Proceedings of the 8th WSEAS International Conference on APPLIED MATHEMATICS, Tenerife, Spain, December 16-18, 2005 (pp257-262)

A Novel Computational Algorithm for Traffic Signal Control SoC

HSUNG-JUNG CHO* and MING-TE TSENG
Department of Transportation
National Chiao Tung University
TAIWAN
*Corresponding Author. E-mail: hjcho@cc.nctu.edu.tw

Abstract—This paper describes the key methodology in the design of a novel computational algorithm for traffic signal control in an oversaturated traffic network. The system is integrated with a computational traffic control algorithm and a system on chip (SoC) traffic controller. Traditional traffic network algorithms need many vehicle detectors and a high performance computer system for its complex optimal algorithm. The novel algorithm only needs critical approach detector's information and is simple enough to put into a SoC system which has fewer computing resource. Comparing to full actuated traffic controller, the simulation shows that this system has better traffic performance index, works on less resource system and needs fewer vehicle detectors.

Key-Words—Computational Algorithm, Traffic Signal, Controller, System on Chip.

1 Introduction

Urban road networks in many of today's world cities exhibit high levels of traffic congestion during peak periods. In oversaturated conditions, the traditional signal fixed-time plans tend to worsen the problem. And in order to get more information about how is the oversaturated condition, traffic managers intend to install more vehicle detectors around those intersections. And coordinated control of traffic signals to effectively manage traffic is a very complex task that involves numerous data processes and computations.

The design trend of today's controller is system on chip[1,2]. The traffic controller system is an application of SoC. The key module of the developed SoC system is the traffic control algorithm. Hence, we will focus on the development of the traffic control algorithm.

There have been a few offline signal optimization programs widely used which are: SOAP for single intersection, PASSER-II [3] for arterial, TRANSYT-7F [4] for network. These algorithms need working on a high performance and high resource computer.

Some adaptive signal controls were also developed. Full actuated algorithm is the first algorithm which uses full vehicle detectors to dynamically adjust signal plans. SCOOT [5, 6] was developed in the United Kingdom for the operations of systems of signals. SCATS [7] was developed in the 1970's by Australia. These algorithms need to install at least one vehicle detector on each link of the network. And they rely on more computing resource than offline signal optimization programs.

Signal optimization for oversaturated conditions has been studied since the 1960s. Gazis [8] proposed a graphical method to minimize total delay for two oversaturated, closely spaced intersections. D. Longley [9] suggested that each controlled intersection show adjust its green time split on the basis of queue length ratios on its various approaches. Abu-Lebdeh and Benekohal [10] developed a traffic and queue management procedure for oversaturated arterials. These algorithms also need working on a high performance and high resource computer.

Also, Cho and Lo [11, 12] formulate dynamic traffic model to simulate the traffic flow. Cho and Tseng [13] develop a system on chip approach to traffic signal control using a finite state machine concept.

Many vehicle detectors and high computing resource mean high cost. High cost prevents the city government's improving plans from carried out. Hence, traffic managers may ask following questions. Dose any algorithms work with fewer vehicle detectors? Or does the algorithm can work on a few computing resource systems like SoC? That is, the algorithm need lower cost and get better performance index.

The proposed novel traffic control algorithm uses contour line's concept to reduce to the number of vehicle detectors. And the algorithm uses only some simple matrix operations and 5 selecting rules to decrease the computing resource. And according the SoC concept, describing that the whole functionality of the system is placed on a single chip, we development the new computational intelligent algorithm on an ARM-SoC chip.

After simulation, it shows that the novel computational algorithm has better performance index (lower total link delay times and more balance) than traditional full actuated algorithm on an oversaturated network.

