SMART AUTOMATION SYSTEM BASED ON FIELD PROGRAMMABLE GATE ARRAY

A Thesis Submitted to the Dept. of Electrical & Electronic Engineering, BRAC UNIVERSITY in partial fulfillment of the requirements for the Bachelor of Science Degree in Electrical & Electronic Engineering

FATIMA NAFSIA TASLIM	12121050
TASFIA LAMISA AHONA	12121053
NISHAT TASNIM	12121056
NAZIFA REZWANA HAQUE	12121106

DECEMBER 2016

DECLARATION

We do hereby declare that the thesis work titled “Smart Automation System Based on Field Programmable Gate Array” is submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Electrical and Electronics Engineering. The thesis work was carried out under the supervision of Dr. Md. Belal Hossain Bhuian. We declare that this project has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged/referred.

Date:

Signature of Supervisor

Dr. Md. Belal Hossain Bhuian

_____

Thesis Supervisor

Signature of Authors

_____

NazifaRezwanaHaque

ID-12121106

_____

Fatima NafsiaTaslim

ID-12121050

_____

TasfiaLamisaAhona

ID-12121053

_____

NishatTasnim

ID-12121056

ACKNOWLEDGEMENT

We are extremely thankful to our supervisor Dr. Md. Belal Hossain Bhuian, Associate Professor, Department of Electrical and Electronic Engineering (EEE), BRAC University for his supervision and guidance. Without his continuous support this thesis would not have been possible. We would also like to thank every other individual without whose contribution the project would not be a success.

ABSTRACT

Smart automation system is an evolution of human kind to a new era. These systems do all the jobs quickly and efficiently which saves time in our daily life, and in order to make these smart systems work more efficiently, faster and accurately we have used an FPGA Board (Field- Programmable Gate Array) in our thesis project. FPGA is an integrated circuit (IC) that allows programmer to create their own design in the field after manufacture and the core is able to operate different modules simultaneously. The main goal of our thesis is to implement four features that will save us from various accidents and wastage like home on fire or over flow of water and a little bit of smart security where we used PIR motion sensor, fire sensor and water sensor. In order to complete our desired project we had to work in three portions – compilation of module, synthesis and hardware implementation. FPGA board works digitally that lead us to converting all the sensor data or outputs to digital. We designed different modules for compiling different sensors on the FPGA board, then synthesis and hardware implementation were carried out accordingly. In spite of facing many challenges up to the hardware implementation we were successfully able to reach our desired goal.

Table of Contents

List of figures

List of table

CHAPTER-1

INTRODUCTION

1.1: History of Automation System

Home automation began with labor-saving machines such as washing machines and vacuum cleaners. These instruments do all the jobs quickly and efficiently which saves time in our daily life. Self-sufficient electric or gas powered home utilizations became sustainable in the 1900s with the introduction of electric power distribution and led to the launching of washing machines (1904), water heaters (1889), refrigerators, sewing machines, dishwashers, and dryers. In 1975, the first all-purpose home automation network technology, was developed, it is a communication protocol for electronic devices. Soon after that came the wall switch module. Smart systems, began to increase in popularity in the early 2000s, different technologies began to develop. Smart systems very quickly and swiftly started to become a more reasonable option, and therefore a more feasible technology for consumers. Domestic technologies, Office security home networking, and other gadgets began to emerge more in the market and their availability increased. By 2012, in the United States, according to ABI Research, 1.5 million home automation systems were installed According to Li et. al. (2016) there are three generations of home automation:

1. Wireless technology with proxy server, e.g. Zigbee automation;
2. Artificial intelligence controls electrical devices, e.g. amazon echo;
3. Robot buddy "who" interacts with humans, e.g. Robot Rovio, Roomba.

1.2: Today’s automation systems and its constraints

Nowadays smart homes and offices are mainly about security and protection. Our smart systems are viable, and they help to make sure that our homes and offices aren’t consuming unnecessary energy. These systems also helps to alert us against dangerous hazards like fire and even signals us against intruders even though we are home or not. Even when we are far away we will still be connected to our homes, and when the house is occupied, the high level of automation enables more convenience, control and safety from any part of your property. All these advance features helped us to reduce our worries and increased our

level of enjoyment. Current developments in home automation include remote mobile control, automated lights, automated thermostat adjustment, scheduling appliances, mobile/email/text notifications etc. Some of the constraints of the home automation system with Arduino and other microcontroller are the speed is less, does not work in parallel and its difficult to introduce it in the market.

1.3: Our system with FPGA

In our smart system we have implemented an FPGA Board, by using this board our system will work faster, efficiently and accurately than the other home automation systems. Our system consumes less power compared to the other home automation systems by operating all the features used simultaneously.

1.4: Motivation

Our motive was to create a smart sensing system that can work more efficiently and effectively as we are using more sophisticated board. There were several things which encouraged us to do the project with FPGA rather than other microcontrollers are:

- FPGA are not stuck to any hardware configuration, but the microcontrollers are stuck to the configuration that the supplier provides, they cannot be rewired.
- FPGA's code is executed in parallel and microcontroller is executed sequentially.
- FPGA is faster in more complex solutions and work more efficiently and accurately.
- FPGA has incredible flexibility on its price and lower power consumption.
- It has a vast range of capability and can be interfaced through different ports.
- It is easier to build desired system and design it on chip.

There are some reasons to choose this project;

- To make an easier hardware implementation for experiment
- Easy as a basic stage of abstraction and synthesis
- There are parallel operations with the sensors

1.5: AIM of the Thesis

The purpose of this thesis work is to introduce a smart system using the FPGA Board. In this research we have designed a smart sensing system using FPGA, which will keep the people safe from various accidents which can take place in their home or even offices where they spend most of their time. Another aim of our thesis work is to reduce power consumption. Through our work we will try to reduce power wastage, which is a major problem nowadays in our society. Every person loves their home, thus to protect their home and keep it safe everyone should take a few initiatives like protect the house from fire or water overflow etc. In our project we have included such features through which a person can easily detect whether their house caught an unwanted fire or if there is water flow from their water tank or washroom especially bathtub or bucket. Most people often tend to forget to turn off the lights or fan of the room, so through our smart system we have even included a feature of automatic turning the lights and fan off of the room by using a motion sensor.

CHAPTER-2

Modeling concept of Verilog HDL in FPGA

2.1: Introduction of HDL

A Hardware Description Language (HDL) is a particular code used to portray the structure and conduct of electronic circuits, and most generally, the digital logic circuits. In electronics a HDL is a language from computer language used for formal description of electronic circuits. HDL is used to describe any digital circuits in the form of architectural, gate level and behavior of discrete electronic system. It is a very easy language which can describe circuit operation, its design and tests to verify its operation at any level of circuits and it is found to be an excellent programming language for FPGAs and CPLDs.
The Top-down design and hierarchical design method allows the design time; design cost and design errors to be reduced. Another major advantage is related to complex designs, which can be managed and verified easily. HDL provides the timing information and allows the design to be described in gate level and register transfer level. Reusability of resources is one of the other advantages.
The three common HDLs are Verilog, VHDL, and SystemC. Of these, SystemC is the newest. The HDLs will allow fast design and better verification. Nowadays, Verilog and VHDL are very common in industries. Verilog, one of the main Hardware Description Language standardized as IEEE 1364 is used for designing all types of circuits. It consists of modules and the language allows Behavioral, Dataflow and Structural Description of any complex circuit.

2.1.1: A brief History of HDL

Hardware description languages were first appeared in the late 1960s, looking like more traditional languages. The first that had a lasting effect was described in 1971 in C. Gordon Bell and Allen Newell's text Computer Structures and that text presented the idea

of register transfer level. At first it is used in the ISP language to define the behavior of the Digital Equipment Corporation (DEC) PDP-8.

Later, the language became more common with the introduction of DEC's PDP-16 RT-Level Modules (RTMs) and a book telling their use. At least two implementations of the basic ISP language (ISPL and ISPS) followed. ISPS was well suitable to define the relations between the inputs and the outputs of the design and was quickly adopted by commercial teams at DEC, as well as by a number of research teams.

2.1.2: Different Types of HDL

The most common language used in widespread are Verilog HDL and VHDL.

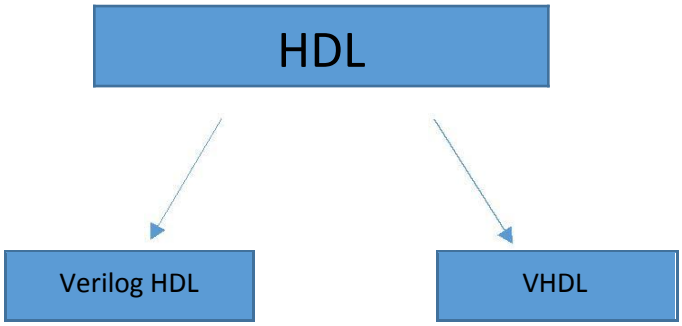

Figure-2.1: Types of HDL

There are different HDLs available for describing analog circuits, digital circuits and PCBs. HDLs for digital circuit design other than Verilog and VHDL are as follows.

- VHDL (VHSIC HDL)
- Verilog
- TL-Verilog (Transaction-Level Verilog)
- THDL++ (Template HDL inspired by C++)
- SystemVerilog
- SystemTCL
- SystemC

2.2: Abridged History of Verilog HDL

The history of Verilog HDL goes back to the 1980s, a company called Gateway Design Automation developed a logic simulator, Verilog-XL, and with it a hardware description language. It was created by PrabhuGoel and Phil Moorby between 1983 and 1984. Cadence Design Systems acquired Gateway in 1989, and with it the rights to the language and the simulator. Cadence now has full proprietary rights to Gateway's Verilog and the Verilog-XL, the HDL-simulator that would become the de facto standard for the next decade [3]. Originally, Verilog was only intended to describe and allow simulation, the automated synthesis of subdivisions of the language to physically realizable structures and gates and was developed after the language had achieved widespread usage. In 1990, Cadence put only the language into the public domain, with the intention that it should become a standard, non-proprietary language. Later, the Verilog HDL maintained as a nonprofit making group. In December 1995 Verilog HDL became IEEE Std. 1364-1995.

