

# Real-Time Traffic Light Controller System based on FPGA and Arduino

S.L.Qaddori and N.T.Gadawe

{[sahar.qaddoori@uoninevah.edu.iq](mailto:sahar.qaddoori@uoninevah.edu.iq), [noor.gadawe@uoninevah.edu.iq](mailto:noor.gadawe@uoninevah.edu.iq)}

Electronic Engineering Department, Electronics Engineering College, Ninevah University

**Abstract.** This paper aims to implement and simulate the real-time traffic light system by using an FPGA programmed on the Spartan 3E platform and a micro-controller programmed on the Arduino Mega platform. The system is designed to manage street traffic control and assist walkers to move freely to prevent automobile crashes. The simulation of a traffic light system is executed using an FPGA joined to an electronic circuit board on Spartan 3E through the external expansion and also using an Arduino Mega platform linked to the same electronic circuit board. During the simulation of the traffic light system, C++ programming language is used to programme The Arduino platform while the VHDL is used to programme the FPGA platform. Furthermore, some semiconductor components such as switches, resistors, sensors, and LEDs are used to build the electronic circuit to obtain the best performance of the traffic lights. The comparison between the two designs is done dependant on some parameters such as cost, speed, memory space, and other parameters which are considered very important when designing the traffic light system. To solve congestion problems at intersection roads, it is necessary to create a simple and reliable traffic control system.

**Keywords:** Traffic Light Controller (TLC), VHDL, Spartan 3E, FPGA, C/C++programing, Arduino.

## 1 Introduction

In many of the cities and towns all over the globe, traffic congestion is a serious predicament. It has been causing many challenges and setbacks in most major capitals all over the world. Moreover, due to traffic jams, the productivity of traders, suppliers and workers are all affected thus raising the prices of goods [1] [2] [3]. Heavy congestion occurs because of the lack of organisation on each road. Another problem appears when there is no congestion, but the waiting still goes on. The solution is to regulate the delay time of the traffic lights and detect the level of congestion. This problem requires an evaluation of the situation and switching to manual control of the traffic [4],[5],[6]. This work aims to suggest that a traffic light system will provide a solution at a low cost.

A traffic light controller (TLC) can be made using an FPGA and a microcontroller. There are advantages and disadvantages to each. Some of the parameters that are used to compare between them are; the number of input/output ports, performance, speed, and cost all of which are very significant in TLC design [7].

In this paper, a real-time traffic light controller system is simulated and implemented by an FPGA chip and Arduino environment. Comparison between the two designs is done to find the best choice to develop the TLC. The paper's outline is structured as follows: Section 2 covers the literature review of traffic light systems which are created using an FPGA and Arduino. The proposed traffic light controller system is the subject of section 3. The simulation and results with a discussion of the proposed traffic light system beside the hardware implementation of it on Xilinx Spartan 3E FPGA and Arduino are demonstrated in section 4. Section 5 includes a comparison between the two types of platforms (FPGA and Arduino) based on the most important parameters. Finally, section 6 provides the conclusions of this paper.

## **2 Related Works:**

A wide variety of research has been completed concerning traffic light systems aiming to overcome complex traffic phenomenon but most of them fail to deal proficiently with the complex, time-changing traffic controller and conditions can't satisfy real-time traffic signals [2] [8]. The literature reviewed elucidates some of the recent research based on FPGA and Arduino.

### **2.1 Literature Survey based on FPGA:**

In 2014, Surabhi S. et al. [9] presented an adaptive TLC which customized numbers of signals for different junctions and a user-defined number of junctions. The suggested system prototype is created using an FPGA and typical TLCs are designed using the paradigm of finite-state and depend heavily on software design flow. By using the structural style of VHDL programming, the hardware design has been modelled to obtain more robustness.

In 2015, V. V. Dabahde et al. [10] proposed the Intelligent TLC system to decrease waiting times of vans at traffic signals. The suggested system makes use of the FPGA platform along with traffic sensors to regulate traffic according to traffic flow and thus decreases the time of waiting at a junction of two ways. By using ALTERA Cyclone II- FPGA, the system has been effectively implemented and tested in hardware. The system has numerous benefits over the existing TLC.

