IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
6-8 September 2007, Dortmund, Germany

FPGA-Based Advanced Real Traffic Light Controller System Design

WM El-Medany [1], MR Hussain [2]
1) CE Department, IT College, UOB, 32038 Bahrain wmelmedany@itc.uob.bh, http://people.man.ac.uk/~mbgedwme/
2) GEOMATEC, Spatial Information Research, Manama, Bahrain, marwan@geomatec.com.

Abstract – *Traffic light controller establishes a set of rules and instructions that drivers, pilots, train engineers, and ship captains rely on to avoid collisions and other hazards. Traffic control systems include signs, lights and other devices that communicate specific directions, warnings, or requirements. Traffic light controller (TLC) has been implemented using microcontroller, FPGA, and ASIC design. FPGA has many advantages over microcontroller, some of these advantages are; the speed, number of input/output ports and performance which are all very important in TLC design, at the same time ASIC design is more expensive than FPGA. Most of the TLCs implemented on FPGA are simple ones that have been implemented as examples of FSM. This paper concerned with an FPGA design implementation of a low cost 24-hour advanced traffic light controller system that was built as a term project of a VLSI design subject using VHDL. The implemented traffic light is one of the real and complex traffic lights in Kingdom of Bahrain, for four roads and motorway with sensors and camera. The system has been successfully tested and implemented in hardware using Xilinx Spartan 3 FPGA. The system has many advantages over the exciting TLC.*

Keywords - *FPGA, Xilinx, VHDL, VLSI.*

I. INTRODUCTION

Field programmable gate arrays (FPGAs) are extensively used in rapid prototyping and verification of a conceptual design and also used in electronic systems when the mask-production of a custom IC becomes prohibitively expensive due to the small quantity [1]. Many system designs that used to be built in custom silicon VLSI [2] are now implemented in Field Programmable Gate Arrays. This is because of the high cost of building a mask production of a custom VLSI especially for small quantity [3].

In this paper the main objective was to design a 24-houre traffic light controller to manage the traffic movement of four roads at the same time, and achieve maximum utilization for the four roads. Optimal traffic light control is a multi-agent decision problem, our design learns the expected waiting times of cars for red and green lights at each intersection [4, 5, and 6]. In the rush hours, when people going to work or coming back to home the traffic lights of all roads are controlled with fixed time. However, in the normal time, the main roads are controlled with a fixed time while the narrow roads are controlled autonomously by sensors [6, 7, and 8].

The material of this article is arranged as follows: in section II, we describe the structure of the four roads that has been used as an example for the design and the time allocated for each traffic light. Description of the hardware design and VHDL model is the subject of section III. In section IV we explain the state diagram of the design. The simulation of the design and FPGA implementation is studied in section V. Finally we discuss our results in section VI.

II. STRUCTURE OF THE FOUR ROADS THAT ARE GOING TO BE CONTROLLED

Fig. 1 shows the structure of the four roads that has been used as a practical example to design our controller which are located in Manama city, ***Kingdom of Bahrain***. In this structure we have six traffics, T1, T2, T3, T4, T5, and T6. The main roads are T1 and T3. There is also one camera which is placed only for traffic T1. The timing for the traffic's is working as follows:

- T1 and T3 are green for 12 seconds.
- Other traffics are green for 6 seconds.
- The green to yellow sign in each traffic light has 2 seconds delay.
- There is 2 second safety during transition from one traffic light to another (or from one road to another).
- The red to red yellow sign in each traffic light has 1 second delay.

Assuming that the roads are crowded (all sensors equal to logic '1'); the normal scenario of the traffic is as follows:

- The system starts with T1 and T2 having the green sign.
- T1 and T2 will stay green, after six seconds T2 will turn to red and T3 will be green.
- After the remaining time for T1 finishes, T1 will turn to red and T4 will be green.
- After the time for T3 and T4 finishes, both return to red. Then it is the turn for T5 to be green for six seconds.
- Finally T6 will become green after T5 finishes.

Table 1 show a timing states for the six traffic lights for the green and red lights only.

