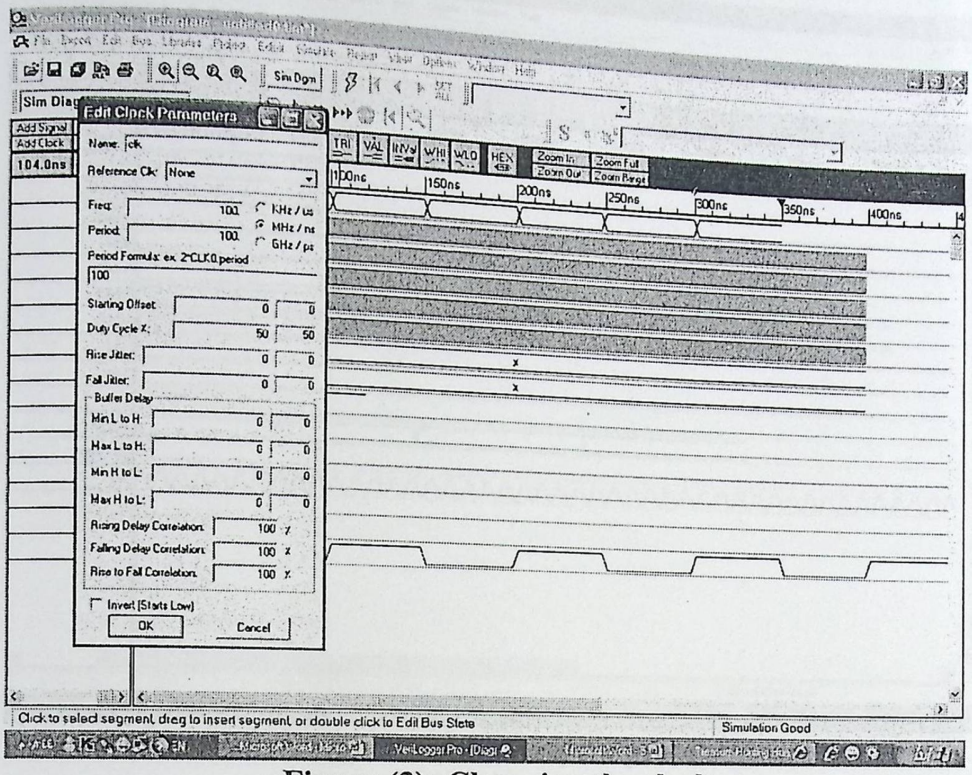


Figure (3): Changing the clock

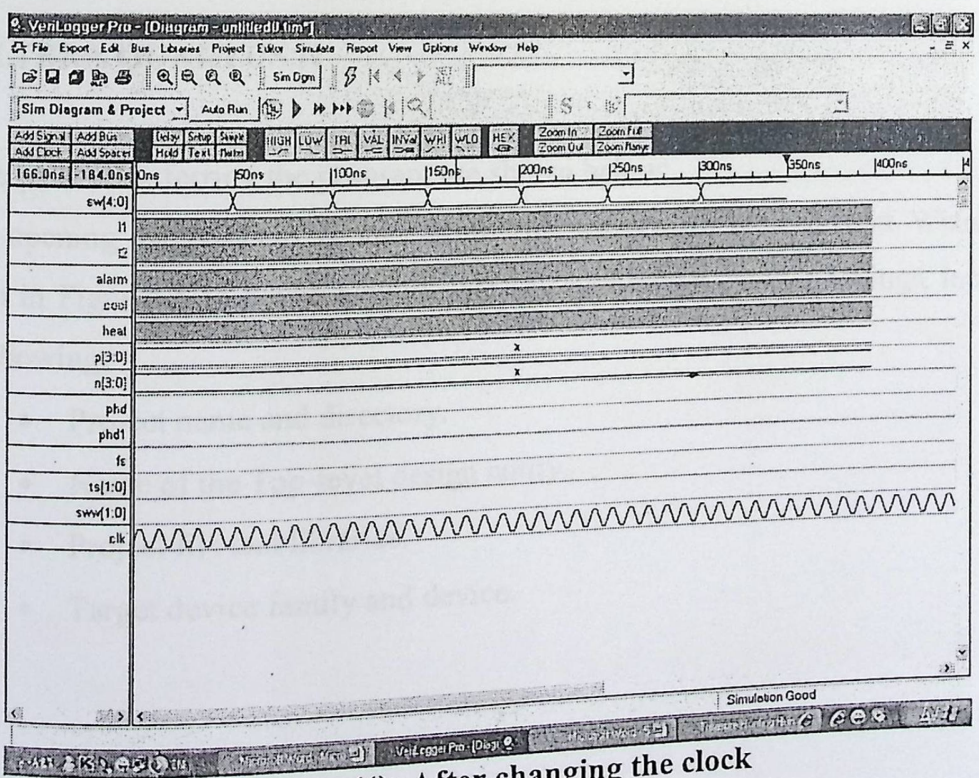
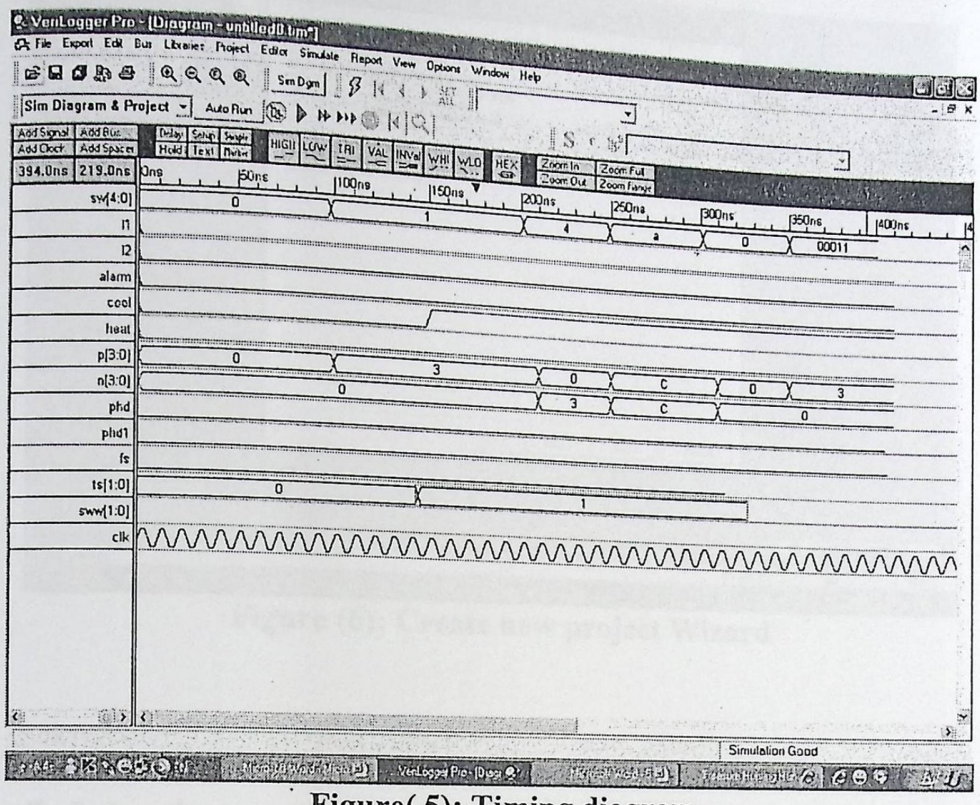


