

Advanced Fuzzy Logic Model for Volume Based Traffic Signal Control

Bandan Kumar Bhoi^{1,*}, Abhipsa Panda², Sarthak Acharya²

¹Assistant Professor in Electronics and Telecommunication, VSSUT, Burla, Sambalpur, Odisha, India

²Student in Electronics and Telecommunication, VSSUT, Burla Sambalpur, Odisha, India

*Corresponding author: bandanuce@gmail.com

Received November 30, 2013; Revised December 20, 2013; Accepted December 26, 2013

Abstract This paper describes a functional Fuzzy Traffic Controller (FTC), which utilizes fuzzy logic algorithm to achieve a smart and flexible knowledge based system in hardware design while achieving better efficiency in the traffic control and minimizing traffic jam occurrences at interchange on road area. The designing part of FTC chip into verilog program eliminates the short comings of other custom facilities and conventional controller design available today. The Finite State Machine (FSM) of the FTC is coded in verilog for controlling the specific traffic flow application. The main features of FTC chip consist of various operational modes, safety features and capability through fuzzy logic algorithm.

Keywords: Fuzzy Traffic Controller, Finite State Machine, fuzzy logic algorithm

Cite This Article: Bandan Kumar Bhoi, Abhipsa Panda, and Sarthak Acharya, "Advanced Fuzzy Logic Model for Volume Based Traffic Signal Control." *Journal of Embedded Systems* 1, no. 1 (2013): 1-6. doi: 10.12691/jes-1-1-1.

1. Introduction

Traditional traffic controllers are bottleneck and inefficient in handling dynamic traffic load and this is evident in virtually all the cities across the globe. In a conventional traffic light controller, the traffic lights change at constant cycle time, which is clearly not the optimal solution. It would be more feasible and sensible to pass more cars at the green interval if there are fewer cars waiting behind the red lights or vice versa. The Fuzzy logic theory is introduced in the traffic controller to provide an intelligent green interval response based on dynamic traffic load inputs. This discards the use of preset cycle time. This traffic control system is designed for a 6-way junction.

In the IEEE paper of 2006 named "Design and Implementation of an Intelligent Fuzzy VLSI Chip for Traffic Control Application", numbered 1-4244-0549-1, a traffic controller was designed for a 4-way junction using Fuzzy logic. The Fuzzy logic theory was used and a Fuzzy logic control scheme was proposed to overcome the inefficiency of conventional traffic controllers that had a preset cycle time regardless of the dynamic traffic load. In this paper the Fuzzy logic algorithm was developed using VHDL code.

The present work is an improvement over the above paper. Our work is about the design of a Fuzzy Traffic Controller for a 6-way junction. This FTC also operates based on dynamic traffic load inputs. The code for this FTC is designed using verilog which is easy to understand. This work also has an additional feature of detecting any

vehicle breaking the signal rules. The total delay due to this FTC is also found to be less as compared to the reference. Hence it would further reduce the percentage of stopped vehicles by adjusting its green cycle time.

2. Architecture

Consider a 6-way junction for which the FTC has to be designed. The FTC on a single 6-way intersection can be used as local controller that is responsible for regulating the traffic on its own junction.

The possible 8 traffic flow paths for a 6-way FTC are considered as follows:

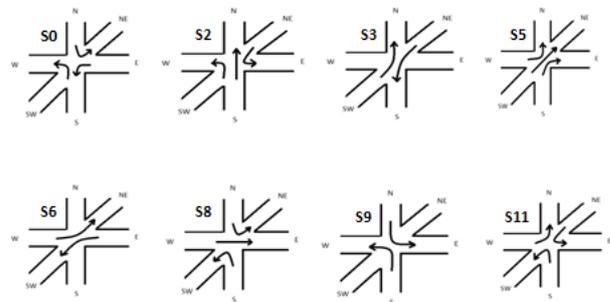


Figure 1. Path Diagram

The various directions considered in the path diagram (Figure 1) are north, south, east, west, north-east and south-west. The arrows shown in Figure 1 indicate traffic flow in the direction to which they are pointing as well as in the reverse direction. The places where arrows are not

indicated mean that there is no traffic flow in that direction.