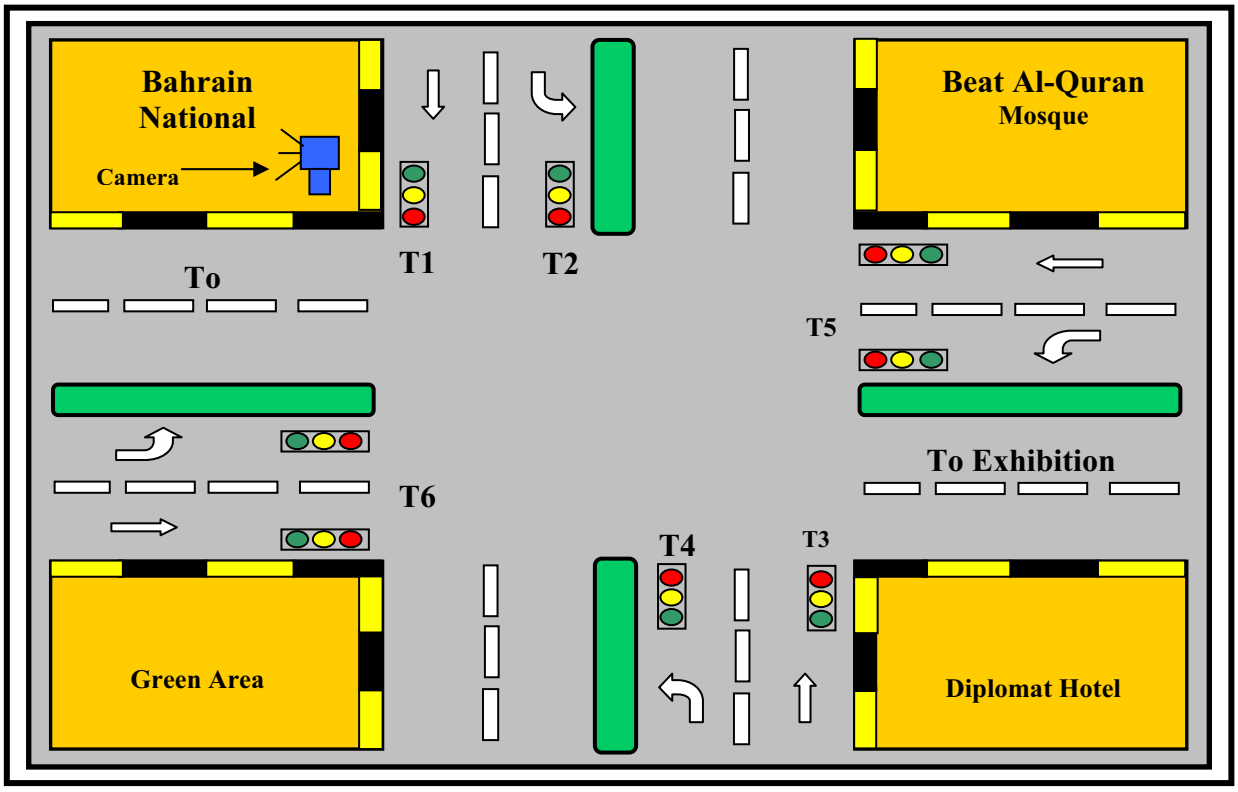

Fig. 1. The Four Roads Structure.

What if some of the roads are empty (their sensors are logic '0'):

There are some procedures to be done before turning from one traffic light to another, and they are as follows:

- If the timer of the current traffic finishes and still there are cars on its road , the traffic will not get red unless it makes sure that there is another traffic with sensor = logic '1'.
- If there is no other traffic having cars, so the current traffic will stay green unless one of the two conditions appears:
  1. A car came in other traffic light (sensor will be logic '1'). Here the current traffic will changed to red and the other traffic light will be green.
  2. If the sensor of the current traffic become logic '0' (means the cars in its road finished) and still other sensors are equal to logic '0';

Here the current traffic will be turned to *red* and T1 and T3 will be *green* as we assumed they are the two main traffics (main roads).

These steps are the conditions that has to be checked before switching between any two traffics for the whole times (24-hours).