Figure (4): After changing the clock

Figures 5 show how to deal with timing diagrams, and by changing the input the out put will change, the inputs in black and the outputs in red.



Figure(5): Timing diagram

2 - ALTERA QUARTUS II

The steps of transferring the program are shown below:

After opening QUARTUS II and from File we choose New Project Wizard…as shown in Figures 6 to create new project and preliminary project settings, including the following:

- Project name and directory.
- Name of the Top-level design entity.
- Project file and libraries.
- Target device family and device.

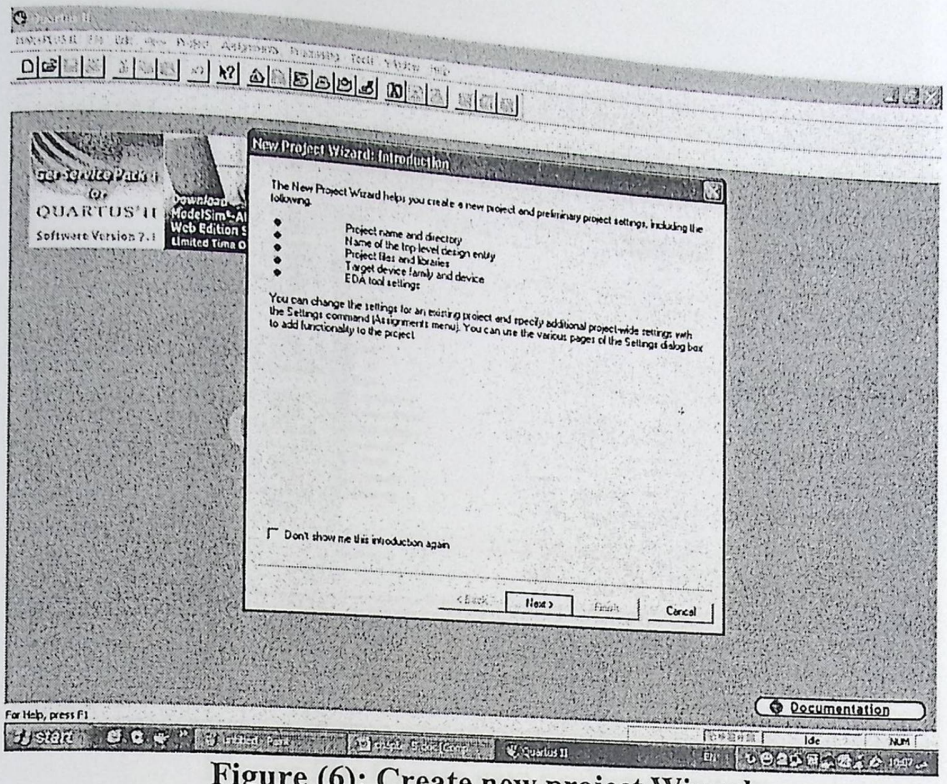


Figure (6): Create new project Wizard

Figures 7 , 8 show how to create new project name and new project directory.