Verilog is a combination of the words "verification" and "logic". So after that there was a further revision in 2005. Moreover, Accellera have also developed a new standard, System Verilog, which extends Verilog. Then the System Verilog became an IEEE standard (1800-2005) in 2005.

2.2.1 Modern concepts of Verilog HDL

A design methodology is very important to understand before working with the Verilog language because the designer must use the best design methodology to do efficient Verilog HDL based on the designs. There are basically two types of digital design methodologies: a top- down design methodology and a bottom- up design methodology.

In a top-bottom methodology:

The desired design-style of all designers is the top-down one. A real top-down design allows primary testing, easy modification of different technologies, structured system design and offers many other benefits. But it is very tough to follow a pure top-down design. Due to this fact most designs are a combination of both methods, employing some key elements of both design styles.

In a bottom –up methodology:

The old-fashioned method of electronic design is bottom-up. Each design is performed at the gate-level using the standard gates. With the increasing complexity of new designs this approach is nearly impossible to maintain. New systems consist of ASIC or microprocessors with a complexity of thousands of transistors. These traditional bottom-up designs have to give way to new structural, hierarchical design methods. Without these new perform it would be impossible to handle the new complexity. Verilog supports designing at many different levels of abstraction. Three of them are very important: Verilog is both a behavioral and structural language. There are four level of abstraction to describe a module which can be changed without any change in the settings.

They are as follows:

Behavioral or algorithmic level

Dataflow level

Gate level

Switch level

Behavioral or algorithmic level:

This level defines a system by simultaneous algorithms (Behavioral). Each algorithm itself is sequential, that means it consists of a set of instructions that are executed one after the other. Functions, Tasks and Always blocks are the main elements. There is no regard to the structural comprehension of the design.

Dataflow level:

For small circuits, the gate level modeling method works very well because the number of gates is inadequate and designer can instantiate and links every gate individually. However in complex design the number of gates is very large. Thus executing the function at a level higher than gate level is good idea. Dataflow modeling has become a popular design approach as logic synthesis tools have become sophisticated. This approach permits the designer to concentrate on optimizing the circuit in terms of data flow.

Gate level

Within the logic level the features of a system are defined by logical links and their timing properties. All signals are discrete signals. They can only have definite logical values (`0', `1', `X', `Z`) [3]. The usable operations are predefined logic primitives (AND, OR, NOT etc gates). Using gate level modeling might not be a good idea for any level of logic design. Gate level code is generated by tools like synthesis tools and these lists are used for gate level simulation and for backend.

Switch level

In Verilog this is the lowest level of abstraction. A module can be implemented in different forms like as switch, nodes and the interlinked between them. Design for this level requires knowledge of switch level implementation.

2.2.2: Fundamental purposes for choosing Verilog HDL

- It is easy to learn and easy to use, due to its similarity in syntax to the C programming language.
- In the same design different abstraction level can be mixed.
- There is availability of Verilog HDL libraries for post-logic synthesis simulation.
- Most of the synthesis tools support Verilog HDL. This makes the language of selection for most of the designers.
- The Programming Language Interface (PLI) is a powerful feature that allows the user to write custom C code to interact with the internal data structures of Verilog.
- Designers can modify a Verilog HDL simulator to their needs with the PLI.

2.3: An Overview of FPGA

FPGA (Field- Programmable Gate Array) is an integrated circuit (IC) that allow programmer to create their own design or program in the field after manufacture. Basically we can consider an FPGA as a blank state that means designer can create their very own configure file or bit file. The bit file loaded on FPGA will act like the digital circuit the designer designed. There is no processor in FPGA to run the software until the designer build one, it is up to the designer to create a simple file like an and gate to multi-core processor . A Hardware Description Language (HDL) is required for FPGA configuration. The elementary building block of a FPGA is programmable logic block (PLB) or

configurable logic block (CLB). This logic block can similarly be assigned the function to act as computer memory which permits to store certain values and also between this logic blocks there are reconfigurable wiring circuitry. The logic block of a FPGA consist of a 4 input LUT (Look up Table), a programmable multiplexer which assists to choose whether the FPGA be programmed in registered or non-registered output and finally a flip-flop for storage. FPGA is a volatile device, they store their configure file in a RAM so after losing the power they forget about the configure file. Designers have control over the hardware with FPGA.

2.3.1: Reasons for working with FPGA

- **Massively parallel data processing:** The ability of parallel processing is one of the huge advantages of FPGA. Since we have done our thesis on smart sensing system so we needed to interface a lot of sensors in parallel. Processors are usually more functional for routine control of particular circuits and have fixed serial port but in FPGA blocks are allowed to configure at any ports. There are enough spaces for choosing port. Designer can choose any of the port according to their need. FPGA permits to do hall bunches of processing at once and this is the main reason for choosing FPGA for our thesis.
- **Super-fast:** FPGA is a super-fast device. Since the customer can determine the hardware structure of FPGAs so FPGA can be programmed to process larger data with few clock cycle, whereas this is not possible with the processor [20].Because data flow is limited by processor bus (16-bit, 32 bit, etc.) and the processing speed. A basic FPGA have input output blocks and serial functionality works in giga byte per second comparing to microcontrollers and other devices that works in 100 megahertz.
- **Field programmable:** As it is named field programmable so updates and feature development can be carried out even after delivery consumer's site. **.** It is feasible to define and utilize processor and user-specific hardware functions on only one chip by using FPGA. This solution gives engineers the chance to control the hardware because of its great flexibility. Designers can change and update whole design (FPGA on the processor and

other logic circuits) by only changing the code on FPGA, without any change on circuit board layout. Thus if designers face any problem in manufacturing there is a scope to modify it.

- **It can do anything:** It is possible with FPGA to do anything. Designers can build a microcontroller, DSP (Digital Signal Processor) or anything they want. FPGAs have no fixed hardware processor thus it is programmable according to designers desire and able to do anything in digital demand.
- **Extremely short time to market:** Through the use of FPGAs the development of hardware prototypes is drastically accelerated as a huge part of the hardware development process is shifted into ip core design, which can take place in parallel. In addition, because of the early availability of hardware prototypes, time-consuming activities like the start-up and debugging of the hardware are brought onward alongside to the overall development.
- **Performance gain for software applications:** Complex designs are often handled through software implementations in arrangement with high-performance processors. In this case FPGAs offers a spirited alternative, which by means of parallelization and customization for the definite task even establishes a supplementary performance gain
- **Real time applications:** FPGAs are absolutely suitable for applications in time-critical systems. On the contrary to software based solutions with real time operating systems, FPGAs provide real deterministic performance. By means of the featured flexibility even complex computations can be accomplished in enormously short periods.
- **Cost Effective:** In this board we can use all sensors simultaneously and this board has huge number of I/O ports. Thus, by using only this one board we can perform a lot of tasks side by side and at a minimum cost.

CHAPTER-3

Description of Instruments We Used

3.1: FPGA Development Board

3.1.1: Types of FPGA Devices

Device Family	Voltage required
FPGA	
Stratix V, Stratix IV and Stratix III	As specified by V_{CCGM} or V_{CCPD}
Stratix II, Stratix II GX and Stratix GX	As specified by V_{CCEL}
Arria 10	As specified by V_{CCPGM} or V_{CCIO}
Arria V	As specified by V_{CCPD}
Arria II GX	As specified by V_{CCPD} OR V_{CCIO} Of BANK
Arria GX	As specified by V_{CCEL}
Cyclone V	As specified by V_{CCPGM} or V_{CCPD}
Cyclone IV	As specified by V_{CCA} OR V_{CCIO}
Cyclone III	As specified by V_{CCA} OR V_{CCIO}
Max 10	As specified by V_{CCIO}
Configuration	
EPCS	3.3v
EPCQ	3.3v
EPCQ-L	1.8v

Table-3.1 Types Of FPGA Devices

3.1.2: ALTERA Cyclone-iii EP3C5E 144C8N Development Board

We have chosen EP3C5-C board for our thesis.