2 Deduction of the novel Computational algorithm

The scope of the novel traffic control algorithm is as Fig. 1. This network consists of a critical intersection which has two oversaturated approaches and some normal intersections. In order to reduce the number of vehicle detectors, traffic managers need to draw some contour lines here. Suppose the traffic manager draw three contour lines on the two oversaturated approaches of the critical intersection. Each contour line denotes same saturated level. Six vehicle detectors are installed at the six intersect points. (As Fig.1. shows) Vehicle detectors are used for detecting the presentation of the queue. The vehicle detector is installed on links which traffic flow has key impact to critical intersection.

After reducing the number of vehicle detectors, we need some method to adjust phase green time for each intersection. The key is to keep critical approaches' traffic queue length on same contour line. When the two approach queues are present on different contour line, the new algorithm will try to adjust green time to balance the two approach queues such that two approach queues are present on the same contour line finally. Outer contour line has longer traffic queue length, hence has higher discharge priority and longer green time. Same contour line's queue discharge order is based on the priority value given by traffic managers. The higher priority value's approach gets longer green time. The lower priority value's approach gets shorter green time.

The control algorithm for these five intersections can be classed into 2 different types. The first type is the critical intersection. The second type includes the other critical four intersections. The following two subsections discuss two methods for these intersections.

2.1 Critical intersection's control algorithm

The definitions of the symbols used in the deduction are as follows:

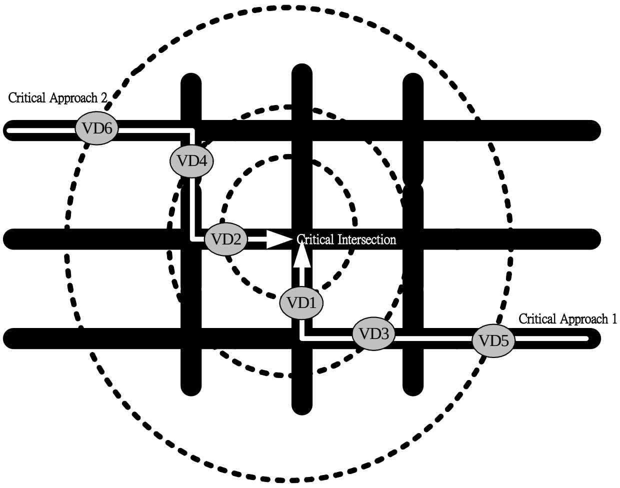

Fig. 1. System scope

C adjust constant given by traffic engineer

t_j the end of red time of phase j

$q^k_{VD}(t)$ the vehicle detector's queue state function at cycle time k, t

r_{VD_i} going through traffic volume ratio of vehicle detector i

$q^k_{VDi}(t_j)$ queue state function of vehicle detector i at cycle time k, t_j

Q^k_i phase queue state of phase i at cycle time k

p_{VD_i} priority value of vehicle detector i

P^k_i phase priority value of phase i at cycle time k

S^k_i sign of Q^k_i

g^k_i green time of phase i at cycle time k

$\min_{g_i}$ minimal green time of phase i

$\max_{g_i}$ maximal green time of phase i

First, we define a vehicle detector's queue state function at cycle time k as Function 1. The function value become positive when the vehicle detector detect some car stops in it's detecting area for more than 3 seconds. The function value become negative when the vehicle detector detect no car present in it's detecting area for more than 3 seconds. Fig. 2. gives an example for queue state function.

Function 1. queue state function of vehicle detector at cycle time k.

$$q^k_{VD}(t) = \begin{cases} S, \text{ when detect stopped queue} \\ -E, \text{ when detect empty queue} \\ 0, \text{ other.} \end{cases}$$

The following two definitions and one equation are used to compute the queue state of each critical approach.