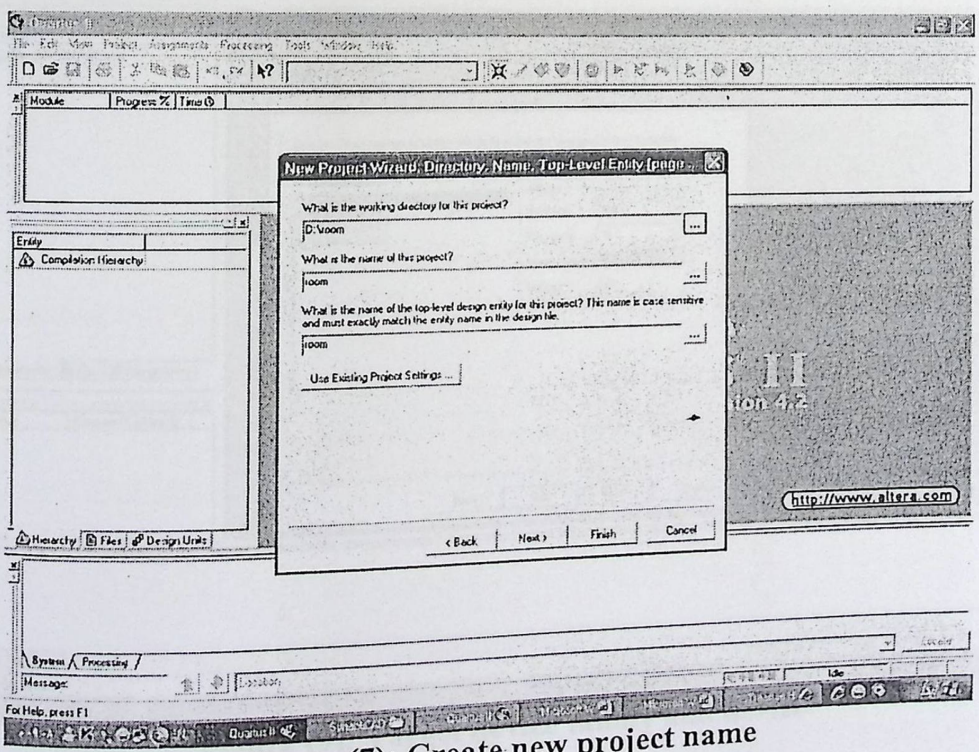


Figure (7): Create new project name

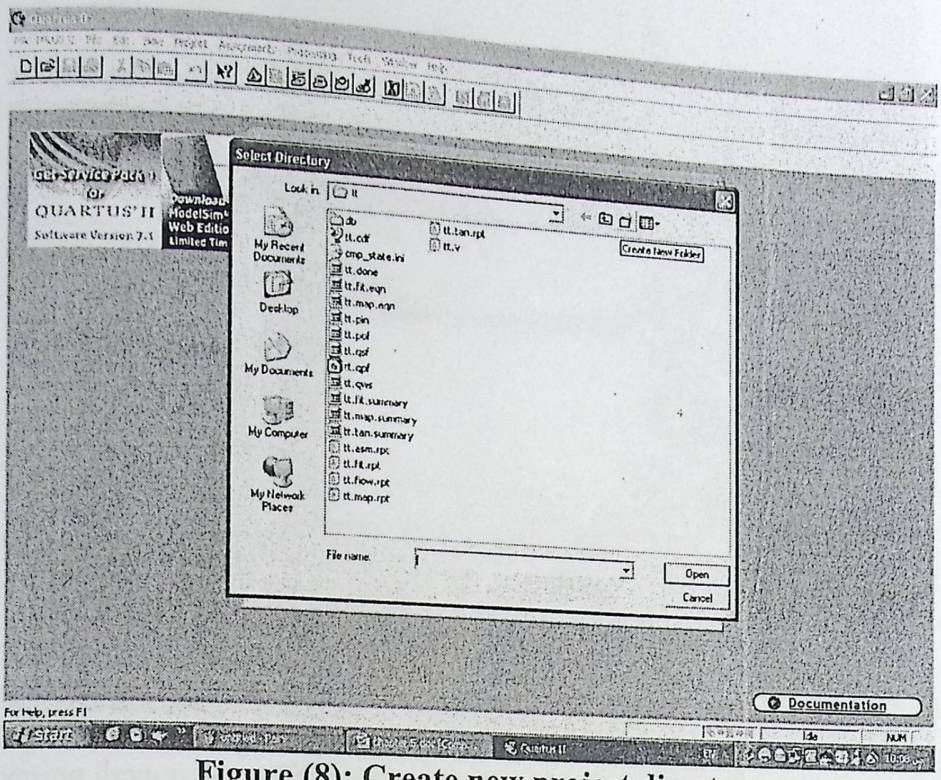


Figure (8): Create new project directory

Figure 9 show how to select the chip family and the chip itself.

