Implementation of a Basic Flow Model in Traffic Controlling

Sándor Gacsal

Budapest Tech, Bécsi út 94-96, H-1034 Budapest, Hungary
gacsal.sandor@gmail.com

Timót Hidvégi

Széchenyi István Egyetem, Egyetem tér 1, H-9026 Győr, Hungary
hidvegi@sze.hu

Abstract: The contribution introduces a basic model, a manner, how to control – using the specific tools – a crossing of a highway and a lower volume road, while we stop the highway's traffic only in a critical case otherwise just slow it down.

1 Introduction

During our presentation we will introduce a kind of a basic model, whereby we are going to be able to reduce the chance of traffic congestion in a crossing. Seeing that in our future plans we would like to try out and test several solutions, achieving various models are easy to solve using the test environment we shaped. It seems plausible to use different programmable logics like CPLD's or FPGA's. Henceforth we will introduce some of the simplified solutions of solving our ambitions. In the forth chapter you'll find the description of the test environment. Lastly the fifth chapter contains the conclusions or rather the future plans.

2 Introducing the Crossing

2.1 The Crossing

In our concept (Figure 1) we have two roads crossing each other. A highway ("A") and a lower volume road ("B"). The traffic's direction on each road is shown by the figure. Both roads have traffic lights ("a1" and "b1") at the entrance of the crossing.

2.2 The Ambition

What we would like to try for - with our model - is, to keep the traffic arriving from "A" direction in a continous flow, while we let the "B" traffic to pass the crossing also. Nevertheless we would like to achieve this by ruling the speed of "A" traffic, or in a critical case stopping it.

To obtain the ruling of "A" traffic we placed two displays/indicators over the "A" road, and these displays are going to inform the traffic about the actual speed limitations. If the participants of the "A" traffic keep their vehicles driving with the allotted velocity, they are going to get green light on the arrival to the crossing.

The critical case is when the "B" traffic is congested in such a volume, where we are not able to let "B" traffic to cross the road without stopping vehicles coming from "A" direction. In our concept, the monitoring of the congestion of the "B" road would be achieved by three sensors ("b2","b3" and "b4").

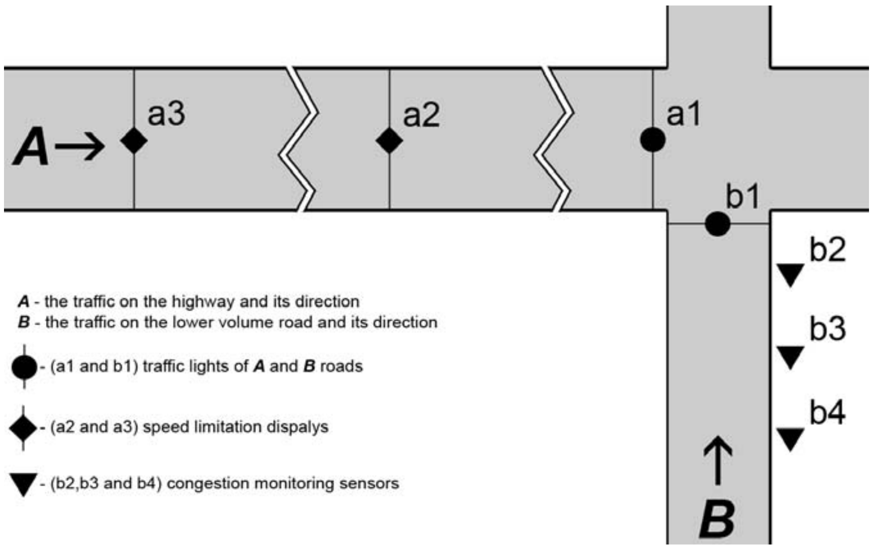

Figure 1

The block-scheme of the crossing

3 Simplified Solutions

3.1 Solution A: Using a Timer

In a crossing there are several solutions to control the traffic. If it's allowed to stop both "A" and "B" traffics, we could simply use a timer and letting one traffic to pass the crossing, and than change and let the other one to do the same (Figure 2).

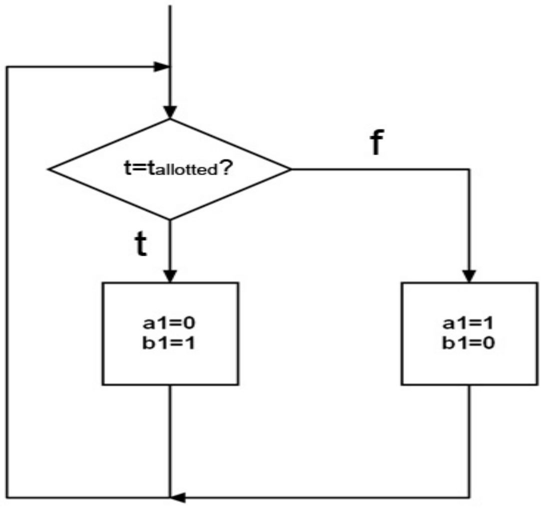

Figure 2
The simplified method of using the timer

But this solution is disadvantageous and could cause various problems. Put the case that there are no vehicles coming from one direction, and on the other road there is a line of cars waiting for the traffic lights to change from red to green sign. In this case we are needlessly keeping the car line waiting instead of letting them to pass the crossing.

