Multiprocessor Architecture for Adaptive Control of Traffic Light Systems

Dr. J.A.K.S. Jayasinghe, University of Moratuwa
Multiprocessor Architecture for Adaptive Control of Traffic Light Systems

Dr. J.A.K.S. Jayasinghe, University of Moratuwa

Abstract

Construction of new highways or expansion of existing highways to meet the ever-increasing highway traffic is a very expensive and time-consuming task. Management of existing highways effectively is considered as a high priority option in Traffic and Highway Engineering. Traffic control by a system of traffic signal lights and variable message signs that can adapt to the changing environment is an essential part of such management scheme. The multiprocessor architecture described in this paper maintains near neighbor connectivity and shares information via these near neighbor interconnections for calculation of optimum traffic plans. The near neighbor connectivity simplifies the expansion of the multiprocessor architecture with city developments.

1. Introduction

Highway traffic control by means of traffic signal lights that can adapt to the changing environment is an essential part of managing the ever-increasing highway traffic. Management of highway traffic is considered very attractive, as construction of new highways or expansion of existing highways to meet the ever increasing highway traffic is a very expensive and time consuming task. In such a traffic management system, time duration allocated to each phase of traffic signal lights installed at each junction is calculated by obtaining the instantaneous data related to vehicle flow rates, timing of other traffic signals, etc. to minimize the delays caused to the motorists. Sensors such as loop detectors, CCTV cameras, etc. provide necessary data. In case of congestion, traffic is diverted with the aid of Variable Message Signs (VMS). Calculation of signal timing with instantaneous traffic flow parameters is a computational intensive task that needs a multiprocessing environment with the flexibility to adapt to city developments. This paper describes a multiprocessor architecture suitable for this purpose.

2. Multiprocessor Architectures for Real-time Computing

Multiprocessor architectures are widely employed in many computational intensive problems, when a single processor is unable to perform the required calculations within
the required time frame. The calculation of traffic plans for a given set of traffic signal controllers based on the instantaneous traffic flow parameters is a computational intensive task as these calculations must be performed in real-time. The real-time computational systems are classified as hard real-time systems and soft real-time systems [1], [2]. According to these classifications, a soft real-time system must produce results as fast as possible such that statistically described response time is satisfied. A hard real-time system must produce results before a given set of deadlines. Online timing calculation for a traffic signal control system is classified as hard real-time because the computational results must be available before the next timing cycle is commenced.

Parallel processing on a set of processors in a coordinated manner to achieve the required task is defined as multiprocessing. Multiprocessing provides a viable means to achieve a huge computational power. Multiprocessing has been a key research area in the field of computer science and engineering, and several types of architectures have been invented. Time-shared Bus Architecture, Multi-ported Memory Architecture, Cross-point Switch Architecture, Array Processor Architecture are some examples. The performance of these architectures for a given real-time problem varies significantly as the required data transfer patterns are better matched to the architectural features of some of these architectures.

According to [1], a multiprocessor system for hard real-time problem must possesses features such as homogeneity, scalability, flexibility and survivability. The array processor architecture is a good candidate for hard real-time computing as it inherits these features. In array processor architectures, a set of identical processors is used while each processor is interconnected to its near neighbor processors. Further, each processor executes its program when the required inputs are supplied by the relevant near neighbors. Although array processors are used in many applications, it has not been reported in literature for traffic signal control applications. Several systematic array processor design methodologies have been reported in the literature [3], [4], [5], [6]. Based on the design methodology reported in [5], novel array processor architecture for the adaptive control of traffic light systems is presented in this paper.

3. Multiprocessor Architecture for Adaptive Control of Traffic Light Systems

Figure 1 describes the topology of the traffic control system where the multiprocessor architecture is implemented. Electronic Traffic Controllers (ETC) control the traffic signal lights and Variable Message Signs (VMS). ETCs are located at key road intersections and conventional ETCs with remote control facility via an external input/output port are used in this topology. Each ETC is connected to the Control Center (CC) via a communication link and the communication link provides a data communication path between the ETC and the CC. The multiprocessor system that performs the timing calculation is physically located at the CC.
Due to the extensive nature of the design process, the design process is outlined with a simple example and the details of the resulting architecture are reported in this paper. The computational tasks carried out for timing calculations are first described as a Structured Single Assignment Code (S^2AC) for constructing the Structured Dependence Graph (SDG) [5]. The S^2AC consists of a set of hierarchical routines where each routine is described by a header and a body. Only single assignments are made to every variable in the S^2AC. The SDG is the graphical representation of the S^2AC, where nodes represent the assignments (calculations) and edges represent the dependence between variables. This SDG is expanded and non-linearly mapped into an uni-processor type array processor. To simplify the design process, the mapping is carried out on a scaled dependency graph as described in [5].