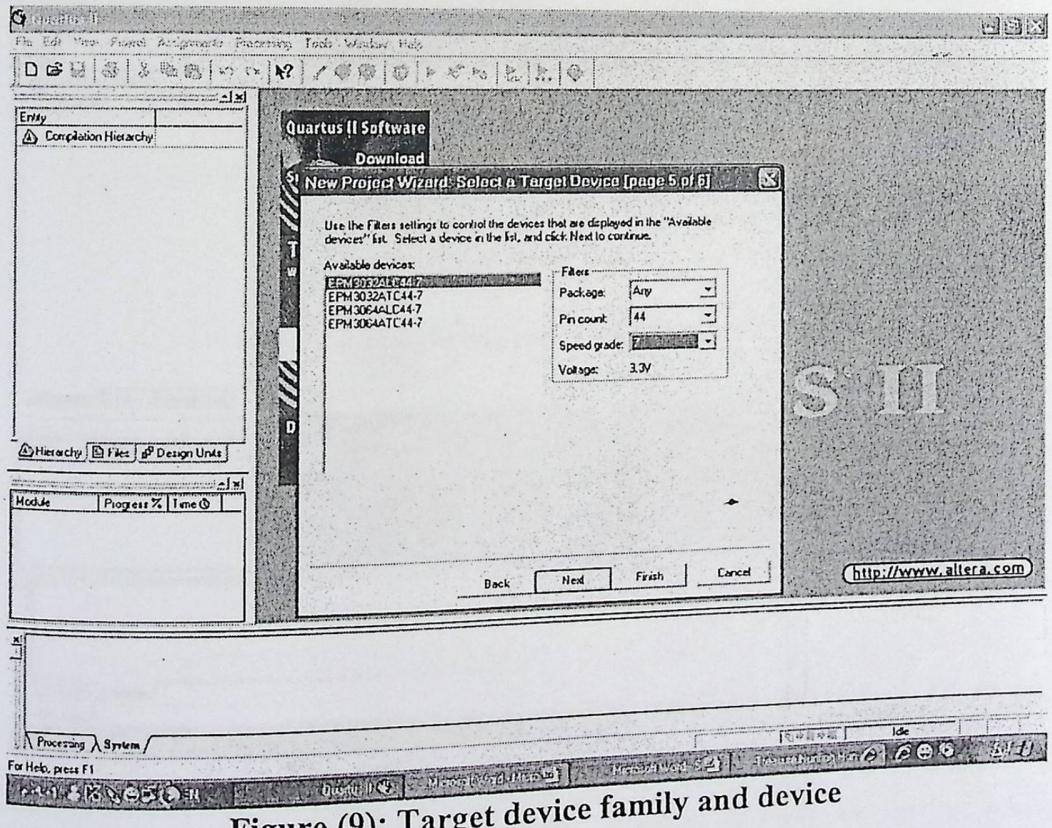


Figure (9): Target device family and device

Figure 10 shows how to select device design file which is in our project Verilog HDL file.

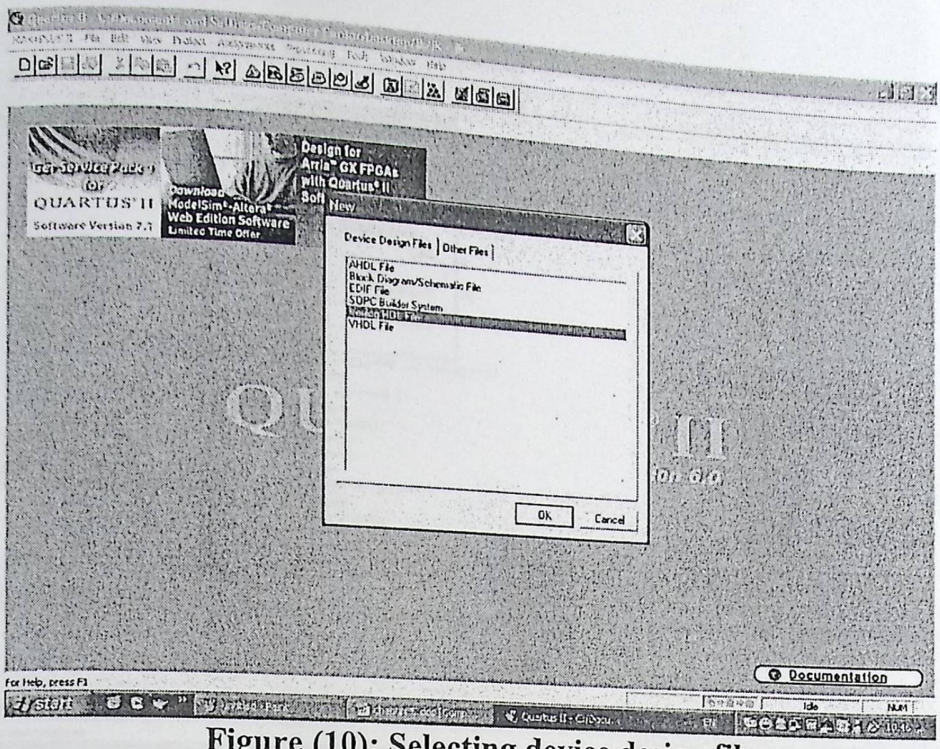


Figure (10): Selecting device design file

After opening device design file we select the program where it is saved and copy it to the new opened page as shown in Figure 11.