In 2016, Ali Kareem Abdulrazzaq [11] designed an intelligent TLC system using VHDL and FPGA. Furthermore, multiple levels of functionality such as the motion sensor handling part, special request implementation, adding a standby control signal, and overloaded traffic design have been taken into deliberation. Using ModelSim, the system function was simulated and tested.

In 2017, P. Giri Prasad et al. [12] designed an intelligent transportation system (ITS) using VHDL and implemented on FPGA. It determines the traffic on each road by using sensors. Using the traffic status, the signal time can be managed and in this way, traffic on the road can be handled. On each particular junction, the IR sensors were placed to detect the traffic density and give the current traffic status particular at the junction. By using Model Sim, the function of the system was verified and simulated.

In 2019 [13], we proposed a smart TLC system where the design and implementation were based on FPGA Spartan 3E which is programmed using VHDL language. Four types of junctions have been chosen and are controlled based on whether traffic is heavy (main road) or light (a side road). A pedestrian request button is added and a sensor for the side street to sense the congestion. Furthermore, the design includes switches to manually regulate the traffic lights. The junction controller has four timing parameters which can be manually modified. The system

has been verified successfully using the Xilinx ISE 14.7i software environment and Chip-Scope. The proposed system is simple to use and is low cost when compared with others. Lastly, this paper is extended to the current paper.

## **2.2 Literature Survey based on Arduino:**

In 2014, Ganiyu R. et al. [14] developed a traffic light system for street junction control using a micro-controller to direct the movement of vehicles meeting at a road intersection without any crash. To do this, the micro-controller assigns times for each direction. Vans from one direction will stop and the other vans along the other lane will move. The traffic light timing is constant.

In 2016, K. K. Modak et al. [15] proposed a new technique called “Smart Traffic Control System”. This technique is used to make the controller of traffic lights more proficient. The green and red light timings will be cleverly determined depending on the traffic on neighboring highways. Furthermore, the new system has a clever traffic control system which permits the passing for the emergency vans such as fire brigade, ambulance, etc.

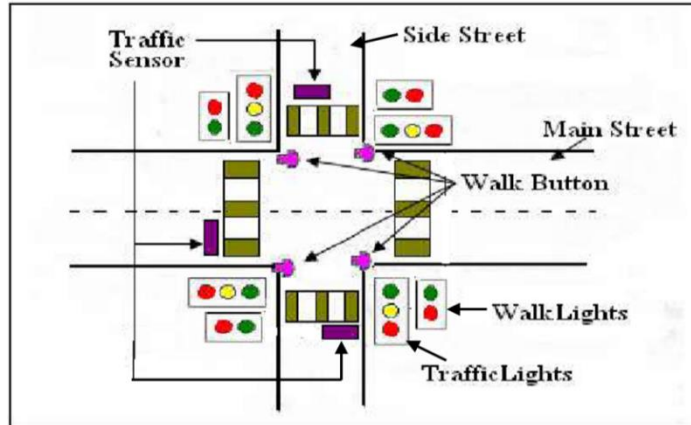
In 2017, S. Ghosh et al. [16] demonstrated a scheme to create a dynamic street signal depending on intensity. To discover traffic intensity at the junction, the sync signal spontaneously mutates. The scheme contained an Arduino Uno and sensors’ interface which spontaneously mutates the synchronization of the unity to accommodate traffic gradually thus escaping needless waiting at the junction. The infrared proximity sensors are used in this paper in the vision line configuration to discover the traffic intensity. The vehicle intensity is scaled depending on the times allocated as a result.

In 2018, Sh. Sahu et al. [17] observed the improvement of a traffic light controller in a metropolitan area using the Arduino platform. The traffic controller used IR sensors (transmitter and receiver) which attached to either side of the road on poles. It gets stimulated and collects the signal as the vehicles pass nearby it. The limitation of this model was that the IR sensors occasionally may ingest normal light as well.