The 8 paths are represented by 8 states i.e, s0, s2, s3, s5, s6, s8, s9, s11.

The various states can be explained as follows:

S0:

In the state s0 the traffic is allowed to flow from north to north-east and north-east to north direction, east to south and south to east direction, south-west to west and west to south-west direction simultaneously.

S1:

In the state s1 the traffic is allowed to flow from south to north and north to south direction, south-west to west and west to south-west direction, north-east to east and east to north-east direction simultaneously.

S3:

In the state s3 the traffic is allowed to flow from south-west to north and north to south-west direction, north-east to south and south to north-east direction simultaneously.

S5:

In the state s5 the traffic is allowed to flow from west to north and north to west direction, south-west to north-east and north-east to south-west direction, south to east and east to south direction simultaneously.

S6:

In the state s6 the traffic is allowed to flow from west to north-east and north-east to west direction, east to south-west and south-west to east direction simultaneously.

S8:

In the state s8 the traffic is allowed to flow from north to north-east and north-east to north direction, west to east and east to west direction, south to south-west and south-west to south direction simultaneously.

S9:

In the state s9 the traffic is allowed to flow from north to east and east to north direction, south to west and west to south direction simultaneously

S11:

In the state s11 the traffic is allowed to flow from west to north and north to west direction, north-east to east and east to north-east direction, south to south-west and south-west to south direction simultaneously.

All the above states can be obtained if the required volume conditions are satisfied i.e, traffic is allowed to flow in that direction in which the volume of vehicles is more. In all above states we have also considered reverse direction of the arrows as shown in Figure 1 because we are considering a 2-way path which allows going on one side along with coming on the other side.

Irrespective of the volume difference given as input, a particular state can be obtained if there is an emergency but this would require an additional input called emergency input.

3. Development of FTC Algorithm

The FTC has got 15 inputs i.e, the volume difference in each possible direction. They are represented by vector 'v'. It has 15 outputs representing green signal in every possible direction, represented by vector 'g'. It has 3 outputs each for yellow and red signals, represented by vectors 'y' and 'r'.

It has also got 8 emergency inputs in order to facilitate police or ambulance, represented by vector 'e'. It has got

an output represented by variable 'z' to detect tress passing.

The different vectors are related to different directions in the following way:

Table 1.

Volume difference	
Vector	Direction
V[0]	V [NS]
V[1]	V [EW]
V[2]	V [NEWS]
V[3]	V [NEE]
V[4]	V [SWW]
V[5]	V [ES]
V[6]	V [WN]
V[7]	V [NNE]
V[8]	V [SSW]
V[9]	V [NE]
V[10]	V [SW]
V[11]	V [NES]
V[12]	V [SWN]
V[13]	V [ESW]
V[14]	V [WNE]
Green Signal	
Vector	Direction
g[0]	g [NS]
g[1]	g [EW]
g[2]	g [NEWS]
g[3]	g [NEE]
g[4]	g [SWW]
g[5]	g [ES]
g[6]	g [WN]
g[7]	g [NNE]
g[8]	g [SSW]
g[9]	g [NE]
g[10]	g [SW]
g[11]	g [NES]
g[12]	g [SWN]
g[13]	g [ESW]
g[14]	g [WNE]
Yellow Signal	
Vector	Direction
y[0]	y [NS]
y[1]	y [EW]
y[2]	y [NEWS]
Red Signal	
Vector	Direction
r[0]	r [NS]
r[1]	r [EW]
r[2]	r [NEWS]

Here v [NS] represents the volume difference in between north and south directions, v [EW] represents the volume difference in between east and west directions, v [NEWS] represents the volume difference in between north-east and west-south directions and so on. In a similar way the 15 green outputs correspond to different directions. The red and yellow signals are given for north and south, east and west, north-east and west-south directions.