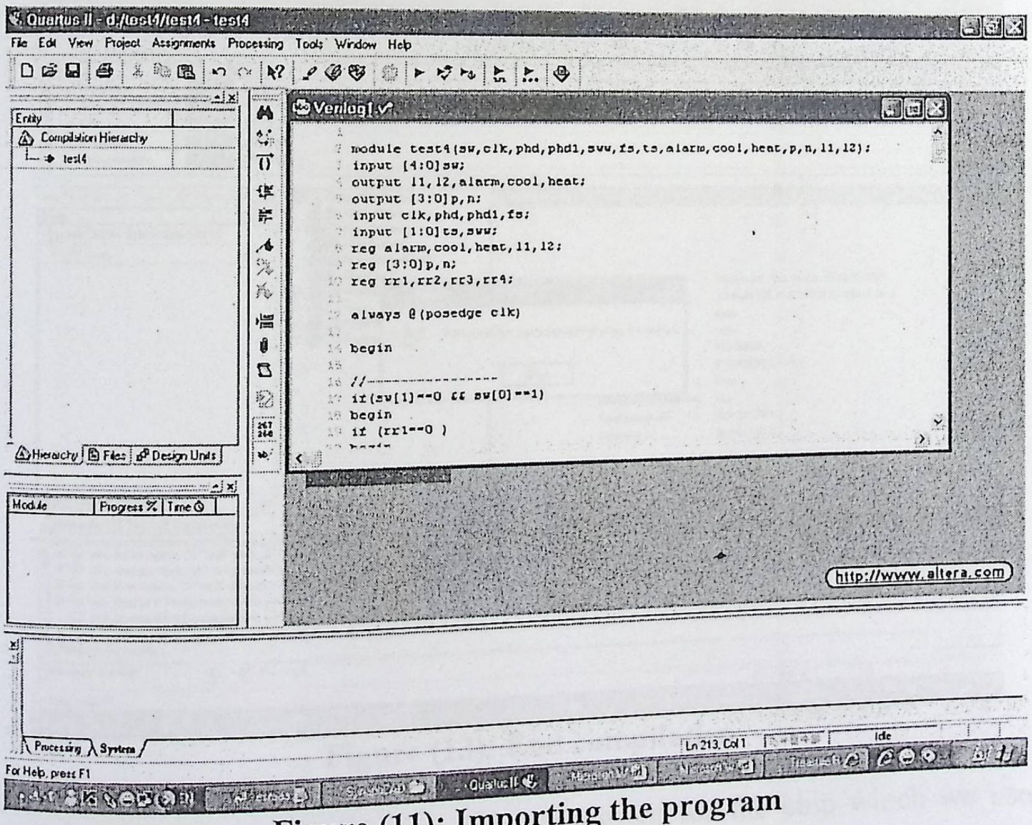


Figure (11): Importing the program

After saving the program we make compilation for it to check for errors and convert it to machine language as in Figures 12, 13.