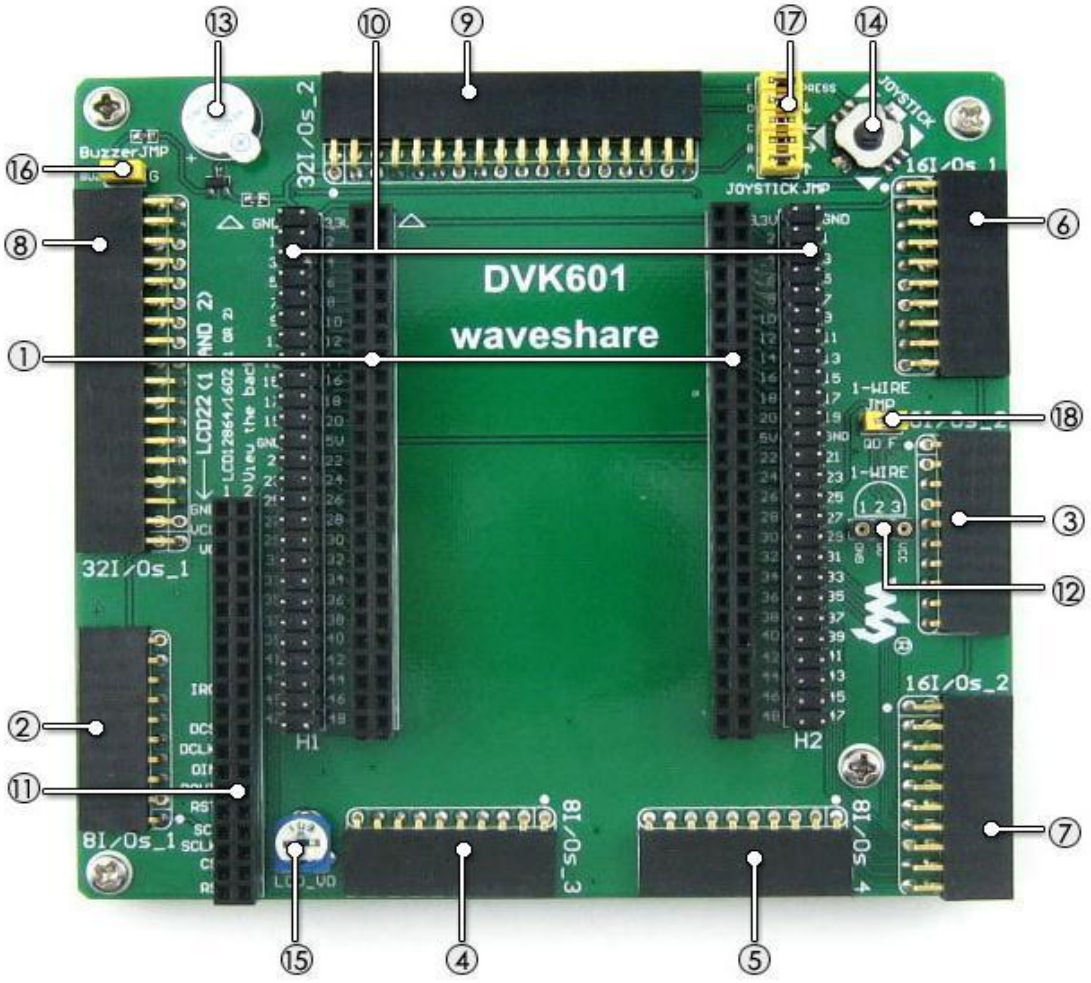

Figure-3.1 ALTERA Cyclone-iii EP3C5E 144C8N Development Board

3.1.3: Pin Configuration of ALTERA Cyclone-iii Board

1. FPGA CPLD core board connector: for easily connecting core boards which integrate an FPGA CPLD chip onboard
2. 8I/Os_1 interface, for connecting accessory boards/modules
3. 8I/Os_2 interface, for connecting accessory boards/modules
4. 8I/Os_3 interface, for connecting accessory boards/modules
5. 8I/Os_4 interface, for connecting accessory boards/modules

6. 16I/Os_1 interface, for connecting accessory boards/modules
7. 16I/Os_2 interface, for connecting accessory boards/modules
8. 32I/Os_1 interface, for connecting accessory boards/modules
9. 32I/Os_2 interface, for connecting accessory boards/modules

All the I/O interfaces above:

- Capable of being simulated as USART, I2C, SPI, PS/2, etc.
- Capable of driving devices such as FRAM, FLASH, USB, Ethernet, etc.

10. FPGA expansion connectors
    - FPGA pins are accessible on expansion connectors
    - for connecting SDRAM accessory board
11. LCD interface, for connecting LCD22, LCD12864, LCD1602
12. ONE-WIRE interface: easily connects to ONE-WIRE devices (TO-92 package), such as temperature sensor (DS18B20), electronic registration number (DS2401), etc.
13. Buzzer
14. Joystick: five positions
15. Potentiometer: for LCD22 backlight adjustment, or LCD12864, LCD1602 contrast adjustment
16. Buzzer jumper
17. Joystick jumper
18. ONE-WIRE jumper

For jumpers 16-18:

- short the jumper to connect to I/Os used in example code
- open the jumper to connect to other custom pins via jumper wires

Core EP3C5:

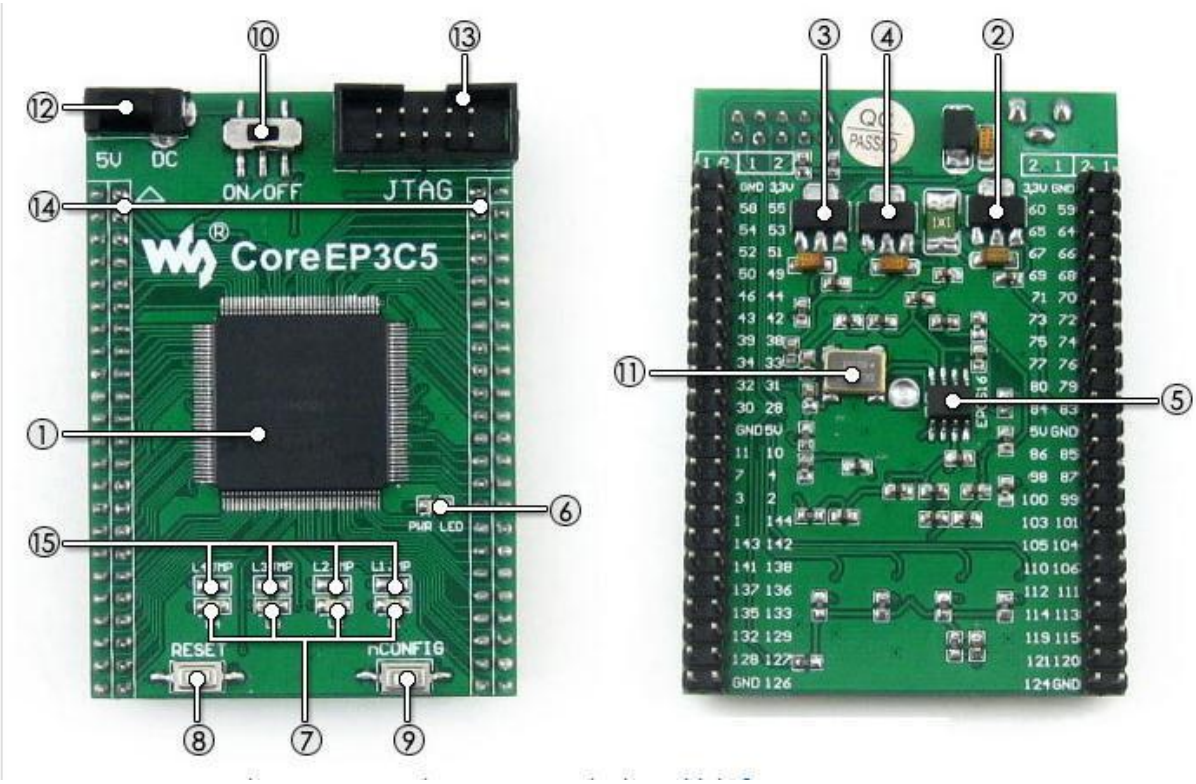

Figure-3.2 Core EP3C5

1. EP3C5E144C8N: the ALTERA Cyclone III FPGA device which features:
    - **Operating Frequency:** 50MHz
    - **Operating Voltage:** 1.15V～3.465V
    - **Package:** QFP144
    - **I/Os:** 82
    - **Les:** 5K
    - **RAM:** 414kb
    - **PLLs:** 2
    - **Debugging/Programming:** supports JTAG
2. AMS1117-3.3, 3.3V voltage regulator
3. AMS1117-2.5, 2.5V voltage regulator

4. AMS1117-1.2, 1.2V voltage regulator
5. EPCS16, onboard serial FLASH memory, for storing code
6. Power indicator
7. LEDs
8. Reset button
9. Nconfig button: for re-configuring the FPGA chip, the equivalent of power resetting
10. Power switch
11. 50M active crystal oscillator
12. 5V DC jack
13. JTAG interface: for debugging/programming
14. FPGA pins expander, VCC, GND and all the I/O ports are accessible on expansion connectors for further expansion
15. LED jumpers, short the jumpers to drive the LEDs [19].

3.1.4: Features

1. Minimum power consumption
2. Design security feature
3. Increased system integration [20].

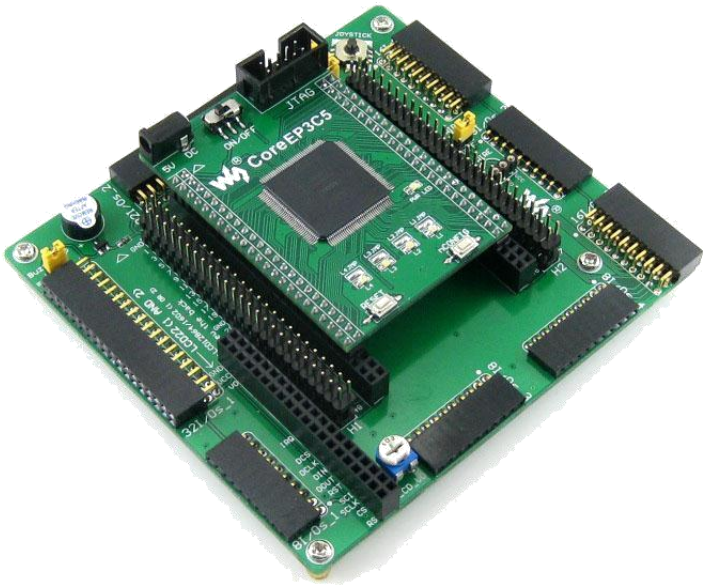

Figure-3.3 Combined FPGA Board

3.2: USB Blaster

The USB blaster download cable interfaces a USB port on a host computer to an Altera FPGA mounted on a printed circuit board. The cable sends arrangement information from the PC to a standard 10-stick header associated with the FPGA. We can utilize the USB Blaster cable to iteratively download configuration information to a system in prototyping or to program data into the system during construction.