3.2 Solution B: Based on the Traffic Congestion Monitoring

Another solution could be if we would base our decision on the informations received from the sensors and we would let that traffic to pass the crossing wich has a higher volume at the moment (Figure 3).

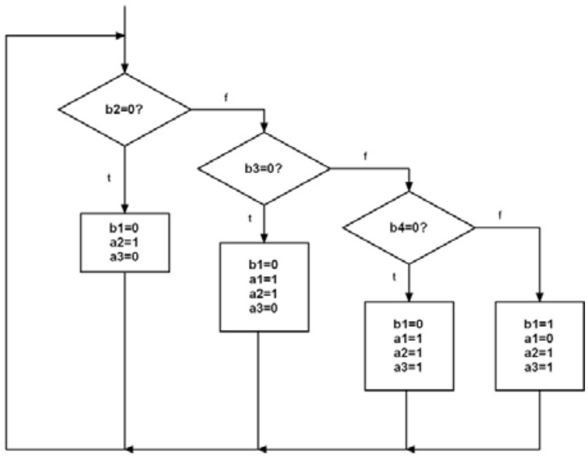

Figure 3
The simplified method of the monitoring based solution

The problem with this solution is that, if there is a higher volume of traffic coming from "B" direction for some reason, the "A" traffic would often or worse, constantly get red sign, so "A" road would loose it's priority against "B" road.

3.3 Solution C: the Practical Combined Solution

The third, wich seems to be the most practical solution for us, combines the two methods mentioned above. It uses the timer and the congestion monitoring also. First of all let's distinguish four parts of the day, as four working periods. The first is the night-time, second is the morning rush hours, third is the calm period in the afternoon and the fourth is the evening rush hours. I nour concept both first and third periods could work with a simple timer based solution. The difference between these periods would be sensible in the lenghts of the green lights, because usually the night-time traffic is less than in the afternoon.

In the second and the fourth period we would like to use the timer and the congestion monitoring together. The congestion monitoring would inform the system about the length of the line of cars waiting on the "B" road, wich is on a lower level of priority. With this information we could control the crossing in the way mentioned below (Figure 4).

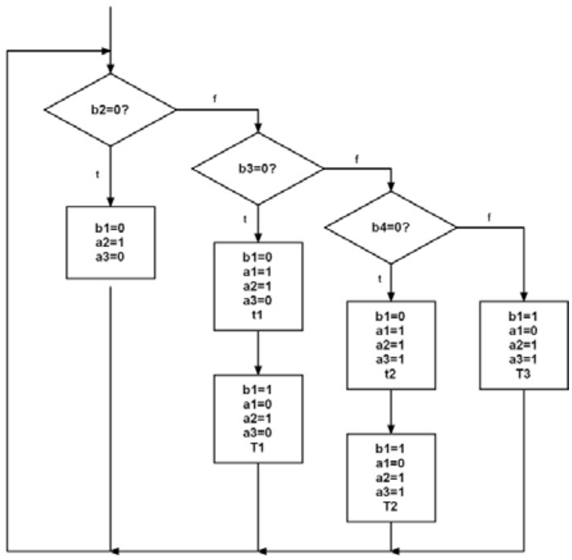

Figure 4
The simplified method of using the timer and the congestion monitoring together

3.4 The Operation of the Congestion Monitoring on Road "B"

There are three sensors ("b2","b3" and "b4"/ figure 5.) placed along the road ahead the traffic lights ("b1"). If a car passes "b4" sensor, the monitoring system notes wich other sensors are passed by the same vehicle, so we can locate the car between "b4" and "b1" points. If the car passes all three sensors ("b4","b3" and "b2") and the traffic lights ("b1") show red sign, than it is located between "b2" and "b1" points. In case "b2" does not sense the car, than it has to be between "b3" and "b2". This means the line of cars started to grow. The growth of this line reaches the critical value if a vehicle passes only "b4" and no other sensors. In this case the monitoring system indicates that there is a higher density of traffic on "B" road, so we have to stop the traffic coming from "A" direction.