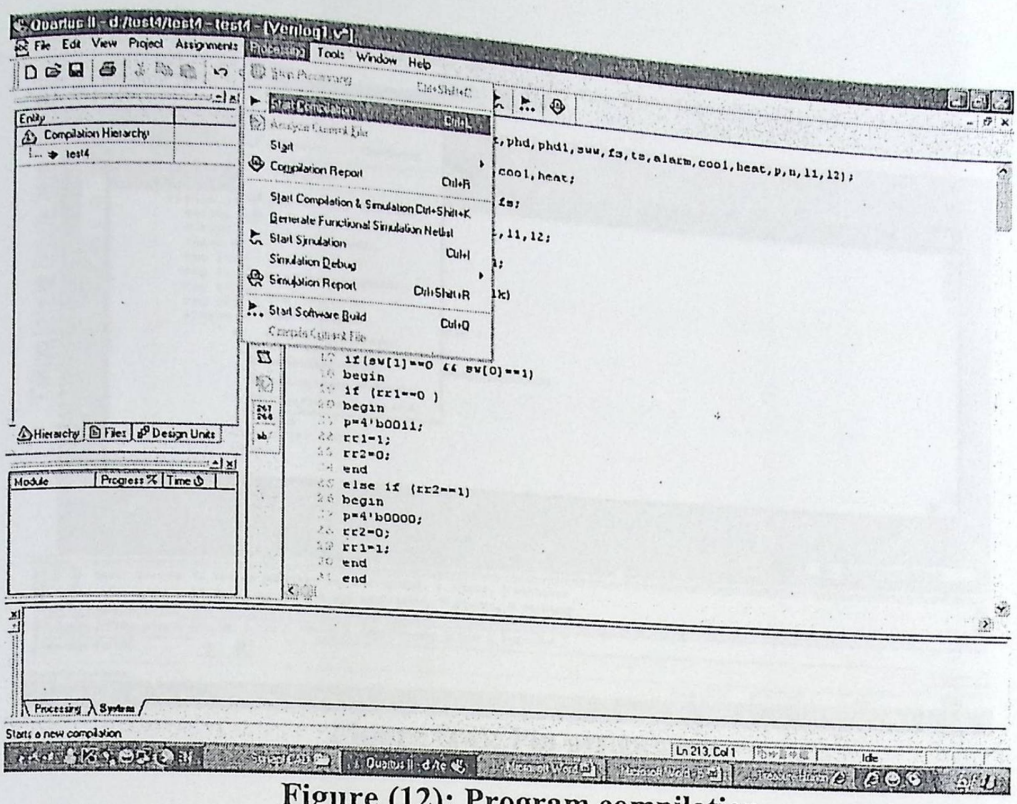


Figure (12): Program compilation

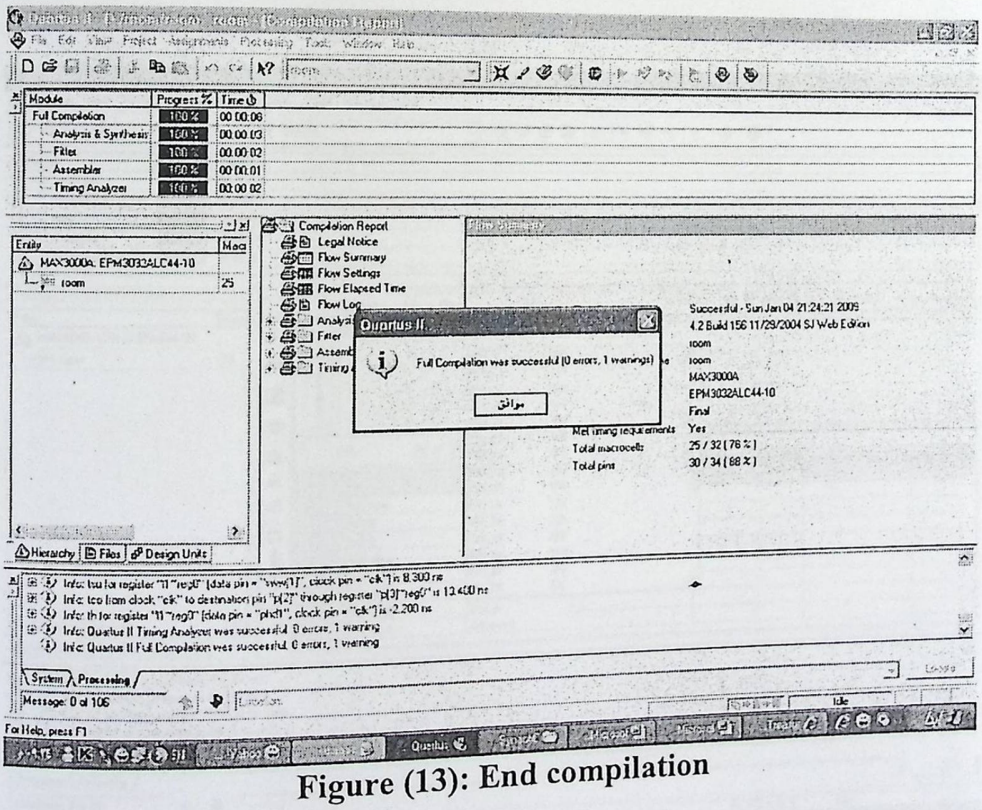


Figure (13): End compilation

Figures 14, 15 show how to make pin assignment for the chip which we used to assign the inputs and outputs and control signals.

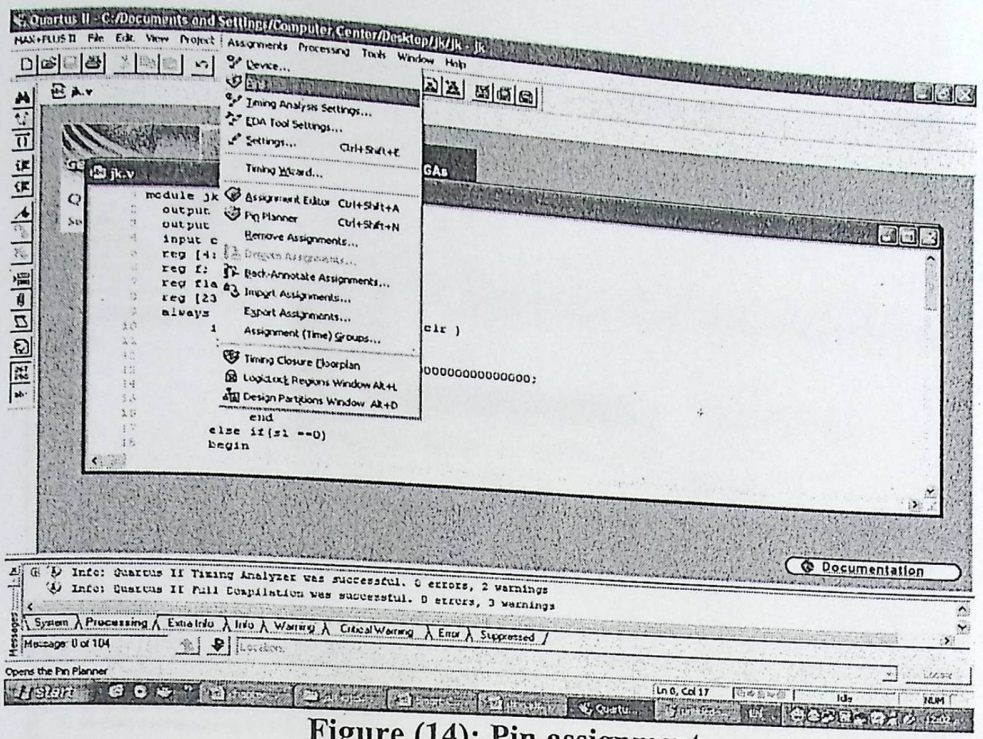


Figure (14): Pin assignment

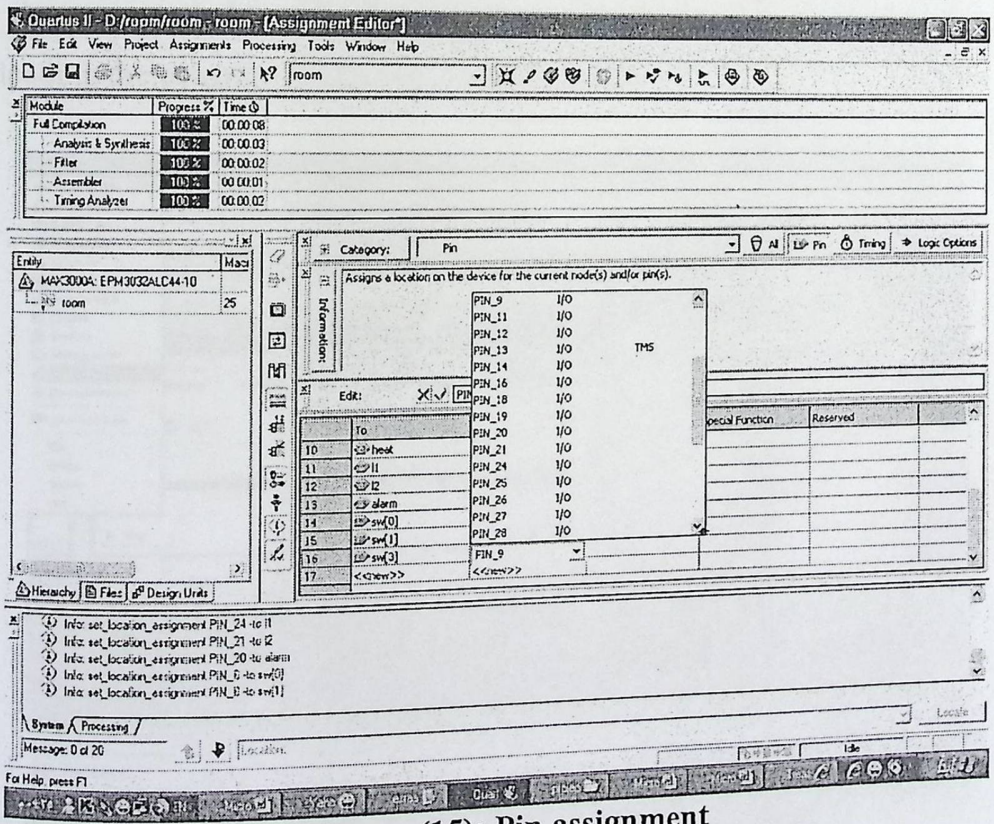


Figure (15): Pin assignment

Then we make compilation again to make sure that the pin assignment is correct and to change it to machine language as in Figure 16.