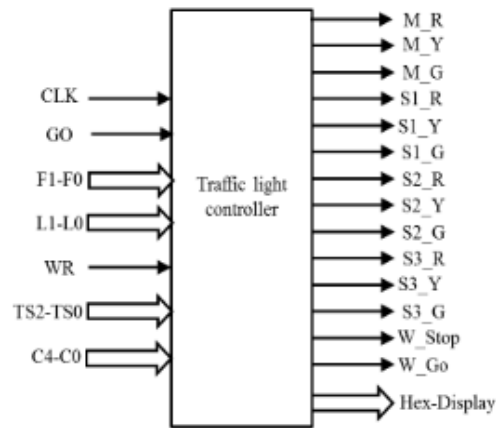
## **3. The Proposed Traffic Light Controller System**

The structure of the chosen traffic light design for four highway junctions (one main street and three side streets) is illustrated in figure 1. Generally, traffic light systems contain three lights (yellow, red, and green). The red light means stop, the yellow light means that in a few seconds the traffic is going to be stopped, the green light means the traffic can go and yellow and red lights at the same time mean that in few seconds the traffic is going to be moving. The intersection is fitted with traffic sensors for the side streets.

Moreover, the traffic light controller has the facility for a pedestrian crossing light. This facility has only 2 lights, red meaning pedestrians must stop for traffic and green meaning that they can go. Figure 2 displays a block diagram of the traffic light controller system.



**Fig.1.** The top-view of the traffic light model for four roads



**Fig.2.** the traffic light controller system block diagram

There are four timing parameters in this system as shown in Table 1a. They are; the base interval (TBASE) for side green, the time for yellow light (TYEL), an extended interval for the main green and a walk green light (TEXT), and a blink interval (TBLINK). The user can specify the four timing parameters using two switches (L0, L1) manually.

The user can execute one of four possible functions specified by two function switches (F0, F1). These functions are listed in Table 1b. The system will stay in the reset mode (the lights are turned off and the system does not do anything) until the GO button is pressed.

Using the writing function, the user can specify any one of the four timing parameters as shown in Table 1a using the L1 and L0 switches. The value of the parameter is set using the (C4\_C0) switches. For the reading operation, the user can use the same L1 and L0 switches to denote which of the four timing parameters to view on a set Hex Display. In normal mode or blinking mode, the system just cycles through the various traffic light states. As shown in Table 2, the designed controller has been modelled with nine states without taking the traffic sensors and pedestrian request into account.

**Table 1.** (a) Time operations for TLC system  
(b) Modes operations for TLC system

F1	F0	Mode Type
0	0	Read Mode
0	1	Write Mode
1	0	Normal Mode
1	1	Blinking Mode

L1	L0	Time Type
0	0	TBASE
0	1	TEXT
1	0	TYEL
1	1	TBLINK

In normal mode (shown in Figure 4b), the main road has a longer green interval than the side road, but if there is congestion on the side road when the controller is about to switch the green light off, it will extend the green light by the shorter (side street) green interval. Thus the green light on the side street will stay on until traffic on the side street clears. The traffic sensor switch is used to simulate waiting traffic on the side street. The system complies by keeping the side street green until the traffic sensor is switched off. After finishing the main street yellow interval, and then only if the pedestrian request button is pressed, the walk light will turn on. Late at night or when something in the system is not working, the blinking mode will activate which means the lights will blink on and off, alternating between side red, main yellow and side yellow, main red.

**Table 2.** Operations of the traffic light controller system in normal mode without using the traffic sensors and walk request

Time Type	TBASE	TEXT	TYEL	TBASE	TYEL	TBASE	TYEL	TBASE	TYEL
Main Street	Green	Green	Yellow	Red	Red	Red	Red	Red	Yellow/Red
Side 1 Street	Red	Red	Yellow/Red	Green	Yellow	Red	Red	Red	Red
Side 2 Street	Red	Red	Red	Red	Yellow/Red	Green	Yellow	Red	Red
Side 3 Street	Red	Red	Red	Red	Red	Red	Yellow/Red	Green	Yellow

## 4. Designing of Traffic Light Controller System

In this section, the simulation and realization of TLC using FPGA and Arduino are demonstrated. Discussed is the simulation and its results, as well as the verification of the hardware implementation on each type of platform.