Important Notes:

- When T1 is Green T4, T5 and T6 should be *Red* (T2 or T3 can be *Green* at the same time).
- When T2 is Green T4, T3, T5 and T6 should be *Red* (T1 is *Green* too).
- When T3 is Green T2, T5 and T6 should be *Red* (T1 or T4 can be *Green* at the same time).
- When T4 is Green T1, T2, T5 and T6 should be *Red* (T3 is *Green* too).
- When T5 is *Green* all other traffics should be *Red.*
- When T6 is *Green* all other traffics should be *Red.*

TABLE 1
TRAFFIC LIGHT TIMING STATES

Time	T1	T2	T3	T4	T5	T6
1st 6 sec	Grn	Grn	Rd	Rd	Rd	Rd
2nd 6 sec	Grn	Rd	Grn	Rd	Rd	Rd
3rd 6 sec	Rd	Rd	Grn	Grn	Rd	Rd
4th 6 sec	Rd	Rd	Rd	Rd	Grn	Rd
5th 6 sec	Rd	Rd	Rd	Rd	Rd	Grn

III. HARDWAR DESIGN AND VHDL MODEL

Traffic lights can sense and respond to traffic because of a wire loop embedded in the road. Electric current, run through the loop, creates a magnetic field. When a car

bumper interferes with this field, a signal is sent to a roadside traffic light controller.

The design of our traffic light system went through three stages. The first stage was the implementation of the state diagram. The second stage is writing and simulating the VHDL code [9], which will be the rules to control the traffic signs. The last stage is programming the FPGA and development of the interface circuit. The system has been successfully implemented, tested and compared to the existing traffic lights in Kingdom of Bahrain.

Fig. 2. shows the VHDL model of the controller, where:

- **Clock:** Is the system clock.
- **Reset:** Is the system reset.
- **S1, S2, S3, S4, S5, S6 and Sc:** All are Sensors used to sense if there are any car exists in any road. Sc is used to sense any car break the Red sign in the main road.
- **T1, T2, T3, T4, T5** and **T6**: All represents the six traffic light that the system is going to control them. For example **T1 <2:0>** represents the (red, yellow and green) sign, where each colure represents one bit (**T1 <2>** → Green, **T1 <1>** → Yellow and **T1 <0>** → Red).
- **Seven_Seg:** Represents the Seven Segment Display for the traffic timing.
- **AN:** Represents the four anodes of the Seven Segment Display.
- **Camera:** This will be on when Sc and the main road red sign is on (logic '1').

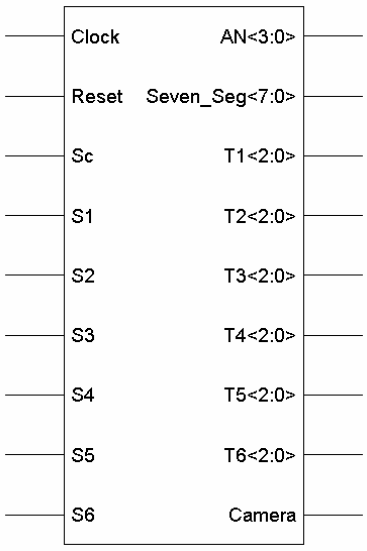

Fig. 2. VHDL Model of the Controller.

IV. State Diagram

Fig. 3. shows the state diagram of the controller which includes 65 states. The transition from state 0 to state 19 depends only on the time delay for each traffic state, which represents the case when all sensors are active; this means that there are cars in all roads. States (s0, s4, s8, s12, and s16) are the main states in which the traffic is either green or red. There are three intermediate states between each two of the main states, these intermediate states represents the Yellow of the active traffic (Green), all red which is the safety state, and Red-Yellow for the next active traffic. When all sensors are active the transition of the main states will follow the sequence of the green light as shown in the state table of figure (2). The sequence of the green light will be (T1&T2, T1&T3, T3&T4, T5, T6, and continue T1&T2 …). That is why we have five states (s0, s4, s8, s12, and s16). The rest of the 65 states used for the transitions in case of some sensors are active and some are not. These states are numbered ($S_{N1\text{-}N2\text{-}N3}$) where N1 refer to the present state, N2 refer to the next state, and N3 refer to light output as mentioned in section III.

V. Simulation and FPGA Implementation

Fig. 4. Shows the simulation results for the controller with all sensors are active, in this case it is clear from the simulation that the transition between the main states depends only on the timer of each state, and the transition will be (st0, st1, …, st19), and then starts again from st0. In Fig. 5., only sensor S5 and S6 are active in state st0, in that case the priority is for traffic T5, then T6 which are correspond to st12 and st16 respectively.

The simulation results in Fig. 4. and Fig. 5. follow the same sequence of the state diagram in Fig. 3., either from the transition point of view or the output of each state point of view.

Fig. 6. shows the RTL schematic of the controller, the total number of logic gates in the design is 62 K. The design has been tested on Spartan 2E FPGA, then has been modified and tested on Spartan 3 FPGA starter kit.