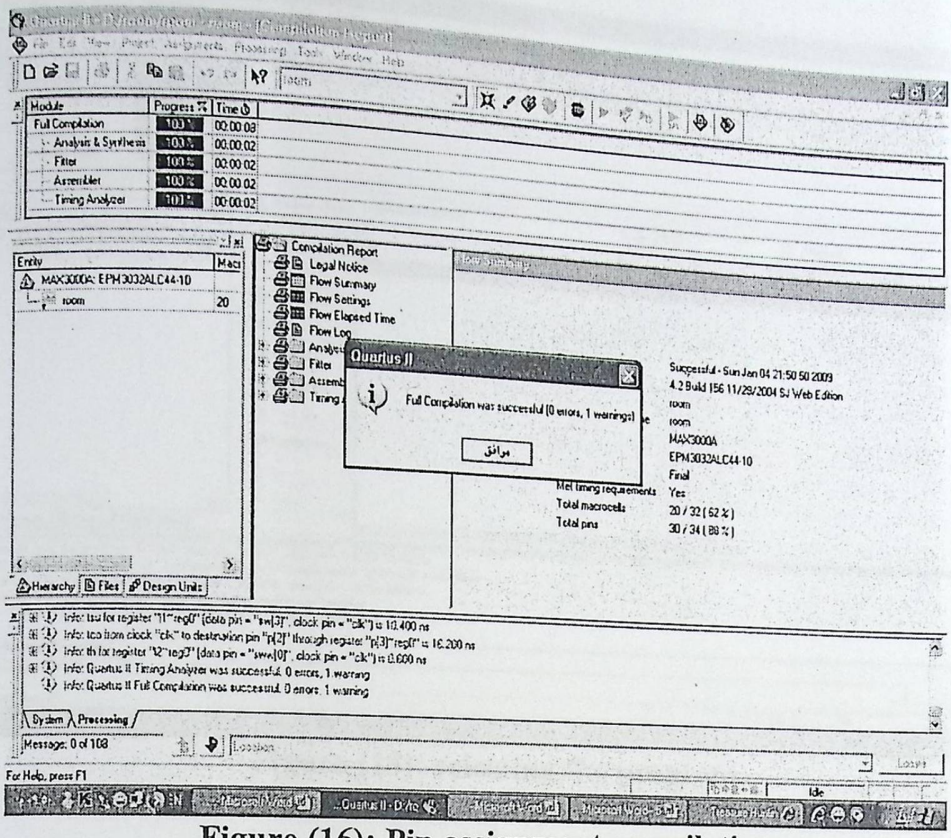


Figure (16): Pin assignment compilation

From the main task bar we click MAX+PLUS II and select programmer to start downloading the program to the chip as in Figure 17.

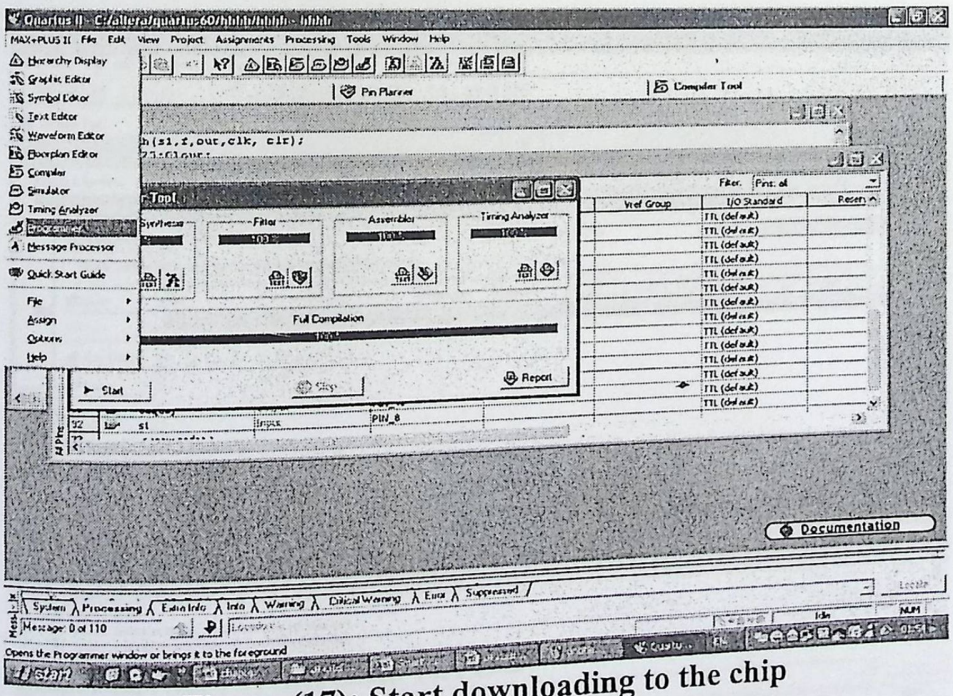


Figure (17): Start downloading to the chip

Figures 18 show how to select the program which will be downloaded to the chip and it is by double clicking file to open the saved program.