4. Implementation of FTC

The required volume differences act as input to the FTC. The volumes at various junctions are obtained by sensor outputs.

In-roadway or over-roadway sensors operating in real time can be used in this case in order to regulate traffic flow. A sensor is embeded in the pavement of the paths if it is an in-roadway sensor. An in-roadway sensor consists of an inductive loop that gets disturbed due to vehicals passing over it thus resulting in detection. A timer can also be used as a default during times of very low traffic

density and as a backup in case of sensor failure. Examples of in-roadway sensors include inductive loop detectors, magnetometers, tape-switches etc.

In case of over-roadway sensors electromagnetic waves are used to detect the presense of vehicles. Examples of over-roadway sensors include video-image processors, microwave radar, ultrasonic and passive infrared sensors, laser radar sensors. The sensors detect the volume of traffic flow through every path. This detected volume is given as sensor output.

As there are 6 paths, six sensors are required. The different sensor outputs are :

- V [N] - volume in north direction
- V [S] - volume in south direction
- V [E] - volume in east direction
- V [W] - volume in west direction
- V [NE] - volume in north-east direction
- V [SW] - volume in south-west direction

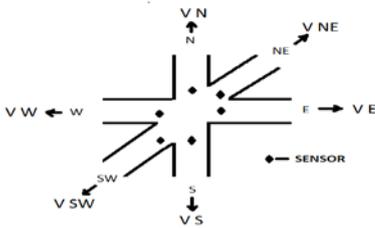


Figure 2. Sensors plotted in the paths

The difference of the sensor outputs is calculated as per Table 2 and then fed to the FTC. The '~' operator gives difference between two operands (volumes). Then the highest volume is decided based on fuzzy logic.

Table 2.

Volume difference	
Sensor output	Difference
V[N]~V[S]	V [NS]
V[E]~V[W]	V [EW]
V[NE]~V[WS]	V [NEWS]
V[NE]~V[E]	V [NEE]
V[SW]~V[W]	V [SWW]
V[E]~V[S]	V [ES]
V[W]~V[N]	V [WN]
V[N]~V[NE]	V [NNE]
V[S]~V[SW]	V [SSW]
V[N]~V[E]	V [NE]
V[S]~V[W]	V [SW]
V[NE]~V[S]	V [NES]
V[SW]~V[N]	V [SWN]
V[E]~V[SW]	V [ESW]
V[W]~V[NE]	V [WNE]

The FTC checks all the conditions and switches to the required state based on given volume difference as input. The output is observed as green, yellow or red signals. The green signal indicates 'GO', the red signal indicates 'STOP' and the yellow signal indicates 'WAIT'.

The state diagram for 6-way FTC can be represented as:

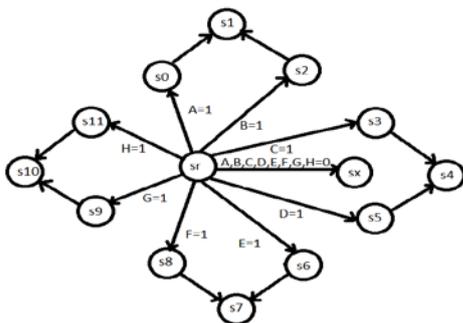


Figure 3. State diagram

Mathematically, the conditions for various states can be written as:

$$A = s0 = (v[4] * V[5] * V[7]) + e[0] \quad (1)$$

$$= (V[SWW]*V[ES]*V[NNE])+e[0]$$

$$B = s2 = (V[0] * V[3] * V[4]) + e[1] \quad (2)$$

$$= (V[NS] * V[NEE] * V[SWW]) + e[1]$$

$$C = s3 = (V[11] * V[12]) + e[2] \quad (3)$$

$$= (V[NES]*V[SWN])+e[2]$$

$$D = s5 = (V[2] * V[5] * V[6]) + e[3] \quad (4)$$

$$= (V[NEWS]*V[ES]*V[WN])+e[3]$$

$$E = s6 = (V[13] * V[14]) + e[4] \quad (5)$$

$$= (V[ESW]*V[WNE])+e[4]$$

$$F = s8 = (V[1] * V[7] * V[8]) + e[5] \quad (6)$$

$$= (V[NS])*V[NNE]*V[SWW])+e[5]$$

$$G = s9 = (V[9] * V[10]) + e[6] \quad (7)$$

$$= (V[NE]*V[SW])+e[6]$$

$$H = s11 = (V[3]*V[6] * V[8]) + e[7] \quad (8)$$

$$= (V[NEE]*V[WN]*V[SSW])+e[7]$$

In case the conditions are not satisfied, it considers a tress passing of the signal. This indication is used to initiate a camera in order to track the faulty person. Even in this case we use over-roadway sensors for detection which include sensors mounted on tall poles adjacent to roadway or traffic signal mast arms over the roadway.

In case of police emergency or ambulance requirements, directly the required state can be switched to, using emergency inputs. The change of traffic flow states takes place through several intermediate states which serve for red to green or green to red signal conversion.

Here s0, s2, s3, s5, s6, s8, s9, s11, represent the states of traffic flow. The states s1, s4, s7, s10 represent the intermediate states.

The state sr represents the reset state and the state sx reresents the state to detect tress passing.

The arrows in Figure 3 indicate that there is a state change in that direction when the given condition(above the arrow) is satisfied. If there are no conditions given then the state change takes place automatically placing the FTC in intermediate state.

Initially the FTC is in reset state i.e, sr. From state sr it changes to required state, allowing traffic flow, via the intermediate states. When FTC is in sr state, all the signals are in low condition or off state.

If we want to get the state s0 then we have to give A=1 which implies that volume vectors V[SWW], V[ES], V[NNE] are to be made high. These vectors are represented by v[4], V[5], V[7] in the verilog code for FTC. The state s0 is also achieved by making emergency input e[0] = 1(high). Once the state s0 is obtained the FTC will automatically reach the intermediate state s1 which

can be verified when outputs $r[1]$, $r[2]$ and $y[0]$ glow (high).

If we want to obtain state s_2 then we have to make $B=1$ which implies that volume vectors $V[0]$, $V[3]$, $V[4]$ are made high. The state s_2 can also be achieved by making emergency input $e[1]$ high. The FTC switches to intermediate state s_1 after state s_2 .

When the state s_3 is to be obtained then we have to make $C=1$ which implies that volume vectors $V[11]$, $V[12]$ are made high. The state s_3 is also obtained by making emergency input $e[2]$ high.

When the state s_5 is to be obtained then we have to make $D=1$ which implies that volume vectors $V[2]$, $V[5]$, $V[6]$ are made high. The state s_5 is also obtained by making emergency input $e[3]$ high. Both the states s_3 and s_5 have same intermediate state s_4 which can be verified when outputs $r[0]$, $r[1]$, and $y[2]$ glow.

When the state s_6 is to be obtained then we have to make $E=1$ which implies that volume vectors $V[13]$, $V[14]$ are made high. The state s_6 is also obtained by making emergency input $e[4]$ high.

When the state s_8 is to be obtained then we have to make $F=1$ which implies that volume vectors $V[1]$, $V[7]$, $V[8]$ are made high. The state s_8 is also obtained by making emergency input $e[5]$ high. Both the states s_6 and s_8 have same intermediate state s_7 which can be verified when outputs $r[0]$, $r[2]$, and $y[1]$ glow.

When the state s_9 is to be obtained then we have to make $G=1$ which implies that volume vectors $V[9]$, $V[10]$ are made high. The state s_9 is also obtained by making emergency input $e[6]$ high.