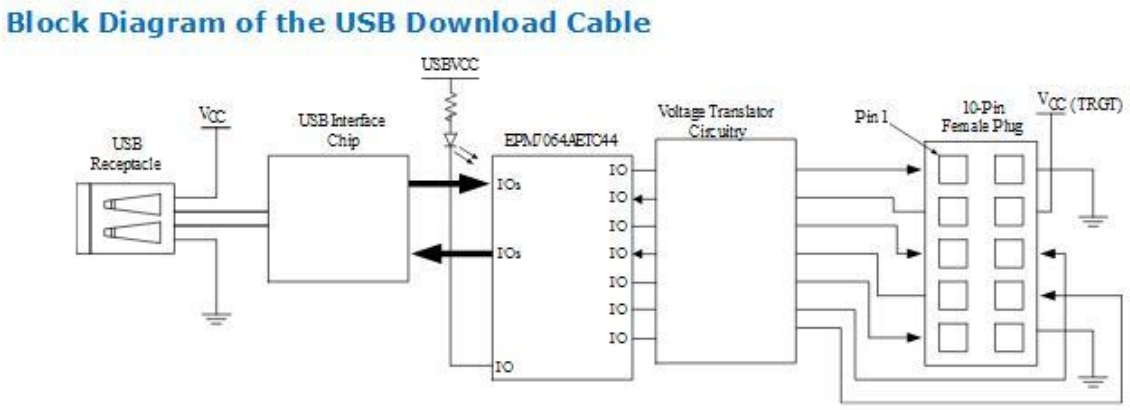

Figure-3.4 Block Diagram of the USB Download Cable

Figure 3.5 USB Blaster

3.2.1: Features

1. High speed
2. Stable
3. Internal FT245R+CPLD designed
4. Supports 1.2-5V programming voltage
5. Supports Signal Tap II embedded logic analyzer
6. Supports Nios II of embedded processor communication and debugging [17].

3.3: Sensors of Our System:

3.3.1: Grove Flame Sensor

The Grove - Flame Sensor can be applied to identify fire source or other light sources of the wavelength in the range of 760nm - 1100 nm. It depends on the YG1006 sensor which is a fast and high sensitive NPN silicon phototransistor. Because of its black epoxy, the sensor is sensitive to infrared radiation.

Figure-3.6 Grove Flame Sensor

Working Mechanism:

A fire sensor identifies the nearness of flame or flares, in extremely dangerous situations, flame sensors work to minimize the dangerous connected with flame. There are a few distinct categories of flame sensors, some will raise an alarm while others may initiate a fire concealment framework or deactivate an ignitable fuel line. In our project we used The Grove- Flame sensor, to mainly detect the fire dangers in houses, offices and even hotels. The grove flame sensor is connected to the Field Programmable Gate Array (FPGA) Board, following the correct pin assignments. After all the pin assignments are done correctly and the simulation is 100% completed the sensor is ready for work. Now whenever the sensor detects any kind of flame in the vicinity, it gives an output of 1 and if it does not detect any flame then it gives out an output of 0.

Figure-3.7 Top and Bottom View of Grove Flame Sensor

Grove flame sensor has the following features like:

- Grove Interface
- High Photo Sensitivity
- Rapid Response Time
- Easy to use
- Sensitivity is adjustable

Specifications:

Item	Min	Typical	Max	Unit
Voltage	4.75	5.0	5.30	VDC
Current	/	20	/	mA
Range ofSpectral Bandwidth	760	940	1100	nm
Detection Range	0	-	1	m
Response Time		15		μS
Operating Temperature	-25	-	85	℃

Table-3.2 Specifications of Grove Flame Sensor

3.3.2: PIR Motion Sensor

Figure-3.8 PIR Motion Sensor

PIR Motion sensors are most complex sensors than many of the other sensors, since there are multiple factors that affect the sensors input and output. There are two slots in the sensor, each slot is made of a special material that is sensitive to IR light.

When the sensor is inactive, both slots detect the same amount of IR, the ambient amount radiated from the room or walls of the room or outdoors. When a warm body like a human or animal passes by, it first stops one half of the PIR sensor, which gives a positive differential change between the two halves. When the warm body leaves the sensing area, the reverse happens, thus the sensor generates a negative differential change. These change pulses are what is detected. PIR sensors allows us to sense motion, it is used to detect whether a human has moved in or out of the sensors range.

PIR Sensors are small, economical, and easy to use and does not wear out. For these reasons they are commonly found in appliances and devices used in homes or businesses

areas. They are often referred to as PIR, "Passive Infrared", "Pyro electric", or "IR motion" sensor.

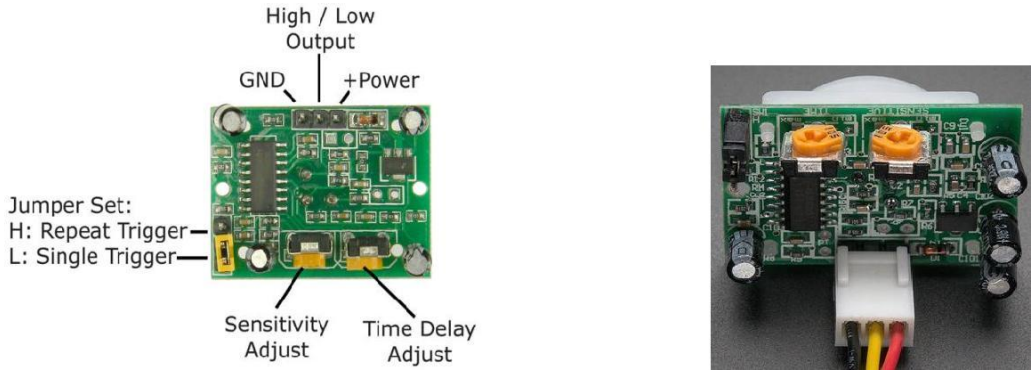

Figure-3.9 Pin Specification Of PIR Motion sensor

For our PIR Motion Sensor:

i. Red cable is + voltage power,
ii. Black cable is - ground power
iii. Yellow is the signal out.

Specifications:

- Digital output: 3.3v (High representing binary 1 when gets motion and less representing 0 without motion)
- Working voltage: 5V-20V DC
- Delay: 0.3s - 18s (adjustable)
- Trigger: H repeatable (default), L unrepeatable.
- Power Consumption: 65mA
- Delay time: Adjustable (.3->5min)
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 ~ +70
- Dimension: 32*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

3.3.3: XINDA Water Sensor:

The Water Sensor indicates whether the sensor is waterless, wet or completely submerged in water by measuring the conductivity. In our project we have used this sensor to detect the level of water mainly in bathrooms, water tanks and for normal household purpose or for office usage, to keep a track if water is overflowing anywhere inside the building compound.

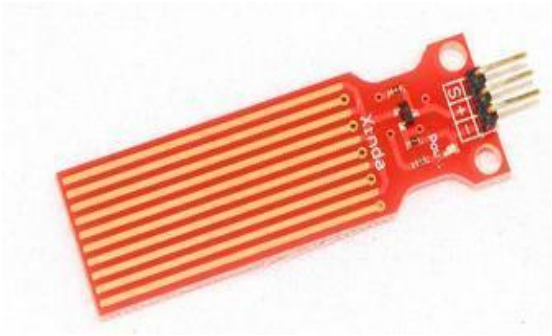

Figure-3.10 XINDA Water Sensor

Features:

- Grove compatible interface
- Low power intake
- 2.0cm x 2.0cm twig module
- High sensitivity

Applications:

- Detecting Rainfall
- Liquid Leakage
- Tank Overflow Detector

Specifications:

- Product name: water level sensor
- Working voltage: DC3-5 v
- Working current: less than 20 ma
- Sensor types: simulation
- Detection area: 40 mm x 16 mm
- Production process: FR4 double-sided tin
- Working temperature: 10 C-30 C
- Working humidity: 10% to 90% without condensation
- Product weight: 3.5 g
- Product size: 62 mm x 20 mm x 8 mm

Pin definition:

- "S" stand for signal input
- "+" stand for power supply
- "-" stand for GND

Chapter- 4

Implementation of the System

4.1: Working procedure of sensors

4.1.1: Flame Sensor

Our first project idea was to make an alarm system that is going to detect any kind of fire. We used the grove flame sensor which gives a high voltage at room temperature when flame is detected. We made a simple module to run an alarm in occurrence of any fire. In this module we used gate level synthesis to get the job done. We adjusted the threshold that can detect a flame from match stick. The FPGA output gets into a buzzer.

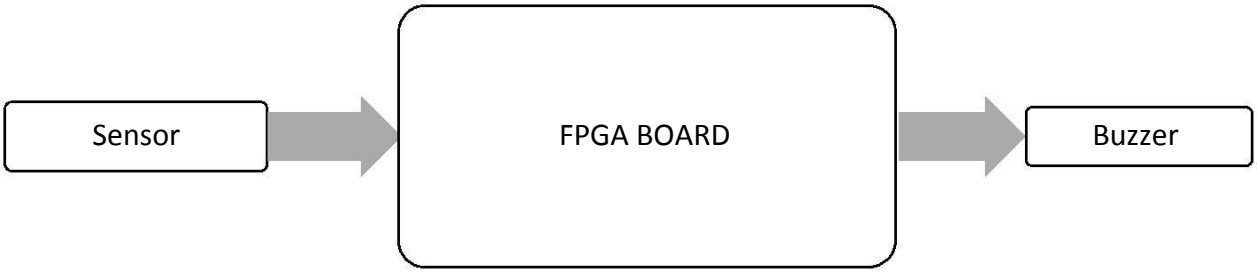

Figure-4.1 Working Procedure of Flame Sensor

4.1.2: Water sensor

Our second project idea was to create a water overflow system. Water sensor gives the output voltage according to the water level and the voltage starts from around 2.7 volts and increases up to 3.7~3.8 due to certain pressure of water at room temperature. We set up a threshold voltage of 2.85 to detect water. We used gate level synthesis to do this sensor work.