### 4.1 Traffic Light Controller Based on FPGA:

Nowadays, one of the most successful of today's technologies for creating systems is an FPGA that needs a real-time process. It is a reconfigurable integrated circuit that deals with two-dimensional matrixes of flip-flops and logic blocks by an electrically programmable interconnection among them. Rapid prototyping and updates for hardware devices even aftermarket launch are enabled by the reconfiguration property [18], [19].

#### 4.1.1 Simulation of Traffic Light System:

Figure 3 displays a TLC system block diagram. The design system is composed of a finite state machine (FSM), data storage (D\_RAM), timer, divider, and various synchronizers (latch, and synchronizer) as declared in the following:

- Finite State Machines (FSM) are the core of the traffic light controller system. This FSM controls the loading of static data storage locations with timing parameters, reading RAM locations to display these parameters, and the control of the actual traffic lights.
- D\_RAM: This component is used to store the four timing parameters which are declared in Table 1. Depending on the signal en\_w, which select to read the contents by L0-L1 switches, or write new timing parameters by C0-C4 switches and display the contents on the HEX\_LEDs.
- A divider is used to generate the clock (1 MHz) for an overall system from 50 MHz of the chip FPGA Spartan 3E.
- Sec\_pulse is used to generate a one-second clock, which is used in the timing of the traffic lights.
- A timer is implemented as a counter
- Latch: A pedestrian signal is latched so that when the user pushes the button once, the signal is queued until the FSM needs it.
- Sensors there are three traffic sensors which are synchronized by simply passing it through a flip flop.

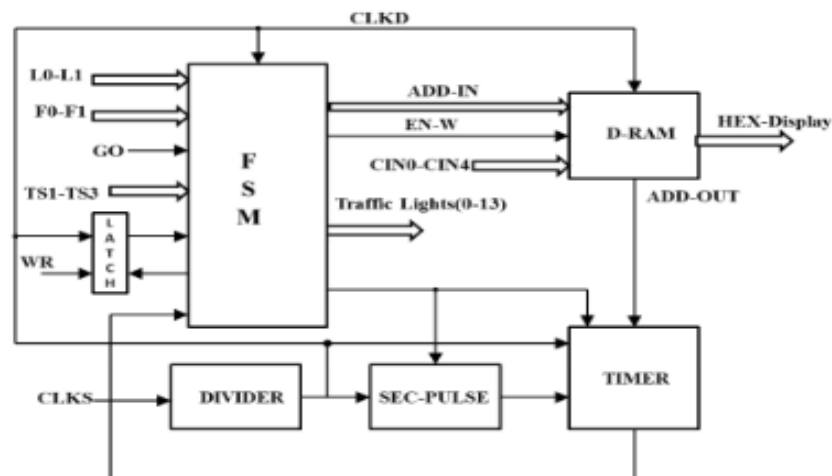
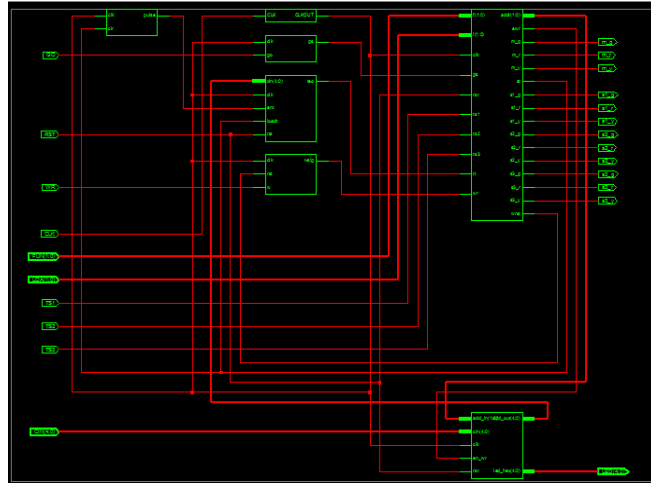