When the state s_{11} is to be obtained then we have to make $H=1$ which implies that volume vectors $V[3]$, $V[6]$, $V[8]$ are made high. The state s_{11} is also obtained by making emergency input $e[7]$ high. Both the states s_9 and s_{11} have same intermediate state s_{10} which can be verified when outputs $r[0]$, $r[1]$, and $r[2]$ glow.

The various volume conditions to reach a particular state are given as follows:

Table 3.

States	Conditions	
S0	$V[4]$, $V[5]$, $V[7]$	$e[0]$
S2	$V[0]$, $V[3]$, $V[4]$	$e[1]$
S3	$V[11]$, $V[12]$	$e[2]$
S5	$V[2]$, $V[5]$, $V[6]$	$e[3]$
S6	$V[13]$, $V[14]$	$e[4]$
S8	$V[1]$, $V[7]$, $V[8]$	$e[5]$
S9	$V[9]$, $V[10]$	$e[6]$
S11	$V[3]$, $V[6]$, $V[8]$	$e[7]$

5. Results and Discussions

This verilog simulation program has got 14 states of which 8 states represent the possible conditions of traffic flow in a 6-way junction. Other 5 states act as intermediate states, of which one is a reset state. There is another state to detect any vehicle crossing the signal or breaking the rules. The emergency inputs correspond to the 8 states representing possible traffic flow. These are used to directly switch to required state without the involvement of intermediate states in case of emergency. The green signal becomes high only when the required volume difference condition is satisfied and thus allows traffic flow in that direction. Similarly the rise of red signal opposes the traffic flow in given direction. The yellow signal is used as intermediate signal when green signal is switched to red or red signal is switched to green. This is because a direct change from green to red or red to green cannot be made. When none of the conditions are satisfied, it is concluded that traffic rules are violated. This results when output 'z' goes high. At this instant a camera can be initiated to capture the image of faulty vehicle.

6. Verilog Simulation Results

The various project properties which were required for the implementation of the verilog code are as follows:

Table 4.

Property Name	Value
Product category	All
Family	Virtex 4
Device	XC4VLX15
Package	SF363
Speed	-12
Top-level sourcetype	HDL
Synthesis Tool	XST(VHDL/Verilog)
Simulator	ISE Simulator(VHDL/Verilog)
Preffered language	Verilog

7. Comparison of Results

As compared to the reference paper titled "Design and Implementation of an Intelligent Fuzzy VLSI Chip for Traffic Control Application" the following results were obtained:

Table 5.

Parameters	Reference	Our work
Selected Device	4vlx15sf363-12	4vlx15sf363-12
Number of Slices	10 out of 6144 0%	40 out of 6144 0%
Number of Slice Flip Flops	12 out of 12288 0%	14 out of 12288 0%
Number of 4 input LUTs	16 out of 12288 0%	70 out of 12288 0%
Number of IOs	20	47
Number of bonded IOBs	16 out of 240 6%	47 out of 240 19%
IOB Flip Flops		22
Number of GCLKs	1 out of 32 3%	1 out of 32 3%
Delay	5.463ns (4.366ns logic, 1.097ns route) (79.9% logic, 20.1% route)	3.879ns (3.613ns logic, 0.266ns route) (93.1% logic, 6.9% route)

Both the programs were simulated using Virtex 4 family.

The RTL schematic representation of the FTC block is obtained as:

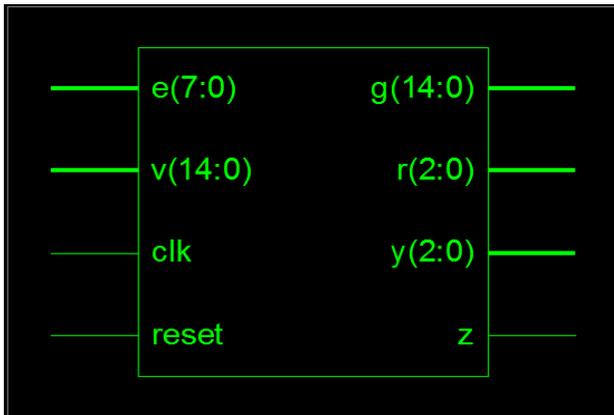


Figure 4. RTL Schematic Representation

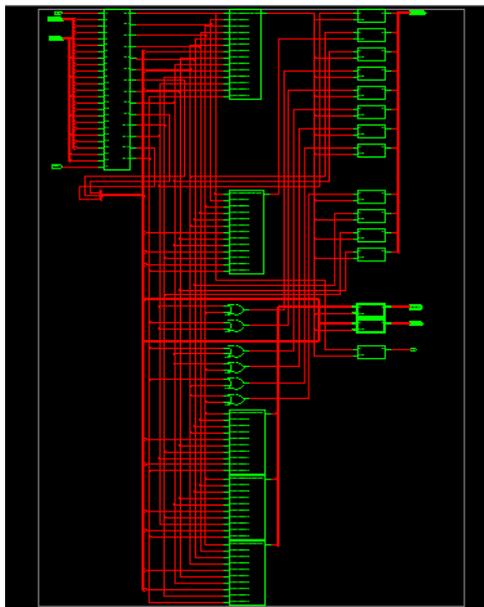


Figure 5. RTL Schematic Representation

From the RTL schematic we can find out there are 4 types of inputs given to the FTC which include the clock, reset, 15-bit volume input vector, 8-bit emergency input vector and the various outputs include 15-bit green signal vector, 3-bit red signal vector, 3-bit yellow signal vector and signal z to detect tress-passing.

The verilog simulation results for 6-way traffic controller obtained using XILINX software are as follows:
Testbench for e[6] = 1:

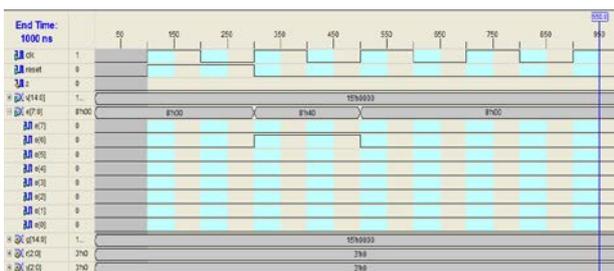


Figure 6.

Output for e[6]=1

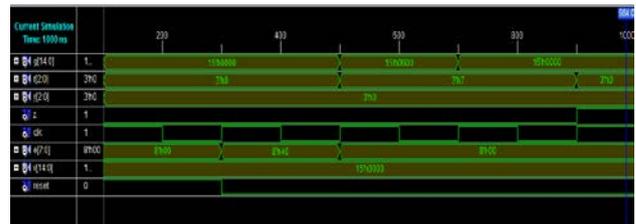


Figure 7.

When the emergency input e[6] is made high, the state s9 is obtained which can be verified from the above simulated result. In this state the green signals g[9] and g[10] become high which implies that traffic is allowed to flow in between north and east diections and between south and west directions. We also find all the three red signals glowing (high) in this case which means there is no traffic flow in between north and south directions, east and west directions, north-east and south-west directions.

Testbench for tress passing:

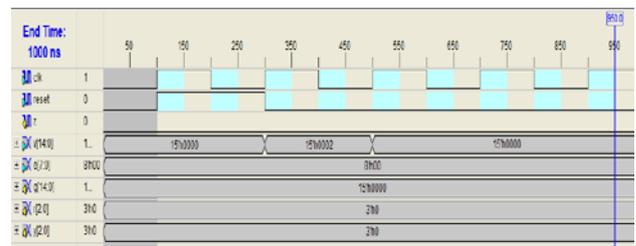


Figure 8.