4.1.2.1: Comparator

As this sensor is analog in nature it was necessary to make the output digital. As FPGA doesn't take analog signal and cannot process we used a comparator circuit that led the output high when water is detected. Using an lm358 operational amplifier we made up a little comparator. We put the threshold voltage as the inverting output and the sensor

output as the non-inverting. When the sensor output crosses the threshold, it will send high signal or binary 1 to the board. The FPGA output gets into a buzzer.

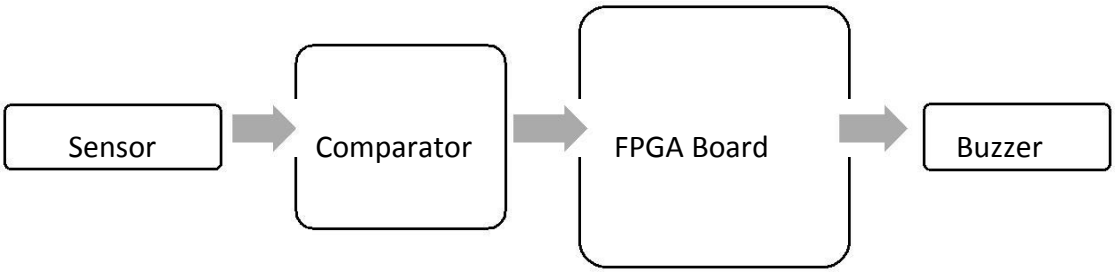

Figure-4.2 Working Procedure of Water Sensor

4.1.3: PIR motion sensor

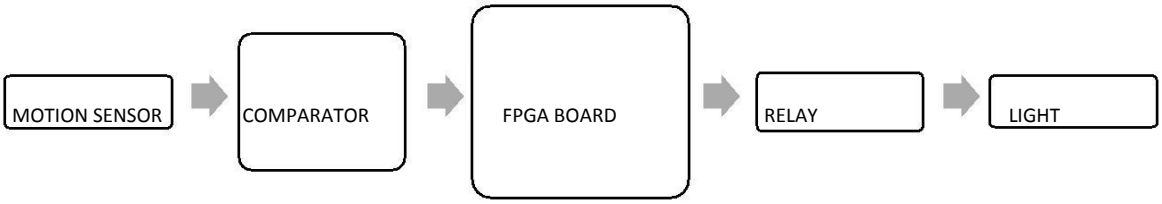

Figure-4.3 Working Procedure of PIR Motion Sensor

PIR motion sensor is sensitive to infrared light to detect any kind of motion. To configure this sensor to a simple module we made a behavioral level of abstraction so that it will give a output voltage corresponding to motion. Getting any motion causes 3.3 volts and no motion means 0 in room temperature. Our plan was then to add a sequential timing constraint to measure the period of motionless environment. It is to make an energy saving system that prevents unnecessary power consumption. To make this work we created a big module consisting of sub-modules that are instantiated within one schematic. There are 3 sub-modules under the mother module.

4.1.3.1: Motion module

Detecting motion it will deliver respective output where motion gives a high voltage and otherwise 0 voltages to the counter module. Motion sensitivity was fixed at the midpoint so that it only covers a small room for experimental convenience.

4.1.3.2: Counter module

If there is motion it doesn't need to count as there is no waste of energy. If there is no motion we need to measure the duration. Here we r using counter as the timing constraint because delays are impractical in this case(cannot be implemented in hardware) and we have a very high frequency clock signal from the on board built in oscillator. Being very fast in running programs this core ep3c5 runs the module for 402653840 times which represents binary 11110000000000000000000000000000. Using a 32 bit counter and sending the output through a multiplexer we determine the value of counter or the duration. We can change the duration by adjusting the delay potentiometer. We set it to the least delay happening and took a duration of 3 minutes 23 seconds for convenience of implementation. Getting no motion for this time span will switch off the light automatically.

Figure-4.4 Block Diagram Of Counter Module

4.1.3.3: Control light module

The counter value will go to the control light module. When its read 11110000000000000000000000000000, it will send a low control signal to the light. So the light will automatically switch off.

4.1.3.4: Relay

Output from FPGA switches the relay and the relay operates light.

4.1.3.5: Comparator:

Though this sensor gives a high voltage as output, sometimes it flickers about some milivolts which occurs a fluctuating current. To solve any kind of disruption and to make the output smoother we used a comparator circuit where 3.1 is the threshold voltage. Whenever the sensor supplies a voltage greater than this it means some motion occurred and it's taken as binary 1.

4.2: Demo door lock

We made a simple door lock system which takes a 5 bit binary input as a key and opens the door at the correct combination. In this system it is set to 10001.

4.3: Combined parallel compilation

Our first and foremost goal was to operate multiple systems simultaneously in parallel at one time. FPGA gives us the privilege to make these ideas work together. To run them together we had to create a master module which enslaves these slave modules. Instantiating these modules we got them on one track those results in the three sensors sensing simultaneously and giving out the results with same efficiency, speed and accuracy. Adding the project files into one master project, using multiple i/o ports for hardware configuration and merging them into the master module was all we had to do.