Definition 1. Vehicle detector's queue state matrix at cycle time k

$$\begin{pmatrix} q^k_{VD1}(t_1) & q^k_{VD3}(t_1) & q^k_{VD5}(t_1) \\ q^k_{VD2}(t_2) & q^k_{VD4}(t_2) & q^k_{VD6}(t_2) \end{pmatrix}$$

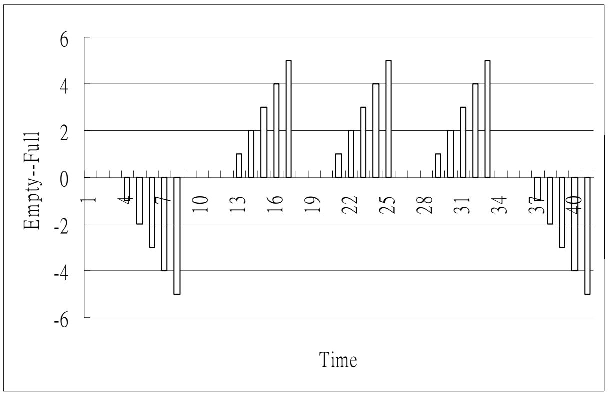

Fig. 2. Queue state function

Definition 2. Vehicle detector's go through ratio matrix

$$\begin{pmatrix} r_{VD1} & r_{VD3} & r_{VD5} \\ r_{VD2} & r_{VD4} & r_{VD6} \end{pmatrix}$$

Equation 1. Phase queue state matrix

We define an operator θ and phase queue state matrix $\begin{pmatrix} Q_1^k \\ Q_2^k \end{pmatrix}$ as:

$$\begin{pmatrix} Q_1^k \\ Q_2^k \end{pmatrix} = \begin{pmatrix} q^k_{VD1}(t_1) & q^k_{VD3}(t_1) & q^k_{VD5}(t_1) \\ q^k_{VD2}(t_2) & q^k_{VD4}(t_2) & q^k_{VD6}(t_2) \end{pmatrix} \theta$$

$$\begin{pmatrix} r_{VD1} & r_{VD3} & r_{VD5} \\ r_{VD2} & r_{VD4} & r_{VD6} \end{pmatrix} = \begin{pmatrix} \sum_{i=1,3,5} q^k_{VD_i}(t) \times r_{VD_i} \\ \sum_{i=2,4,6} q^k_{VD_i}(t) \times r_{VD_i} \end{pmatrix}$$

Definition 3 and equation 2 used to get the priority value of critical approach.

Definition 3. Vehicle detector's queue priority matrix

$\begin{pmatrix} p_{VD1} & p_{VD3} & p_{VD5} \\ p_{VD2} & p_{VD4} & p_{VD6} \end{pmatrix}$ The priority values are given by traffic engineer

Equation 2. Phase priority state matrix at cycle time k.

We define an operator Ψ and phase priority matrix at cycle time k $\begin{pmatrix} P_1^k \\ P_2^k \end{pmatrix}$ as:

$$\begin{pmatrix} P_1^k \\ P_2^k \end{pmatrix} = \begin{pmatrix} q^k_{VD1}(t_1) & q^k_{VD3}(t_1) & q^k_{VD5}(t_1) \\ q^k_{VD2}(t_2) & q^k_{VD4}(t_2) & q^k_{VD6}(t_2) \end{pmatrix} \Psi$$

$$\begin{pmatrix} p_{VD1} & p_{VD3} & p_{VD5} \\ p_{VD2} & p_{VD4} & p_{VD6} \end{pmatrix},$$

where

$$P_i^k = \begin{cases} p_{VD_j}, \text{where } j \text{ is the } \max \text{ value of } i\ q^k_{VD_l}(t_i) > 0 \\ 0, \text{if } q^k_{VD_l}(t_i) \le 0\ \forall l \end{cases}$$

After computing the priority and queue state of each critical approach, we need some rules to dynamic change the green time of each critical approach. The following are those cycle time and split time adjust rules:

Rules 1:

If $\begin{pmatrix} S_1^k \\ S_2^k \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ then

$$g_1^{k+1} = \begin{cases} g_1^k - CQ_1^k, & if \min_{g1} \le g_1^k - CQ_1^k \\ \min_{g1}, & else \end{cases}$$

$$g_2^{k+1} = \begin{cases} g_2^k + CQ_2^k, & if \max_{g2} \ge g_2^k + CQ_2^k \\ \max_{g2}, & else \end{cases}$$