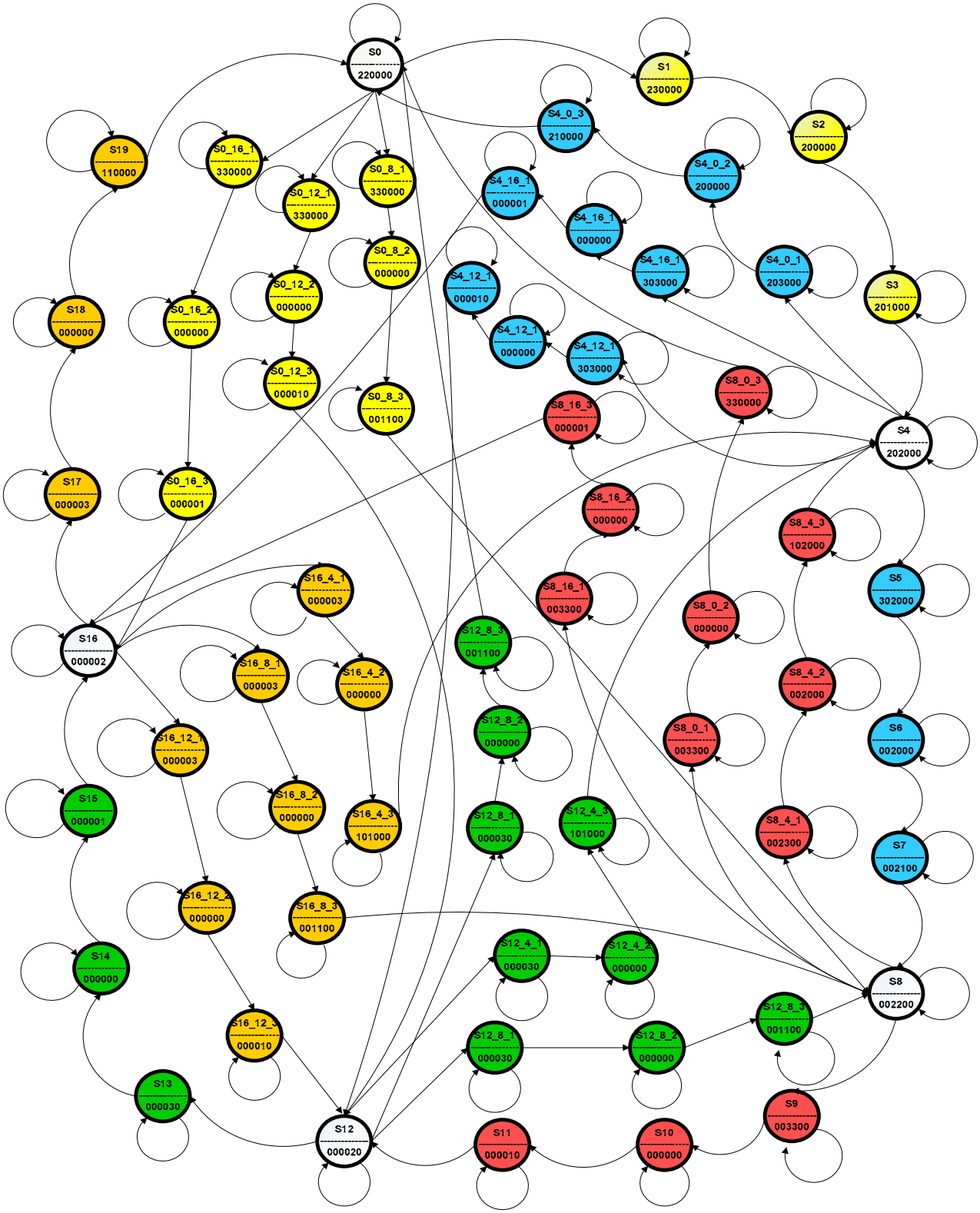

Fig. 3. State Diagram of the Controller.