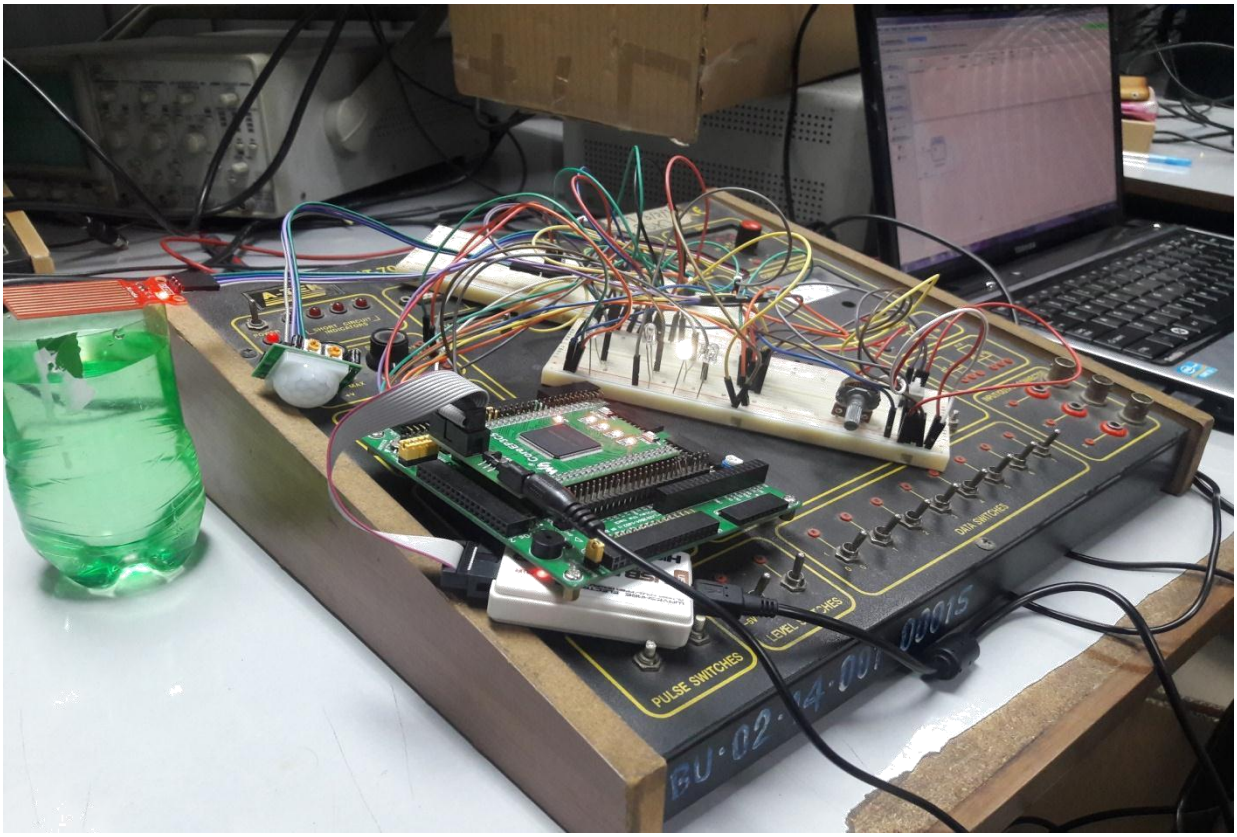

Figure- 4.5 Set up of the Whole System

4.3.1: Pin assignments

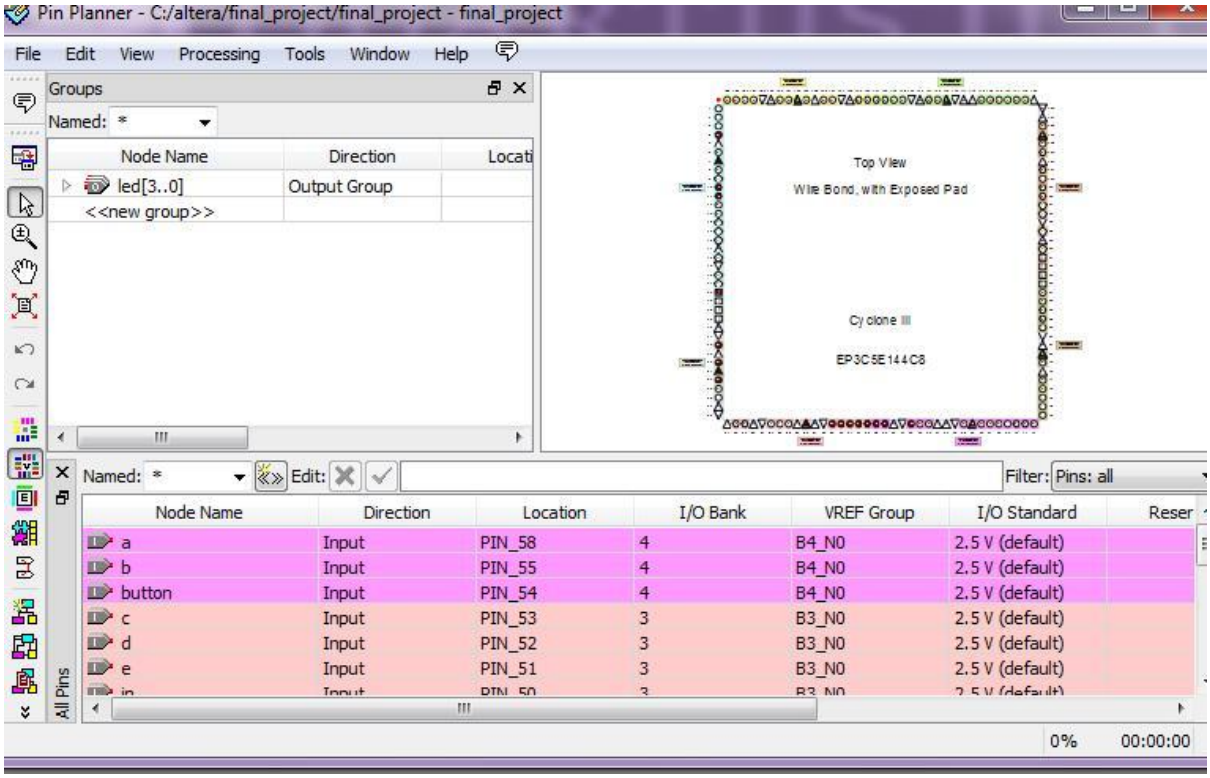

Figure-4.6 Pin Assignment of Total System

CHAPTER-5

Results

After merging the modules and hardware connection the system worked as it was meant to. Our experiment was successful. Here are the compilation reports.

5.1: Compilation reports

- Compilation of flame sensor:

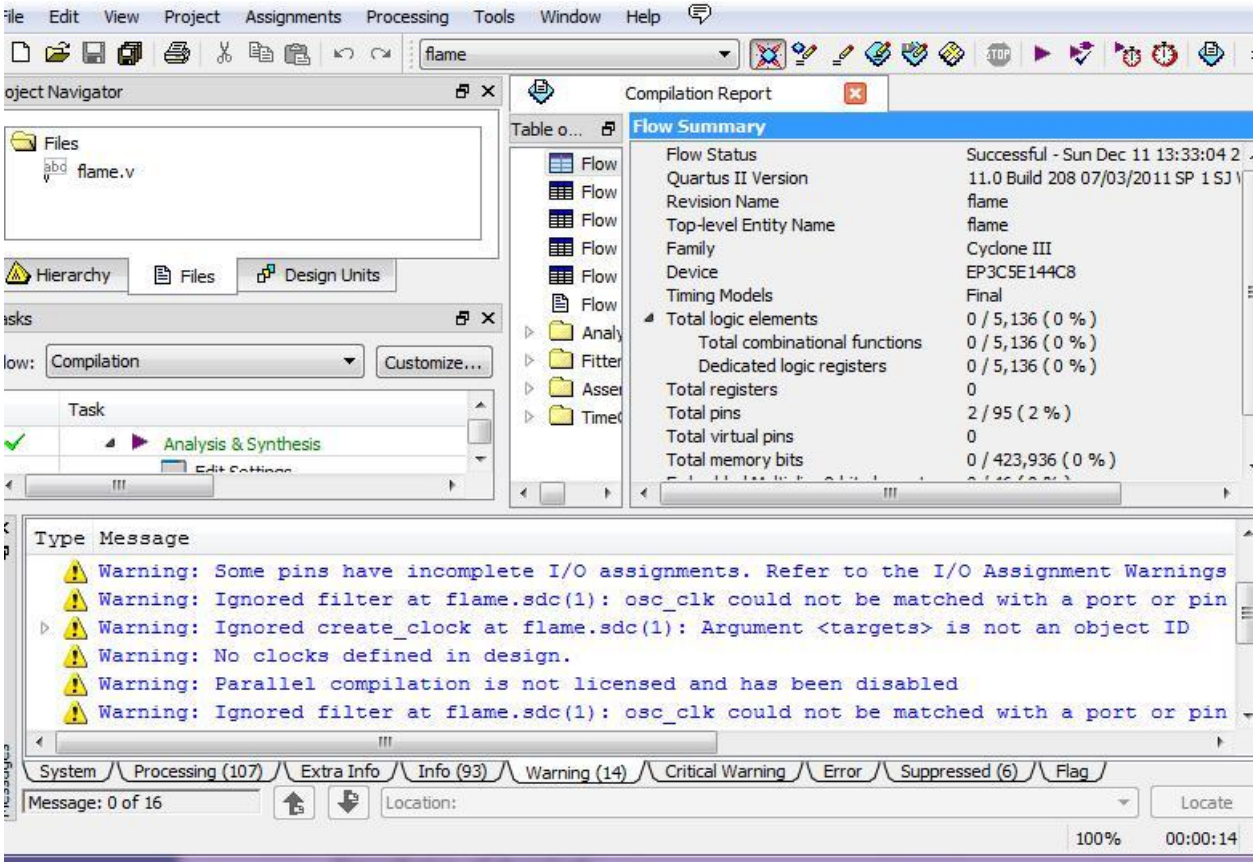

Figure-5.1 Compilation Report of flame sensor

- Compilation of water sensor:

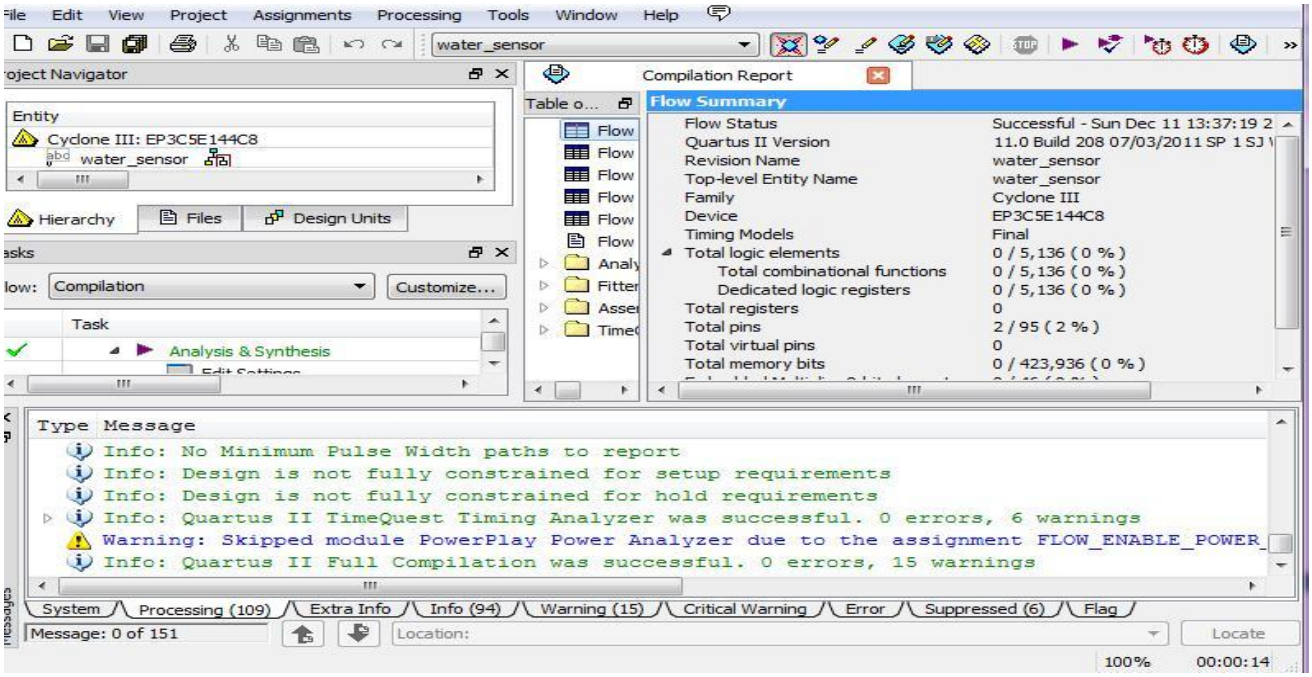

Figure-5.2 Compilation Report of Water sensor

- Compilation of motion sensor:

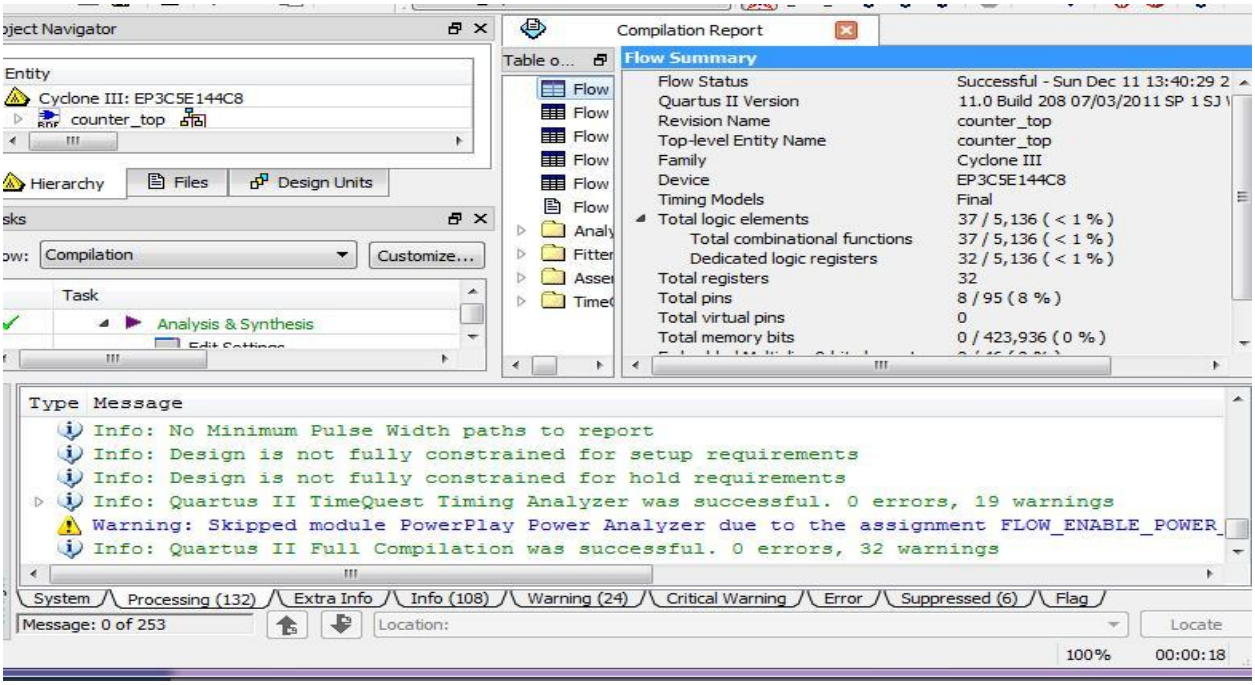

Figure-5.3 Compilation Report of Motion sensor

- Compilation of door lock:

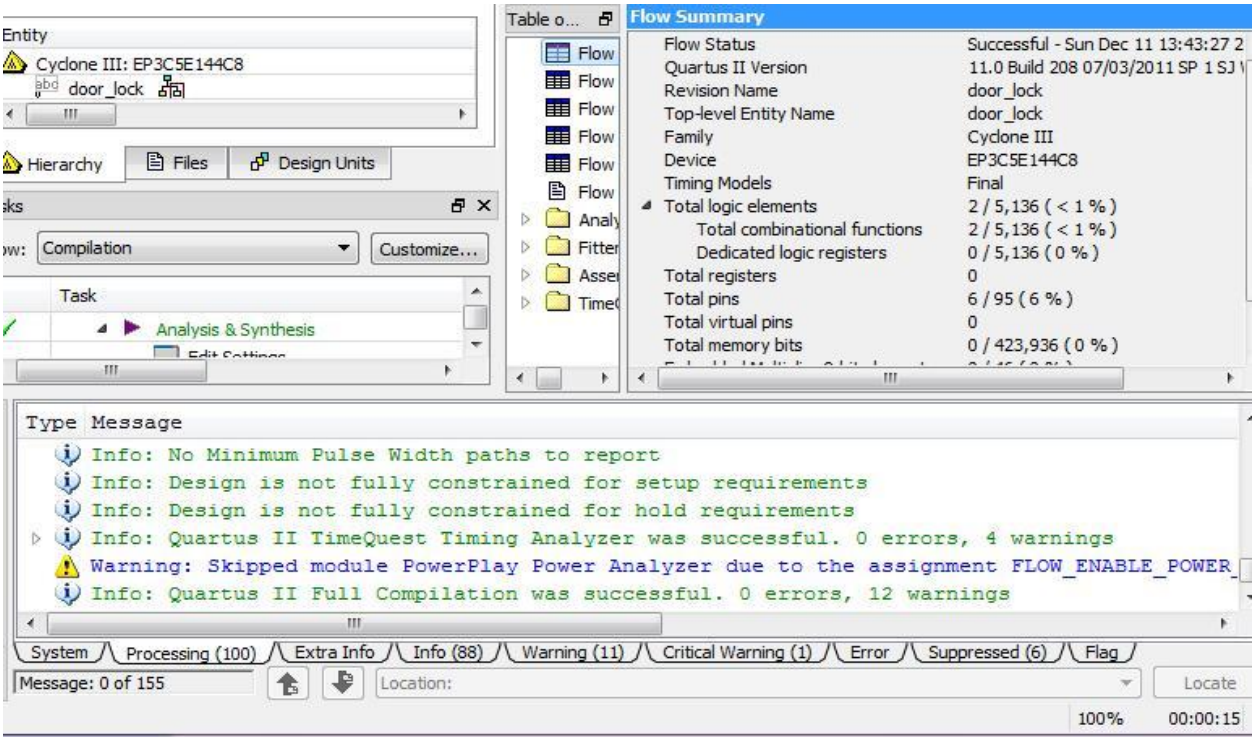

Figure-5.4 Compilation Report of Door Lock

- Parallel compilation report:

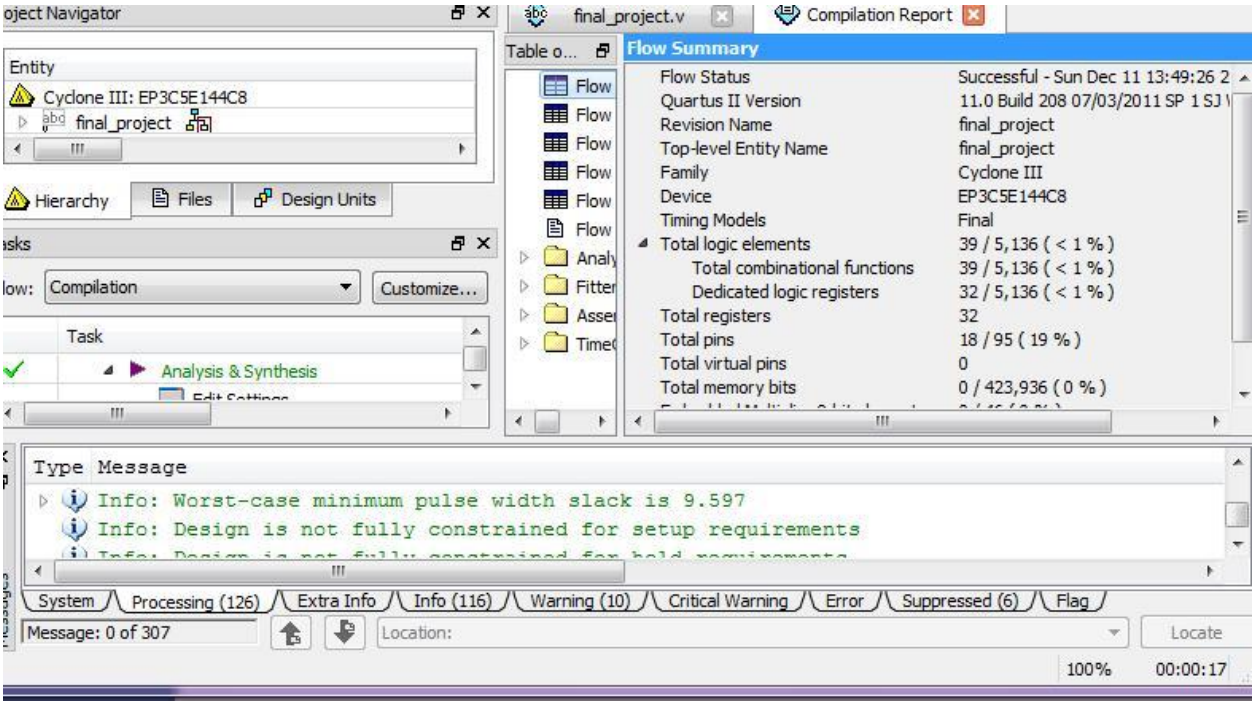

Figure-5.5 Parallel compilation report

5.2: RTL (resistor transfer level) schematic view:

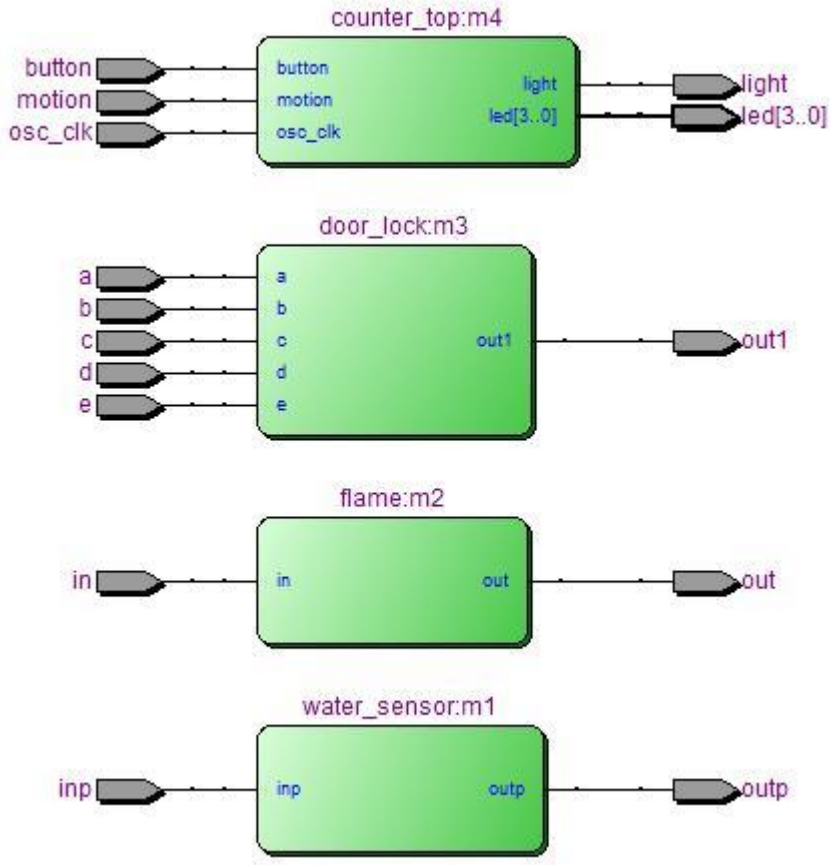

Figure-5.6 RTL View

CHAPTER-6

Challenges and Future Scope

6.1: Challenges

During this project we confronted several challenges where we overcome some of them.