Output for tress passing i.e, v[1]=1

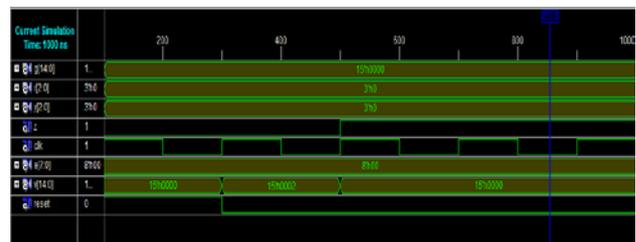


Figure 9.

When no condition to implement the FTC is satisfied as in above case, it is considered as tresspassing. When the vector v[1] is made high, it means that a vehicle is trying to go from east to west direction without getting the particular state. In this case none of the three signals glow implying that the vehicle is not allowed to pass. In this case the signal z goes high, indicating tress passing. This signal z can further be used to activate an alarm or a camera to detect the faulty person or vehicle.

8. Discussions

The various developments made in this work as compared to the given references are:

1. The reference paper was about a 4-way junction while our work is about a 6-way junction.
2. The number of intermediate steps needed in the FSM is less in our work as compared to the reference.

3. The combinations of paths taken in our work vary from that of the reference.
4. The reference paper considered only 4 traffic flow conditions while 8 possible traffic flow conditions are considered in our work.
5. The delay obtained after synthesizing the verilog code in our work is less as compared to the reference i.e, delay is 5.463ns for the reference paper while it is only 3.879ns in our work as we have considered a 6-way path instead of a 4-way path.
6. An additional state is considered in our work to detect the tress passers by initiating an alarm or a camera.

9. Conclusion

Fuzzy logic is a useful tool for solving many real-world problems characterized by imprecise and uncertain information. Fuzzy logic controller device potentially could be used for many applications especially in control engineering. Moreover, this FTC chip can also be further interfaced with other control peripheral hardware modules to control a more complex traffic system. This traffic controller has overcome the weakness of conventional traffic controllers, with the capability of providing varying green interval time based on dynamic traffic load changes, at every lane in a 6-way junction control. It not only saves time but also reduce the manual interruption in traffic control in this busy world where man is over-burdened with work. It has an advantage of detecting the tress passers and thus reducing the work of law. It proves itself more efficient as it uses Fuzzy technique. The verilog

code used to design this FTC is also very simple. The FSM, on which this works, is also easy to understand.

References

- [1] IEEE journal numbered 1-4244-0549-1, titled "Design and Implementation of an Intelligent Fuzzy VLSI Chip for Traffic Control Application" of 2006.
- [2] Pitu Mirchandani, Larry Head. "RHODES Traffic Adaptive Control System," Workshop on Adaptive Traffic Signal Control System, ATLAS Research Center, System and Industrial Engineering Department, The University of Arizona, 2001.
- [3] Nainar, M. Mohammadian, RJ Stonier, J.W.L. Millar. "An adaptive fuzzy logic controller for control of traffic signal," Proceedings of Fourth International Conference on Control, Automation, Robotic and Vision, (ACARCV'96) Singapore, 1996.
- [4] Richard P. Zanardo. "Digital Traffic Signal Controller," unpublished master's thesis, Bradley University, Department of Electrical and Computing Engineering and Technology, 1999.
- [5] Ralf Niemann. "Hardware/Software Co-Design for Data Flow Dominated Embedded System," University of Dortmund, Department of Computer Science XII, Germany, 1998.
- [6] Aleksandar Bobrek and Justin Britten. "FPGA Traffic Light Controller," unpublished master's thesis, Electrical & Computer Engineering, University of Tennessee, 2001.
- [7] Cheng, P.J Denning. "One-Lane Traffic Controller," unpublished master's thesis, George Mason University, 2001.
- [8] Caig Roberts, "Traffic Controller – Intersection Simulator Board," unpublished master's thesis, The Arizona Laboratory for Applied Transport Research, Northern Arizona University, 1998.
- [9] James Fernstrom, Thomas Wenisch. "Traffic Controller Design in PLA," unpublished master's thesis, University of Rhode Island, Kingston, 1998.