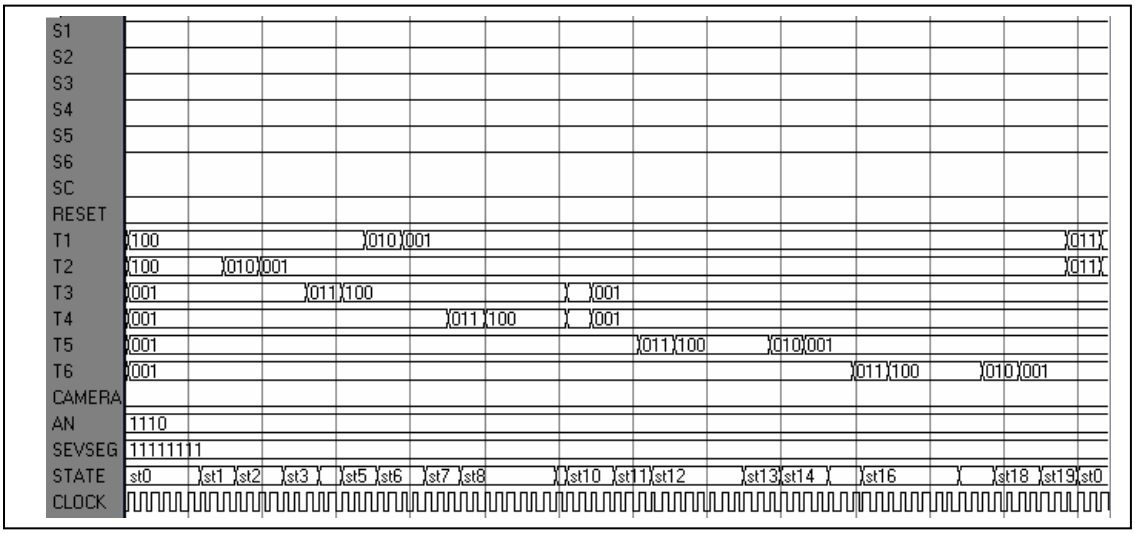

Fig. 4. Simulation Results for the Controller With all sensors are active.

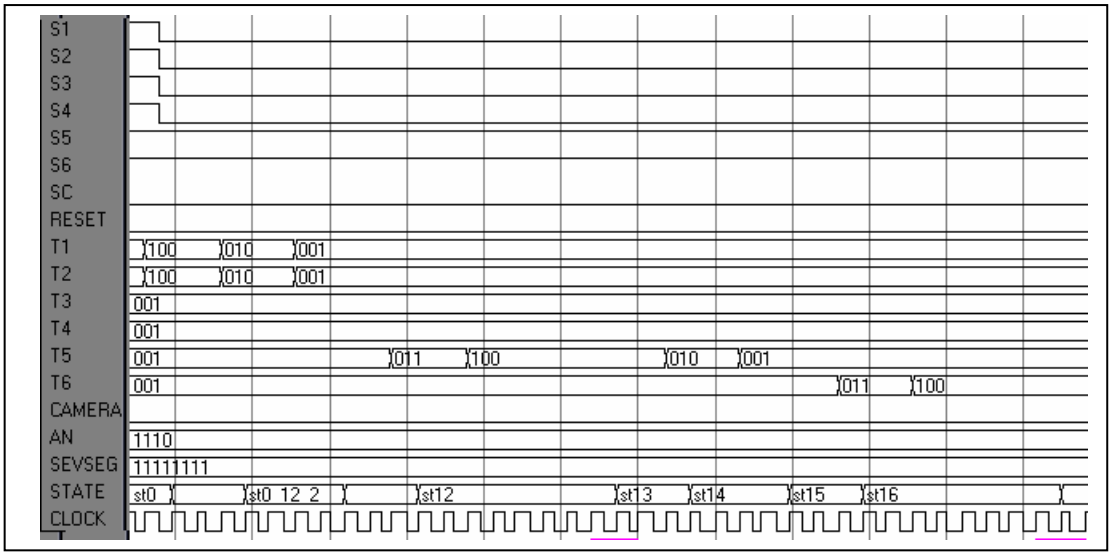

Fig. 5. Simulation Results for the Controller With sensors S5 & S6 only are active