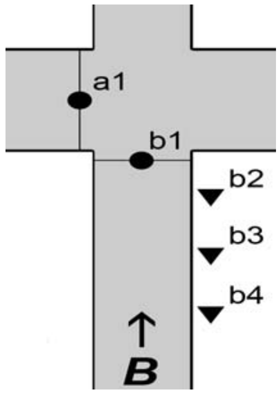

Figure 5
The placing of the sensors

3.5 The Duration of the Green Signs for "B" and the Speed Ruling of "A" Traffic

If the monitoring says the line of cars on "B" road is between "b1" and "b2", we have to change "b1" traffic lights to green for a shorter amount of time (T1) and inform the traffic arriving from "A" direction about the speed limitation to X km/h only on the "a3" display. If the end of the line of cars is between "b2" and "b3" then it's necessary to give green sign for a longer T2 time (T2>T1). In this case first we have to inform the "A" traffic about the speed limitation to X km/h on "a3" display, then to Y km/h on "a2" display (X>Y).With the speed limitations we gain a gap between the vehicles already ruled and the vehicles that did not receive this kind of notifications (Figure 6).

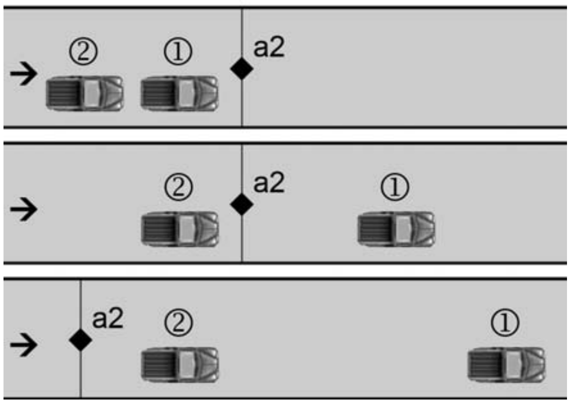

Figure 6
The safety gap's growth after notification

The size of the gap we need to gain is defined by the line of cars located at "B" road. The gap has to be such wide as wide we need it to be to let the "B" traffic to cross the "A" road before this gap ceases within the specified time. As figure 6. shows, both cars pass "a2" display but only car No.2 is passing it under limitation. For the simple reason that there is a difference between the two cars' velocity, after passing the display the safety gap between these vehicles starts to grow, generating the needed gap.

4 Test Environment

When we were developing the test environment, it was an important viewpoint for us to keep the system available for upgrading with various hardwares. So our choice happened to be the PC-104 standard. During the development of the test card [5][6] we modified the BUS interface. The modified cards block-scheme is shown by Figure 7. This card is attachable to computers thru its USB port.

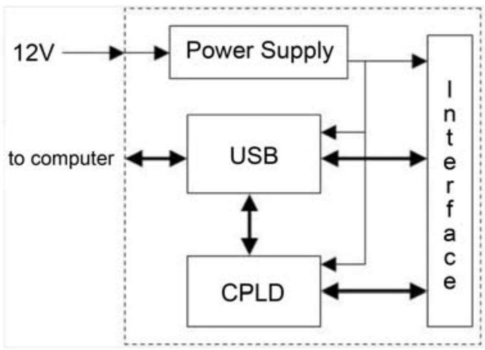

Figure 7
The motherboard of the test environment

This testcard is the motherboard of the system, on this card we generate several voltages for the function, and the linkage to computers is also this card's role. Having a CPLD[3] on this panel allows us to try out some smaller, simplier models using only this card. Other testcards are connected to this motherboard for further functions.

This testcard is the motherboard of the system, on this card we generate several voltages for the function, and the linkage to computers is also this card's role. Having a CPLD[3] on this panel allows us to try out some smaller, simplier models using only this card. Other testcards are connected to this motherboard for further functions.

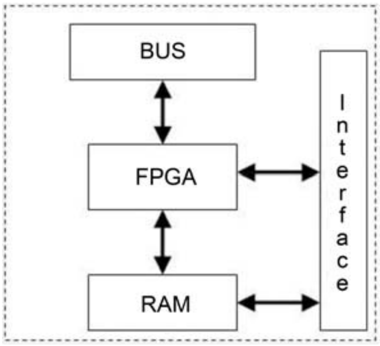

Figure 8
Arithmetic card using FPGA

Figure 8. shows the conformation of the test environment. With this system complex solutions are available to be tested, and the efficiancy of the SPARTAN 3 [3] FPGA allows us to implement architectures [2] based on complicated calculations.

Conclusion

Inasmuch this conception is in a inchoative stage, our future plans are to upgrade the model and its execution and to optimize it. We would like to implement several methods like a traffic flow model described by a non-linear partial differential equation [1]. With this method we could optimize further the ruling of traffic "A".

References

[1] Tyn Myint, Lokenath Debnath (1987), „Partial Differential Equations for Scientists and Engineers", *Elsevier Science Publishing*, New York

[2] Kai Hwang (1979), „Computer Arithmetic Principles, Architecture, and Design" *John Wiley & Sons*, New York

[3] *www.xilinx.com*

[4] Gábor Marton, Timót Hidvégi (2006), „Professional, Configurable Development System for Microcontroller, FPGA and its aplication" *7th International Symposium of Hungarian Researchers on Computional Inteligence*, Budapest, pp. 445-450

[5] Hidvégi Timót (2007), „Útjelzések, akadályok felismerése valós időben", *Innováció és fenntartható felszíni közlekedés*, Budapest