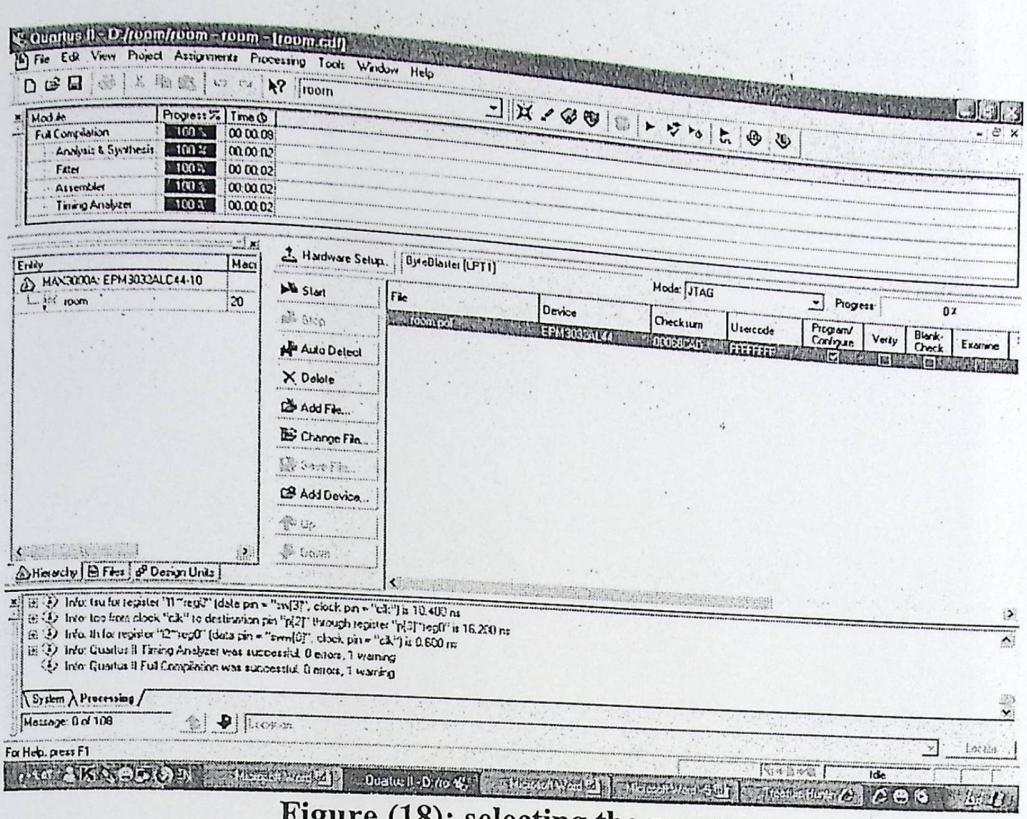


Figure (18): selecting the program

After opening the program (which is XXX.pof extension) we click on program/configure and then we click start to start downloading, the progress shows the percent of downloading as shown in 19.

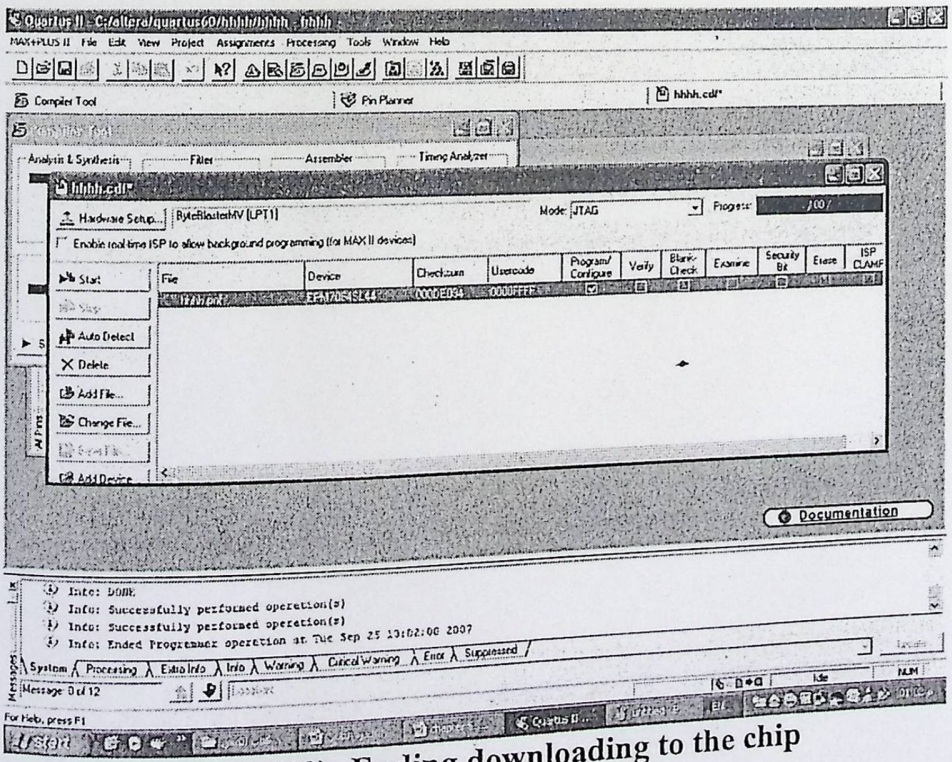


Figure (19): Ending downloading to the chip