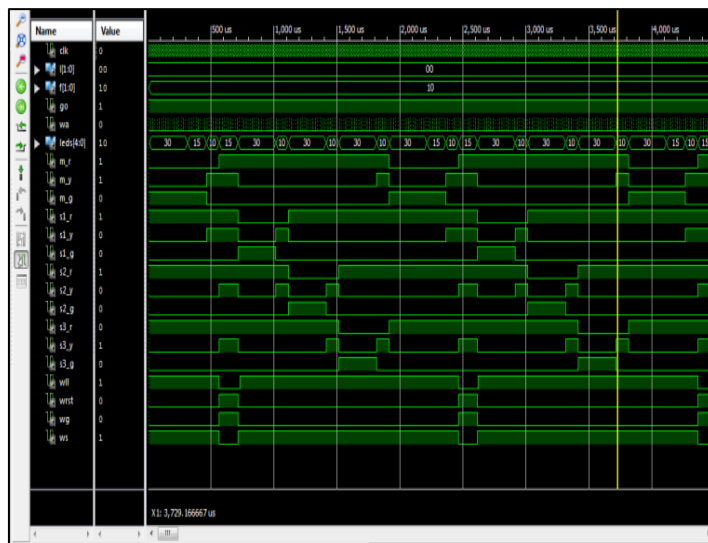
Fig. 3. The structure model of the TLC system on FPGA

The main benefit in using VHDL in the systems model is allowing description and verification of the structure of the required system before the design is translated by the synthesis tools into real hardware (gates and wires)[11]. Figure 4 describes the RTL and technology schematic diagram of the design system. Displaying schematic permits a technology level demonstration of HDL enhanced for specific device structure, that can be assisted to detect the scheme problems early in the model operation. All components of the system are simulated

using Xilinx ISE 14.7i. The state of normal mode is displayed in Figure 5, which represents the operation of the traffic light controller system that is discussed in section 3.



**Fig. 4.** RTL & Technology Schematic of the traffic light controller system



**Fig. 5.** Simulation result of the T.L.C system in normal mode

#### 4.1.2 Hardware Implementation and Results:

The traffic light controller system design is employed by synthesizing the VHDL structural code model. Then, using Xilinx ISE 14.7 tools, the bit file is generated and downloaded to the FPGA Spartan 3E development kit xc3s500efg320. The system's outputs are more than the LED on FPGA, then the LEDs display one state or use the supporting chip (expansion) external pin diligent (FX2 MIB). The real-time implementation of TLC and Chip-Scope implementation

is demonstrated in Figures 6 and 7 respectively. The system design provides the implementation of the hardware system besides the software. The hardware consumptions are listed in Table 3.



**Fig. 6.** FPGA implementation of TLC system

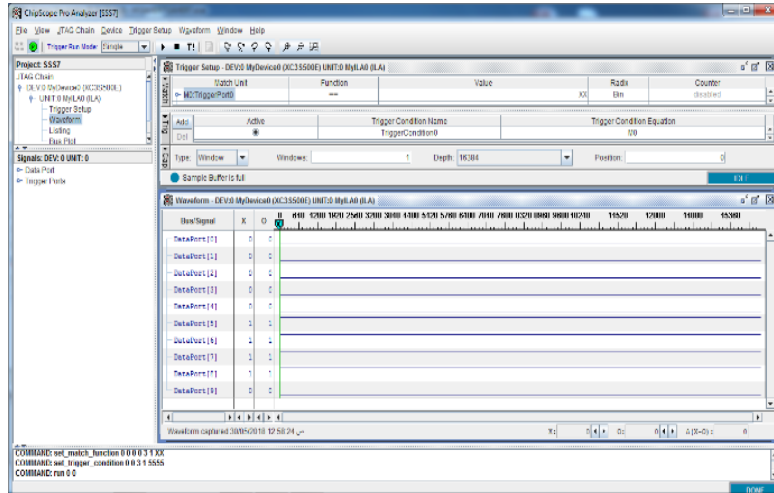
**Table 3:** The utilization of device and timing summary