The S^2AC for a traffic signal timing adaptation scheme where green time for a given road is proportional to the estimated queue length can be described as:

```plaintext
[T_{k+1}] = TIME_CALCULATE (T_k)
{[T_{k+1}, T_k] : ARRAY[1..M] OF INTEGER

V = FLOW()
C = RED_TIME(T_k)
T_{k+1} = TIME_ADAPT(V, C)

END
}
```

The single assignment code for the FLOW routine is as follows:

```plaintext
[V_i] = FLOW()
{[V_i] : ARRAY[1..M] OF INTEGER

FOR j = 1 to L
   F_{1j} = F_j
   /* Assign flow rate of the flow sensor j to single assignment code variables */
END

FOR i = 1 to M
   X_{i,0} = 0
   FOR j = 1 to L
      X_{i,j} = X_{i,j-1} + A_{i,j} * F_{i,j}
      /* Estimate flow rates */
```

30
END

\[ F_{i+1,j} = F_{i,j} \]

\[ V_i = X_{i,M} \]

/\* Propagate flow sensor values to nodes *\/

END

In the FLOW routine, the flow rate of the \( i^{th} \) road is estimated from the measured flow rates with the aid of the weighting factors \( A_{ij} \). This weighting factor is proportional to the number of vehicles coming to road \( i \) from the \( j^{th} \) sensor location. The RED TIME routine determines the red time duration where vehicles are queued up using the single assignment code. This time duration is obtained by adding the green time of other roads coming to the same intersection. Similarly, the TIME ADAPT routine estimates the queue length with the aid of the flow rate and red time duration and determines the green time.

Figure 3 depicts expanded SDG for the above \( S^2AC \) in the scaled down form. Now we map the nodes related to each junction in the road network where ETCs are installed into a single processing element. We label this processing element as the Logical Processor (LP). Figure 4 depicts the logical interconnection scheme of the resulting multiprocessor architecture for the \textit{hypothetical city} \( A \) depicted in Figure 2. In this architecture, the logical data communication between each ETC is structured as near neighbor communication via Logical Links (LL) as depicted in Figure 4. In this, a LL is established between each ETC pair located at intersections when there is a path through the road network subject to the condition that only the source ETC and destination ETC exists at the intersections along the path traversed through the road network. Sharing information via near neighbor interconnections facilitates calculation of signal timing which minimize the delays to the motorists.

By mapping the expanded structured dependence graph, logical processor element is created for each ETC. Each ETC is connected to the relevant ETC via a Physical Link (PL) through the communication network under the control of a Communication Controller (ComC). In order to establish the LLs in the architecture, the required LPs are interconnected via Back-plane Link (BL). This interconnection scheme is depicted in Figure 5, for the hypothetical city \( A \) as an illustration (Note: Only BLs for LP 1 is shown due to large number of BLs).

In this multiprocessor architecture, timing is calculated on each LP by getting the information from other LPs. When a LP requires data from other LPs, the relevant LPs are informed via the corresponding BLs and then each LP contacts the ETC under its control via its PL. Once the data is received from the ETC, each LP sends data back to the LP that requested data. Once the relevant data is received, LP calculates the proper timing for the ETC under its control and new timing data is downloaded to ETC. This process is executed in parallel, in all LPs in coordination with each other.
The PLs are implemented either as dial up lines, leased lines via the public switched telephone network (PSTN) or dedicated point-to-point data communication links, depending on the data communication requirements.

The multiprocessor architecture described above, and illustrated in Figure 5 for the hypothetical city A, has a large flexibility for implementation due to near neighbor connectivity. Depending on the degree of complexity of the control algorithms, physical CPUs can be assigned to LPs on the following basis in decreasing order of the complexity of algorithms executed on LPs:

1. One physical CPU to each LP
2. One physical CPU to a set of LPs and execute the operations by time sharing
3. One physical CPU to all LPs and execute the operations by time sharing

In case of option 1, a processor array is interconnected as depicted in Figure 5. For option 2, a set of LPs is grouped together. In this case, BLs between the LPs in different groups are physically implemented while BLs within the same group of PLs are logically implemented by transferring data between time shared processes executed on the CPU. In the case of option 3, all BLs are implemented by transferring data between time shared processes executed on the CPU. Note that option 3 is only suitable for a simple traffic control system with few ETCs while option 1 is best suited for a complicated traffic control system with many ETCs.

The major advantage of this architecture is its flexibility. As described above, it can be used for a simple traffic control system with few ETCs or with a complex traffic control system with many ETCs. Further, this flexibility enables new CPUs to be added when more ETCs are installed with the development of the city or simple traffic control algorithms are replaced by complicated traffic control algorithms to improve the performance of the system with ever increasing road traffic.

Based on this multiprocessor architecture, a traffic control software package has been developed at the University of Moratuwa to coordinate traffic signal controllers [7]. It has been developed on a single CPU by executing the LPs on time-sharing basis as described in option 3. This software package has been tested extensively and plans have been made to coordinate traffic signal systems between the Fort railway station and Maradana technical college as a pilot project by the Road Development Authority.

4 Conclusion

The multiprocessor system described in this paper integrates all traffic signal controllers at intersections installed in a city to a control center such that the control center can adapt
the traffic signal timing in real-time to cater for varying traffic volumes. This architecture has been used in the advanced traffic control system (ATCS) developed at University of Moatuwa for the City of Colombo.

References


---

1 The architecture described in this paper is protected by Patent number 11607 issued by Registrar of Patents and Trade Marks, Democratic Socialist Republic of Sri Lanka.
Figure 1 Topology of the Multiprocessor Adaptive Traffic Control System

Figure 2 Hypothetical City A
Figure 3 The Expanded SDG for a Simple Traffic Signal Timing Adaptation Scheme
Figure 4 Interconnection Scheme with Logical Links (LL)

Figure 5 Multiprocessor Architecture for Adaptive Traffic Control