Rules 2:

If $\begin{pmatrix} S_1^k \\ S_2^k \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ then

$$g_1^{k+1} = \begin{cases} g_1^k + CQ_1^k, & if \max_{g1} \ge g_1^k - CQ_1^k \\ \max_{g1}, & else \end{cases}$$

$$g_2^{k+1} = \begin{cases} g_2^k - CQ_2^k, & if \min_{g2} \le g_2^k + CQ_2^k \\ \min_{g2}, & else \end{cases}$$

Rule 3.

If $\begin{pmatrix} S_1^k \\ S_2^k \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ then $\begin{pmatrix} g_1^k \\ g_2^k \end{pmatrix} = \begin{pmatrix} g_1^0 \\ g_2^0 \end{pmatrix}$. Where g_i^0 is originally green time of phase i, i=1, 2.

Rule 4.

If $\begin{pmatrix} S_1^k \\ S_2^k \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $P_1^k \geq P_2^k$ then

$$g_1^{k+1} = \begin{cases} g_1^k + CQ_1^k, & \text{if } \max_{g1} \geq g_1^k - CQ_1^k \\ \max_{g1}, & else \end{cases}$$

$$g_2^{k+1} = \begin{cases} g_2^k - CQ_2^k, & \text{if } \min_{g2} \leq g_2^k + CQ_2^k \\ \min_{g2}, & else \end{cases}$$

Rule 5.

. If $\begin{pmatrix} S_1^k \\ S_2^k \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $P_1^k < P_2^k$ then

$$g_1^{k+1} = \begin{cases} g_1^k - CQ_1^k, & \text{if } \min_{g1} \leq g_1^k - CQ_1^k \\ \min_{g1}, & else \end{cases}$$

$$g_2^{k+1} = \begin{cases} g_2^k + CQ_2^k, & \text{if } \max_{g2} \geq g_2^k + CQ_2^k \\ \max_{g2}, & else \end{cases}$$

We summarize these rules as Fig. 3. The higher priority approach get more close to maximal green time, the other get more close to minimal green time.

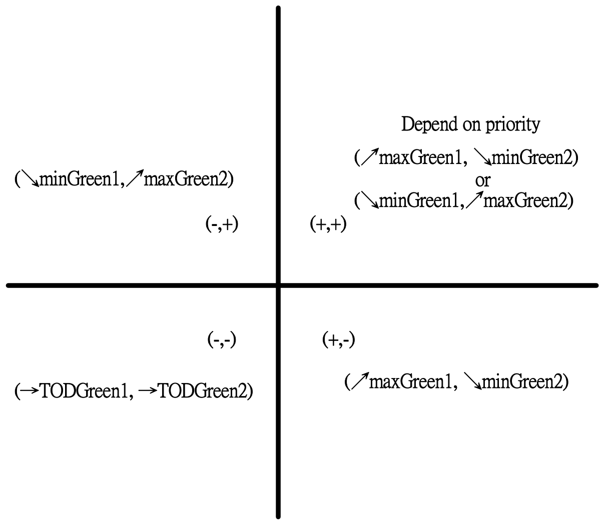

Fig. 3. Rules summarization

Let's give a more general example and summarize the output of the adjustment of phase times. When the queue present sequence is VD1 to VD6 at traffic jam time (as Fig. 4.), the phase1's green time will switch from maximal to minimal and phase2's green time will switch from minimal to maximal. When the queue disappear sequence is VD6 to VD1 at off-peak time(as Fig. 5.), the phase1's green time will switch from minimal to maximal and phase2's green time will switch from maximal to minimal.