Logic Utilization	Used	Available	Utilization
Number of GCLKs	2	24	8%
Number of bonded IOBs	40	232	17%
Number of Slice Flip Flops	115	9312	1%
Number of Slices	83	4656	1%
Number of 4 input LUTs:	141	9312	1%
Timing Summary			
Maximum output required time after the clock			4.532ns
Minimum input arrival time before the clock			5.355ns
Minimum period			6.132ns

#### 4.2 Traffic Light Controller Based on Arduino:

This sub-section briefly explains the features and characteristics of the Arduino Mega microcontroller which was used to realize the signal flow in the TLC system. Generally, Arduino is an open-source platform which is where the creator of a hardware /software allows the end-user access to their end product's design and realization. This means that the end-users have an opportunity to change or modify how the software works or looks and redistribute it. The Arduino Mega 2560 is a microcontroller board depending on the ATmega2560. A reset button, a power jack, a USB connection, and the other features which are summarized in Table 4 are attached on the board. It includes everything required to support the microcontroller; it is powered through an AC-to-DC adapter or battery or can be linked to a computer through a USB cable to get started. The Arduino mega 2560 top view is shown in figure 8. The Atmega2560 on the Arduino Mega comes pre-burned with a boot-loader that permits the user to upload new code to it without using an external hardware programmer. The programming language of Arduino looks like the Programs of C++ [20].





**Fig.7.** TLC system output using Chip-Scope

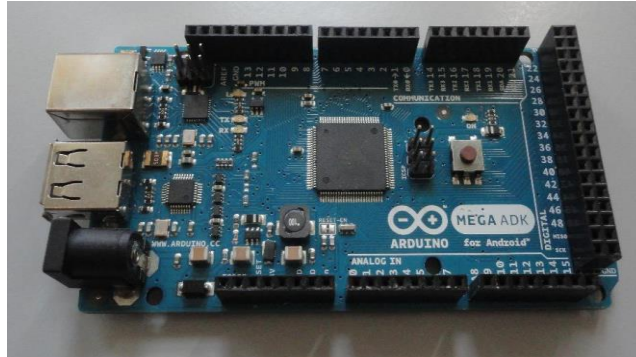
**Table 4.** The features of the Arduino Mega 2560

Feature	Type
Voltage of Operating	5V
Clock Speed	16 MHz
serial ports	4 UARTs
Input Voltage (limits)	6-20V
Input Voltage (recommended)	7-12V
Analog Input Pins	16
Digital I/O Pins	54 (of which 14 provide PWM output)
DC Current for 3.3V Pin	50 mA
DC Current per I/O Pin	40 mA
EEPROM	4 KB
SRAM	8 KB
Flash Memory	256 KB of which 8 KB is used by the bootloader

#### 4.2.1 Simulation of Traffic Light System:

The software Arduino contains two portions: Arduino IDE and Arduino Boot-Loader. Arduino Boot-Loader is a portion of code lying inside the microcontroller which makes the controller private and gives it the power of integration to the Arduino platform and the Arduino IDE. The Arduino IDE has a serial monitor, compiler, etc. Furthermore, Arduino is an open-source software and hardware platform where an Atmel microcontroller has a core hardware component and the C or C++ programming language is considered as the core software language and it is used to control displays, motors, LEDs, and any hobby projects.

The model is dependant on the simulation of a workable TLC system using an Arduino Mega Micro-controller as declared in the next subsection.



**Fig. 8.** The Arduino mega 2560 top view

#### **4.2.2 Hardware Implementation and Results:**

For the design, the display panel is made up of Light Emitting Diodes (LEDs). The display is used to indicate the status of the decoded counter and for performing the purpose for which it is meant to perform i.e. controlling the movement of traffic. The duration of the various states of the simulated traffic lights was done through the programming of the Arduino Mega using C++ programming language. The source codes for the LEDs and switches of the traffic lights are written depending on the states that are discussed in section 3.