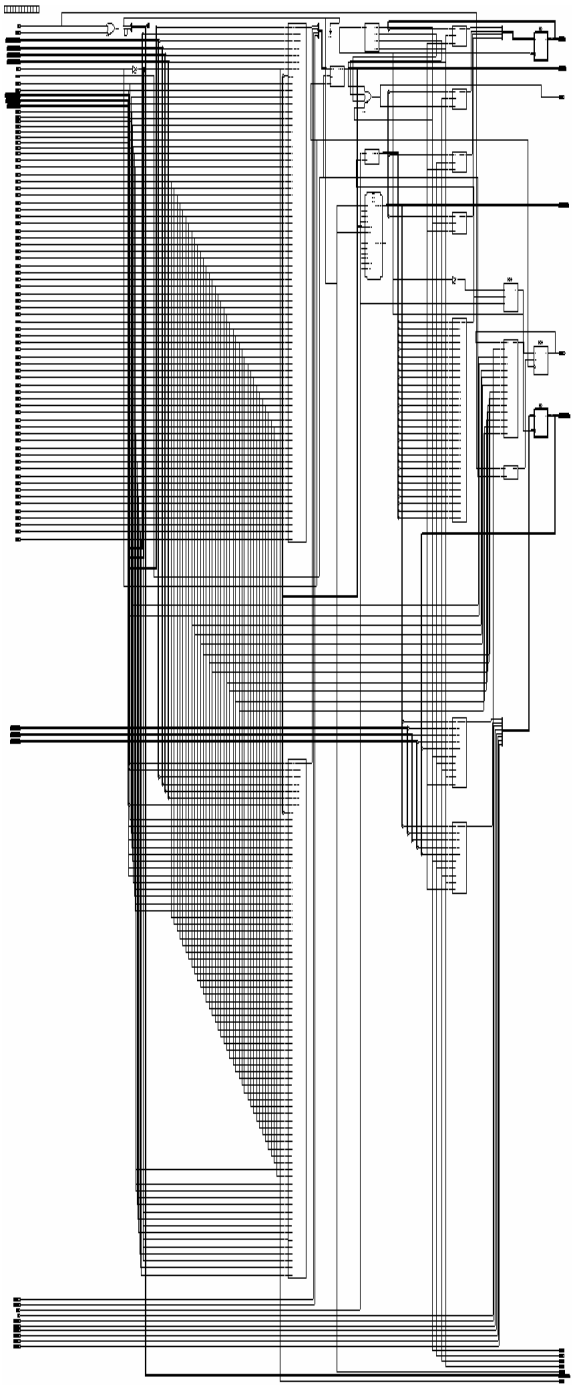

Fig. 6. RTL Schematic for the Controller.

VI. Conclusion

An FPGA design of A 24-hour traffic light controller system of a four roads structure with six traffic lights has been simulated, implemented and tested. The system has been designed using VHDL, and implemented on hardware using Xilinx Spartan 3 FPGA Starter kit. Our design reaches the maximum utilization of the traffic either during rush hours or normal time. More functions could be added to the design. Some of these functions are to control more than six traffic lights. Also, to allow the user to assign the time for each traffic light (i.e. minimum time to be Green), adding more sensors on each road to count the number of cars in each road and check for the longer queue to increase the timer for that road, another function is to link the traffic light with the other traffic lights along the streets to increase the flow of traffic.

Acknowledgment

We would like to thank the project team work Mr. Ahmed Abu-Aesh, Mr Mohammed Ahmed Noor, and Mr. Hamza Ahmed who made this work possible.

References

[1] Wayne Wolf, *FPGA-Based System Design*, Prentice Hall, 2005.
[2] Jan M. Rabaey, *Digital Integrated Circuits, A Design Perspective*, Second Ed., Prentice Hall, 2003.
[3] *Design of a VLSI Integrated Circuit*, IEEE, Piscataway, USA.
[4] Taehee Han; Chiho Lin, "Design of an intelligence traffic light controller (ITLC) with VHDL," *Proceedings 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering (TENCON '02)*, 28-31 Oct. 2002, vol 3, pp:1749 - 1752.
[5] M. Vreeken, J. van Veenen, J. A. Koopman, "Simulation and optimization of traffic in a city," *IEEE WieringIntelligent Vehicles Symposium*, 14-17 June 2004, pp. 453 – 458.
[6] Malik J. Ojha, "Design of a VLSI FPGA integrated circuit," Dept. of Eng., Denver Univ., CO, USA, Technical, Professional and Student Development Workshop, 2005 IEEE Region 5 and IEEE Denver Section, 7-8 April 2005.
[7] Yi-Sheng Huang, "Design of traffic light control systems using statecharts," *The Computer Journal*, Nov 2006, vol. 49, pp. 634 – 649.
[8] Li Lin; Tang N an; Mu Xiangvang; Shi Fubing, "Implementation of traffic lights control based on Petri nets," Proceedings of IEEE Intelligent Transportation Systems Symposium, 12-15 Oct 2003, vol. 2.
[9] Mark Zwolinski, *Digital System Design With VHDL*, Second Ed., Prentice Hall, 2004.