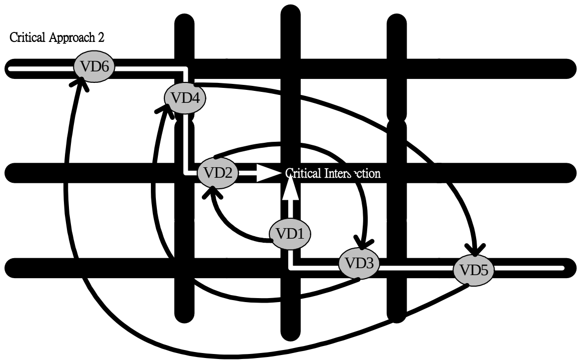

Fig. 4. The queue present sequence: VD1 to VD6

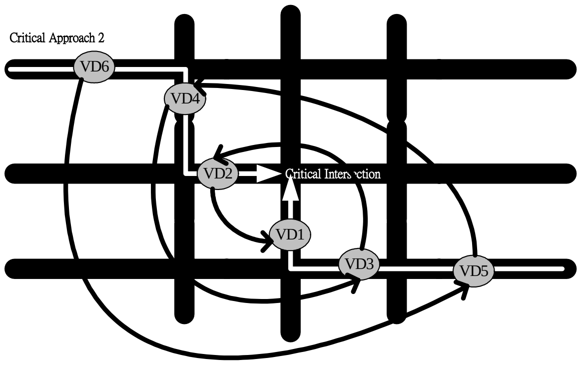

Fig. 5.Queue disappear sequence: VD6 toVD1

2.2 The other intersection's control algorithm

The second type intersection's control algorithm can be the same complex as the critical intersection or more simple. It depends on the number of traffic flows merging into critical intersection's critical approaches. If there are two or more traffic flows merging into critical intersection's critical approach then the algorithm will almost be the same as the critical intersection. If there is only one traffic flow merging into critical intersection's critical approach then the algorithm will be more simplex. Let's describe the simple case. The cycle time and critical approach's green time is the same as the critical intersection. The offset is as the following equation.

Equation 3. The offset

$$offset_i = \begin{cases} \dfrac{l_{VD_i}}{v_{start-wave}} - \dfrac{l_{link\,i} - l_{VD_i}}{v} + t_j, & \text{if VD}_\text{i} \text{ detect queue} \\ 0, & \text{else} \end{cases}$$

where

i is from 1 to 6,

l_{VD_i} is the distance between VD_i and downstream intersection,

$l_{link\,i}$ is the link length of VD_i (as Fig. 6.),

$v_{start-wave}$ is the velocity of start wave,
v is the velocity of car's speed,
t_j is the delta time between the start time of critical approach phase j and the start time of phase 1.

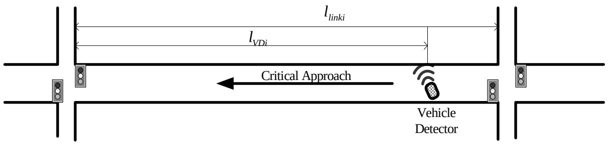

Fig. 6. Link and vehicle detector distance

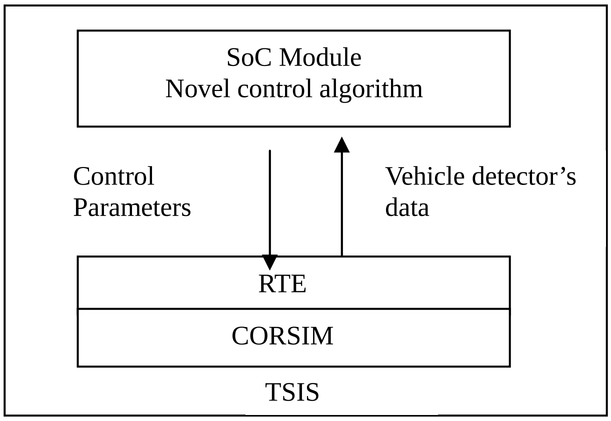

Fig. 7. Simulation framework

3. Simulation results

Before putting the algorithm into the SoC chip, the simulation job should be done. We use TSIS with ARM-SoC module to simulate the computational algorithm. The simulating framework is as Fig. 7. The left block is TSIS environment. The right block is ARM-SoC Module.