Hardware implementation constraints: Learning the very new language within a short period was a challenge. As a hardware description language it has different level of sensitivity and limitations to hardware implementation. A big portion of this language is not synthesizable. Yet we had to find those appropriate for the application.

- **Choosing suitable board**: Among all the boards we had to find which is good for our project.
- **Appropriate level of abstraction:** Extracting the appropriate level of abstraction considering hardware efficiency and speed was also a challenge.
- **Limited scope for analog:** As a digital device there is no space for analog operation.
- **Unavailable FPGA board:** Because of scarcity of Verilog project we confronted severe unavailability of hardware. We used a prototype version of the original board.
- **Software licensing and installation:** It is used industrially. As a result without authentic industrial buyers it's quite impossible to get the software license. Without licensed version it's not possible to install USB-blaster.

6.2: Future scope of advancement

We can improve current features and also implement new features.

Developed features:

Automated fan/air conditioner
Bluetooth/GPRS notifications
Door lock security
Gas leakage detection
Automated moister sensing/plant watering system

Chapter 7

CONCLUSION

This thesis discussed the implementation of some significant sensors which support to detect and notify any stimuli like fire, motion and water to the host. The computation increased as the complexity of more sensors working in parallel, but FPGA has the inheritable parallelism function which helped us to do it sophistically. We used Verilog HDL language in FPGA as it is the best alternative to do repetitive function which can be off load onto FPGA.

References:

1] N. S. Pvt, "Learning FPGA and Verilog A beginner's guide - introduction," Numato Lab Documentation Portal, 2016. [Online]. Available: https://docs.numato.com/kb/learning-fpga-verilog-beginners-guide-part-1-introduction/. Accessed: Jul. 12, 2016.

[2]"Introduction,". [Online]. Available: http://www.asic-world.com/verilog/intro1.html. Accessed: Oct. 12, 2016.

[3] "Introduction to Verilog abstraction levels (theory): FPGA & embedded systems lab: Computer science & engineering: COE PUNE virtual lab,". [Online]. Available: http://coep.vlab.co.in/?sub=29&brch=88&sim=1407&cnt=1. Accessed: Nov. 12, 2016.

[4] A. R. Reserved, "The history of smart homes," 2016. [Online]. Available: http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm. Accessed: Dec. 12, 2016

[5] [Online]. Available: https://www.icontrol.com/wp-content/uploads/2015/06/Smart_Home_Report_2015.pdf. Accessed: Sep. 19, 2016

[6] "Grove - water sensor,". [Online]. Available: https://www.seeedstudio.com/grove-water-sensor-p-748.html?cPath=25_27. Accessed: Nov. 01, 2016

[7] R. Dubey, Introduction to embedded system design using field programmable gate arrays. London: Springer London, 2008

[8] T. Bangladesh, "Grove - water sensor - water - Techshop Bangladesh," 2012. [Online]. Available: https://www.techshopbd.com/product-categories/water/1183/grove-water-sensor-techshop-bangladesh. Accessed: nov 5, 2016

[9] T. Bangladesh, "PIR motion sensor - motion - Techshop Bangladesh," 2012. [Online]. Available: https://www.techshopbd.com/product-categories/motion/1204/pir-motion-sensor-techshop-bangladesh. Accessed: nov 15, 2016

[10] "Grove - flame sensor,". [Online]. Available: https://www.seeedstudio.com/Grove-Flame-Sensor-p-1450.html?cPath=25_27. Accessed: nov 1, 2016.

[11] . [Online]. Available: https://www.techshopbd.com/uploads/product_document/HCSR501_pir_motion_sensor.pdf. Accessed: nov 3, 2016.

[12] [Online]. Available: http://d1.amobbs.com/bbs_upload782111/files_33/ourdev_585395BQ8J9A.pdf. Accessed: Dec. 12, 2016

[13] M.Andrew, FPGAs for dummies, New Jersey: John Wiley & Sons,2014

[14] . [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/tt/tt_my_first_fpga.pdf. Accessed: Oct. 15, 2016

[15] Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_usb_blstr.pdf. Accessed: Dec. 12, 2016.

[16] "USB Blaster V2 - Waveshare Wiki,". [Online]. Available: http://www.waveshare.com/wiki/USB_Blaster_V2. Accessed: Dec. 12, 2016

[17] "USB Blaster ALTERA FPGA CPLD download cable programmer debugger,". [Online]. Available: http://www.waveshare.com/product/USB-Blaster-V2.htm. Accessed: Dec. 12, 2016.

[18] [Online]. Available: http://www.waveshare.com/product/OpenEP3C5-C-Standard.htm. Accessed: Dec. 12, 2016.

[19] "OpenEP3C5-C - Waveshare Wiki,". [Online]. Available: http://www.waveshare.com/wiki/OpenEP3C5-C. Accessed: Sep. 12, 2016.

[20] F. Architecture and the Challenge, "Field-programmable gate array," in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Field-programmable_gate_array. Accessed: Oct. 12, 2016.

Appendix

Verilog code for flame sensor:

```
module flame(out,in);

input in;

output out;

not b(out,in);

endmodule
```

Verilog code for water sensor:

```
module motion(out1,in1);

output out1;

input in1;

reg out1;

        always @(in1)

begin

if(in1)

          out1 <=

1'b1; else out1 <=

1'b0; end


endmodule
```

Verilog code for door lock:

```
moduledoor_lock (a,b,c,d,e,out1);

inputa,b,c,d,e;

output out1;

wire out1;

wirea,b,c,d,e;

assign out1 = a?(b?(0):(c?(0):(d?(0):(e?(1):(0))))):(0);

endmodule
```

Verilog code for motion sensor:

- **Verilog code for counter module:**

```
module counter(clk,motion,count);

input motion;

inputclk;

outputreg[31:0] count;


always

@(posedgeclk) begin

if(motion==0)

begin

count<= count+1;

end
```

```
else

begin

count<=0;

end

end

endmodule
```

- **Verilog code for control light module:**

```
module light(light,led,);

output light;

wire out2;

input [3:0]led;


assign out2=led[3]?(led[2]?(led[1]?(led[0]?(1):(0)):(0)):(0)):(0);

not n(light,out2);

endmodule
```

Verilog code for merging:

```
module final_project(out,outp,in,inp,a,b,c,d,e,out1,osc_clk,motion,button,led,light);

input in,inp,a,b,c,d,e,osc_clk,motion,button;

output out,outp,out1,led,light;

wire[3:0]led;
```

```
water_sensorm1(.outp(outp),.inp(inp));

flame m2 (.out(out),.in(in));

door_lock m3(.a(a),.b(b),.c(c),.d(d),.e(e),.out1(out1));

counter_top m4(.osc_clk(osc_clk),.motion(motion),.button(button),.led(led),.light(light));


endmodule
```

HC-SR501 PIR MOTION DETECTOR

Product Description

HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

Specification:

- Voltage: 5V – 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- Delay time: Adjustable (.3->5min)
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 ~ +70
- Dimension: 32*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

Application:

Automatically sensing light for Floor, bathroom, basement, porch, warehouse, Garage, etc, ventilator, alarm, etc.

Features:

- Automatic induction: to enter the sensing range of the output is high, the person leaves the sensing range of the automatic delay off high, output low.
- Photosensitive control (optional, not factory-set) can be set photosensitive control, day or light intensity without induction.
- Temperature compensation (optional, factory reset): In the summer when the ambient temperature rises to 30 ° C to 32 ° C, the detection distance is slightly shorter, temperature compensation can be used for performance compensation.
- Triggered in two ways: (jumper selectable)
  - non-repeatable trigger: the sensor output high, the delay time is over, the output is automatically changed from high level to low level;
  - repeatable trigger: the sensor output high, the delay period, if there is human activity in its sensing range, the output will always remain high until the people left after the delay will be high level goes low (sensor module detects a time delay period will be automatically extended every human activity, and the starting point for the delay time to the last event of the time).
- With induction blocking time (the default setting: 2.5s blocked time): sensor module after each sensor output (high into low), followed by a blockade set period of time, during this time period sensor does not accept any sensor signal. This feature can be achieved sensor output time “and” blocking time “interval between the work can be applied to interval detection products; This function can inhibit a variety of interference in the process of load switching. (This time can be set at zero seconds – a few tens of seconds).
- Wide operating voltage range: default voltage DC4.5V-20V.
- Micropower consumption: static current <50 microamps, particularly suitable for battery-powered automatic control products.
- Output high signal: easy to achieve docking with the various types of circuit.

Adjustment:

- Adjust the distance potentiometer clockwise rotation, increased sensing distance (about 7 meters), on the contrary, the sensing distance decreases (about 3 meters).
- Adjust the delay potentiometer clockwise rotation sensor the delay lengthened (300S), on the contrary, shorten the induction delay (5S).

Instructions for use:

- Sensor module is powered up after a minute, in this initialization time intervals during this module will output 0-3 times, a minute later enters

the standby state.

- Should try to avoid the lights and other sources of interference close direct module surface of the lens, in order to avoid the introduction of interference signal malfunction; environment should avoid the wind flow, the wind will cause interference on the sensor.
- Sensor module with dual probe, the probe window is rectangular, dual (A B) in both ends of the longitudinal direction
  - so when the human body from left to right or right to left through the infrared spectrum to reach dual time, distance difference, the greater the difference, the more sensitive the sensor,
  - when the human body from the front to the probe or from top to bottom or from bottom to top on the direction traveled, double detects changes in the distance of less than infrared spectroscopy, no difference value the sensor insensitive or does not work;
- The dual direction of sensor should be installed parallel as far as possible in inline with human movement. In order to increase the sensor angle range, the module using a circular lens also makes the probe surrounded induction, but the left and right sides still up and down in both directions sensing range, sensitivity, still need to try to install the above requirements

HC-SR501 PIR MOTION DETECTOR