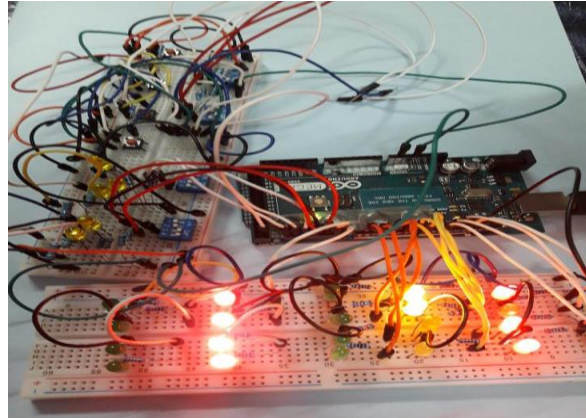
By programming the Arduino Mega platform which is used to display the LEDs at each function, the assembly was run by programming the Arduino board which is used to display the LEDs at each function. Fig. 9 shows the Arduino Mega hardware realization of the TLC system. The input system is through digital signals transmitted by Arduino IDE. This is executed by realizing the digital write operation and the signal's delay is in milliseconds since the Arduino Mega platform understands this. After the Arduino Mega TLC system is programmed and the assembling of the different elements, the following results were obtained:

- (1) 5452 bytes (2%) of program storage space is used. 253952 bytes is Maximum.
- (2) 436 bytes (5%) of dynamic memory is used for Global variables, while local variables use 7756 bytes.
- (3) At the indicated time, Spontaneous ON and OFF action of the traffic light.
- (4) Movement assignment to the heaviest way.

#### **5. Comparison between FPGA & Arduino in Design:**

Many parameters can be taken into account when making comparisons between the FPGA and Arduino, therefore only some parameters have been chosen which are considered very important when designing a traffic light system. These parameters are listed in Table 5.

From Table 5, it can be noted that the Arduino platform is best if the traffic light system needed is for a minor intersection and there is need for it to be simple, user-friendly, easily programmable, efficient and low budget. However, if the traffic light system is needed for the main intersection, it must be high security, highly reliable, highly efficient and have great speed, then the FPGA platform must be used to design this traffic light system.



**Fig.9.** Arduino Mega hardware implementation of TLC system

**Table 5.** The important parameters to compare between FPGA and Arduino

Parameters	FPGA	Arduino
cost	High	Low
Speed	High	Low
memory space	Large	Small
The way of design and processing. sequential and concurrent	sequential and concurrent	Sequential only
The complexity and flexibility of design	High	Low
The security for the design	High	Low
Power consumption	High	Low
Interact and manage I/O port	Low	High
Needed to supporting chip like memory	Needed to supporting chip	Standalone
Sensitivity and reliability	High	Low
Programming Source Language	VHDL	C/C++

## 6. Conclusion:

This paper presented the simulation and the implementation of traffic lights and their practical application to improve traffic conditions. This paper supported the production of a local traffic control system which is affordable, easy to maintain, and efficient in operation. This aim was achieved through the simulation of traffic lights using the FPGA Spartan 3E platform which was programmed by VHDL language. Besides, the simulation of traffic lights using the Arduino Mega board which was programmed with the C++ programming language, the comparison between the two designs was dependant on some parameters like cost, speed, memory space, and other parameters which are considered very important in designing traffic light systems. It was concluded that the Arduino mega was a simple prototype model, economical, user-friendly, easily programmable and worked as a standalone, while the FPGA

Spartan 3E platform has high security, high reliability, high efficiency and works at a great speed. Therefore the choice of between FPGA and Arduino depends on the nature of the need.