The TSIS is run on a Microsoft windows environment. TSIS has a microscope simulating tool called CORSIM. CORSIM provides cars and drivers behavior simulation. It also can output vehicle detector's data to some other program. All simulation jobs almost are done in TSIS except the traffic signal control algorithm.

The new computational traffic control algorithm has developed on the ARM-SoC module which has ARM7 SoC chip and uc-Linux inside. The SoC chip is the S3C4510B. It is a 16/32-bit RISC microcontroller. It offers a configurable 8K-byte unified cache/SRAM and Ethernet controller. Important peripheral functions include two HDLC channels with buffer descriptor, two UART channels, 2-channel GDMA, two 32-bit timers, and 18 programmable I/O ports. On-board logic includes an interrupt controller, DRAM/SDRAM controller, and a controller for ROM/SRAM and flash memory. All network traffic signal control job is done by the ARM-SoC module.

The communication interface is by Ethernet. The RTE block in TSIS plays a key role to do the communication job between CORSIM and the ARM-SoC module. When CORSIM generates some vehicle detector and network data, RTE will sends these data to ARM-SoC module. After applying the computational traffic control algorithm, the ARM-SoC Module send traffic signal control parameters back to RTE. And RTE return these parameters to CORSIM.

In order to compare with some other traffic control algorithm, we do the simulation of the traditional full actuated controller. The full actuated algorithm is only simulated in TSIS, not with ARM-SoC module.

Suppose the approach 2 traffic flows are oversaturated and approach 1 is under-saturated. The total number of the vehicle detectors of the novel computational algorithm is 6 and the full actuated is at least 20 (as Fig.8.). The performance index is focus on "total delay time (vehicle-minutes)". After 30 times simulating for two algorithm, Tab.1. and Fig.9. show the comparison between these two algorithms. They show that the new computational intelligent algorithm has lower total delay time and smaller difference between approach 1 and approach 2. Smaller difference means the new computational intelligent algorithm has more balanced total delay time. The balanced delay time will make two approach drivers feel more fairness.

Hence, the results show that the novel computational algorithm need fewer vehicle detectors and gets better performance index for those critical approach intersections.

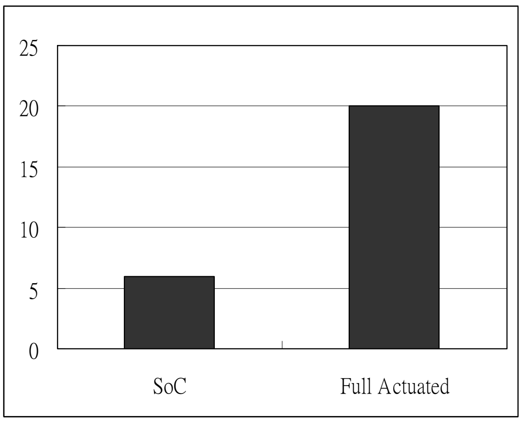

Fig. 8. The number of vehicle detectors

	SOC	Actuated Control
Approach2(Link2+4+6)	22485.6	40335.7
Approach1(Link1+3+5)	18769.8	2236.0
TotoalDelayTime (Approach1+2)	41255.4	42571.7

Tab. 1. Simulation Result: total delay time

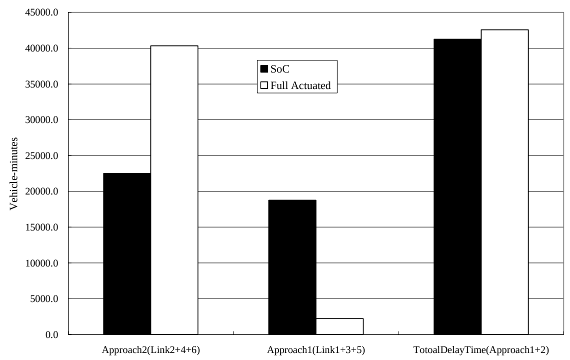

Fig 9. Simulation result: total delay time

4 Conclusions

Traditional traffic control algorithm needs many vehicle detectors and involves numerous data processes and computations. We propose a novel algorithm which only need fewer vehicle detectors and lower computing resource.