## 7. References:

1. A. Raza, A. Kumar, E. Chaudhary, "Traffic Light Controller Using VHDL", International Journal of Modern Trends in Engineering and Research (IJMTER), Vol. 04, Issue 4, PP. 57-62, April 2017.
2. T. Royani, J. Haddadnia, M. Alipoor, "Control of Traffic Light in Isolated Intersections Using Fuzzy Neural Network and Genetic Algorithm", International Journal of Computer and Electrical Engineering, Vol. 5, No. 1, PP. 142-146, Feb. 2013.
3. M. Dev, Prerna, Sh. Sachin, S. Sharma, U. Tyagi, "Smart Traffic Control System Using PLC and SCADA", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 1, Issue 2, PP. 169-172, Dec. 2012, Website: [www.ijirset.com](http://www.ijirset.com).
4. M. E. Safil, "Smart Traffic Light Controller Based on Microcontroller", Iraqi Journal of Computers, Communications, Control, and Systems Engineering (IJCCCE), Vol.16, No.1, PP. 38-45, 2016.
5. J. Alam, M. K. Pandey, "Advance Traffic Light System Based On Congestion Estimation Using Fuzzy Logic", International Journal of Emerging Technology and Advanced Engineering, Vol. 4, Issue 3, PP. 870-877, Mar. 2014, Website: [www.ijetae.com](http://www.ijetae.com).
6. A. Choudhary, "Adaptive Control of Traffic Grid Using Fuzzy Logic", International Journal of Electrical, Electronics, and Data Communication, Vol. 2, Issue 8, PP. 51-57, Aug. 2014.
7. L. Shanigarapu, K. V. Reddy, "Design and Implementation of Intelligent Traffic Light System", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 4, Issue 7, PP. 93 – 102, July 2015.
8. S. Nath, Ch. Pal, S. Sau, S. Mukherjee, A. Roy, A. Guchhait, D. Kandar, "Design of an FPGA Based Intelligence Traffic Light Controller with VHDL", International Conference on Radar, Communication and Computing (ICRCC), PP. 92-97, 21 - 22 Dec. 2012.
9. S. S. GAOPANDE, A. D. BHARADE, "Structural Modeling of Intelligent Traffic Light Controller on FPGA", International Journal of Electrical, Electronics and Data Communication, Vol. 2, Issue 9, PP. 63-66, Sep. 2014.
10. V. V. Dabahde, R. V. Kshirsagar, "FPGA-Based Intelligent Traffic Light Controller System Design", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2, Issue 4, PP. 1268-1271, April 2015, Website: [www.ijiset.com](http://www.ijiset.com)
11. A. K. Abdulrazzaq, "Design an Intelligent Traffic Light Controller", Thi-Qar University Journal for Engineering Sciences, Vol.7, No. 1, pp. 47-57, 2016.
12. P.G. Prasad, S.V. Krishna, "Design and Implementation of ITLC System Using FPGA", Journal of Electronics and Communication Engineering (IOSR-JECE), Vol. 12, Issue 5, Ver. II, PP. 07-13, Sep.- Oct. 2017, Website: [www.iosrjournals.org](http://www.iosrjournals.org).
13. Nour T. Gadawe, Sahar L. Qaddoori, "Design and implementation of smart traffic light controller using VHDL language", International Journal of Engineering & Technology, Vol. 8, Issue 4, PP. 596-602, 2019.
14. Ganiyu R. A., Arulogun O. T., Okediran O. O., "Development Of A Microcontroller-Based Traffic Light System For Road Intersection Control", International Journal Of Scientific & Technology Research, Volume 3, Issue 5, May 2014.
15. Prof.RUYawle, Kiran.K.Modak, Parmeshwar.S.Shivshette, Snehal.S.Vhaval," Smart Traffic Control System" in SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), Volume 3, Issue 3, March 2016.

16. S. Ghosh, Sh. Kumar, G. Ghosh, "Density Based Traffic Light System", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), Volume 6, Issue 6, PP. 590-593, June 2017.
17. Sh. Sahu, D. Paul, S.Senthilmurugan, "Density Based Traffic Signal Control Using Arduino And IR Sensors", International Journal of Novel Research and Development (IJNRD), Volume 3, Issue 4, PP. 73-76, April 2018.
18. D. Nath, V.V.Shete, Abhay Tambe, "Customization of Arm Processor in Mixed-Signal FPGA", International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 6, Aug. 2012.
19. Beyrouthy, L. Fesquet, "An Asynchronous FPGA Block with Its Tech-Mapping Algorithm Dedicated to Security Applications", International Journal of Reconfigurable Computing, Vol. 2013, 2013.
20. "Arduino Mega 2560 Datasheet", Website: [www.robotshop.com](http://www.robotshop.com)