The novel computational traffic control algorithm is focused on an oversaturated traffic network which has a critical intersection and some critical approaches. The first step of the algorithm is to reduce the number of vehicle detector by contour lines concept. The Next step is to adjust the phase green time according Rule 1 to Rule 5.

The simulation of the novel computational traffic control algorithm shows that this computational algorithm need fewer vehicle detectors and gets better performance index (lower total vehicle delay time and more balance) than full actuated traffic controller. And the algorithm can works on a system on chip controller. Hence, the algorithm is low cost to implement.

It is recommended that the novel computational algorithm can be combined with some existed under-saturated adaptive algorithm to manage full scope of a traffic network. In order to put the existed algorithm into a SoC Chip, the existed under-saturated algorithm should be modified. The combined algorithm need to be validated through microscopic simulation, and then tested through field implementation.

Acknowledgements

This work is supported in part by the Ministry of Education of Taiwan, ROC, under grant EX-91-E-FA06-4-4 and in part by the National Science Council of Taiwan under contracts: NSC-93-2218- E-009-040 and NSC-93-2218-E-009-042

References:

[1]Benini, L. and De Micheli, G.,Networks on chips: a new SoC paradigm, Computer, Volume 35, 2002, pp.70 - 78

[2]Tummala, R.R., SOP: what is it and why? A new microsystem-integration technology paradigm-Moore's law for system integration of miniaturized convergent systems of the next decade, Advanced Packaging, IEEE Transactions on ,Volume 27, 2004, pp.241 - 249

[3] Passor II-90 Users Guide Volume 3 in a Series: Methodology for Optimizing Signal Timing, December 1991

[4] TRANSYT-7F User Guide Volume 4 in a Series: Methodology for Optimizing Signal Timing, March 1998.

[5] Bretherton, D, “Current developments in SCOOT: version 3”, Transportation and Research Record 1554, 1996, pp48-52.

[6] Bertherton, R.D., K. Wood and G.T.Bowen, “SCOOT Version4”, Transportation Engineering & Control, 39(7), 1998, pp425-427.

[7] Lowrie, P.R. “SCATS-Syndney Coordinated Adaptive Traffic System-A Traffic Responsive Method of Controlling Urban Traffic.”, Roads and Traffic Authority, Sydney, NSW, Australia, September 1992.

[8] Gazis, D.C., Optimum Control of a System of Oversaturated Intersections, Operation Research, Vol. 12, No. 6, 1964, pp. 815-831.

[9] Longley, D., A Computer Strategy for a Congested Computer-Controlled Traffic Network, Transportation research, Vol.2, 1968.

[10] Abu-Lebdeh, Ghassan and Rahim F. Benekohal, Development of Traffic Control and Queue Management Procedures for Oversaturated Arterials, Transportation Research Record, No. 1603, 1997, pp119-127.

[11] Cho, H. J., and S. C. Lo, Modeling of Self-consistent Multi-class Dynamic Traffic Flow Model, Physica *A*., Vol. 312, pp. 342-362, 2002.

[12] Cho, H. J., and S. C. Lo, Monotone Iterative Simulation of Traffic Dispersion Model, Computer Physics Communication., Vol. 148, 2002, pp. 291-298

[13] Cho, Hsun-Jung and Ming-Te Tseng, A system on chip approach to traffic signal control, The international conference on computer design, 2005, on publishing.