CHAPTER I

INTRODUCTION

Most real-time control applications demand high reliability, high availability, and high efficiency in data collection, data communication, and data analysis. With recent advances in the area of VLSI designs, it has become increasingly feasible to implement multiple-microprocessor system. The multiple-microprocessor system may be used to meet the performance requirements of a real-time application which exceeds the computing capacity of a single microprocessor system [1,2,3]. Examples of applications that require multiple-microprocessor systems are robotic control system [4], speech and vision processing [5], computer aided manufacturing [6], supervisory control and data acquisition system [7], etc.

1.1 Classification of Multiple-Microprocessor Systems.

Multiple-microprocessor systems can be divided as tightly coupled and loosely coupled [3]. A tightly coupled multiple-microprocessor system is usually called a shared-memory multiple-microprocessor system. The shared-memory multiple-microprocessor system contains a number of microprocessors that share part of the available memory. The microprocessors can execute distinct programs as well as cooperate in the same application by using the shared memory for interprocess communication. This makes them suitable for a broad range of applications. In a loosely coupled multiple-microprocessor system, each microprocessor has a large local memory where it accesses most of the instruction and data. Programs which execute on different computer modules can communicate

by exchanging messages through an interconnection network. The degree of coupling in such a system is very loose. Hence, it is offen referred to as a distributed system. The different between the loosely coupled system and the shared-memory system can be seen from Fig. 1.1 .

1.2 Multiple-Microprocessor System Interconnection Networks.

There are three main types of interconnection networks for multiple-microprocessor system : crossbar, multistage, and common bus [8] .

1.2.1 Crossbar Interconnection Network.

A crossbar interconnection network is an array of individually operated contact pairs in which there is one pair for each input-output combination, as shown in Fig. 1.2 . A crossbar network with N inputs and M outputs is referred to as an NxM crossbar. A crossbar supports all possible distinct connections between the microprocessors and memories simultaneously. However, the cost of such a network is O(NxM) for connecting N inputs and M outputs. For a system with a large number of microprocessors and memories, the cost of such an interconnection network is prohibitively high.

1.2.2 Multistage Interconnection Network.

As an alternative to the crossbar network, a multistage interconnection network has assumed importance in recent times. The main advantage of these networks is their cost-effectiveness. They allow a rich subset of one to one and simultaneous connection of microprocessors to memories, while reducing the hardware cost to O(NlogN) in contrast to O(NxN) for crossbar networks (Fig.1.3).

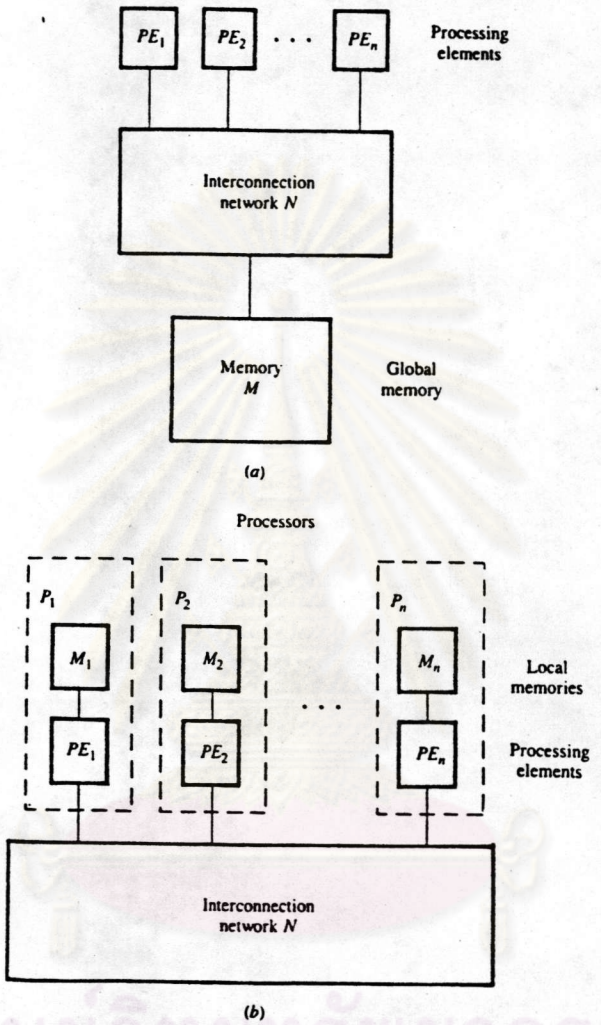

Fig 1.1 Comparison between shared-memory multiple-microprocessor system and loosely coupled multiple-microprocessor system. (a) shared-memory system (b) loosely coupled system

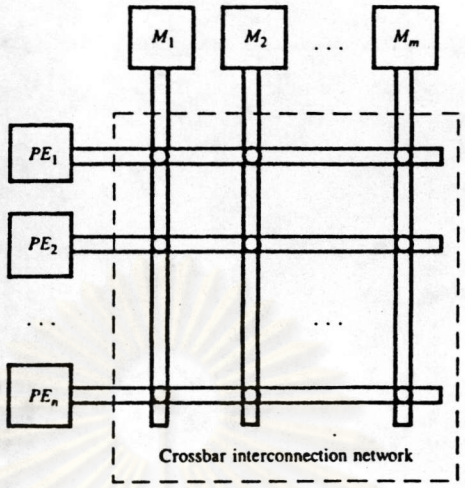

Fig 1.2 Crossbar interconnection network.

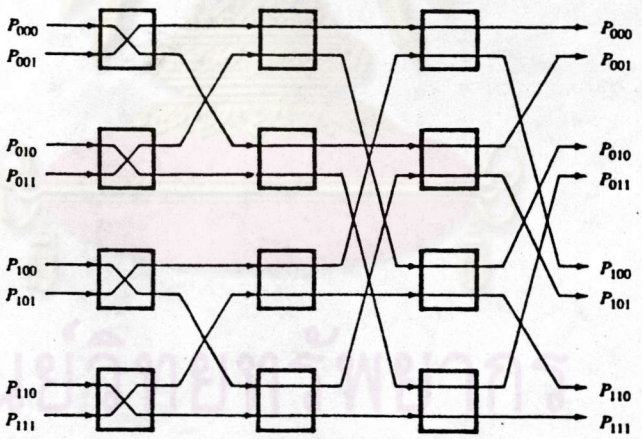

Fig 1.3 Multistage interconnection network.

A serious drawback of the multistage network is that there is only one path from an input to an output. It is necessary to incorporate some fault-tolerance into these networks so that at least a single fault in a switch or a link could be tolerated. Only a few commercial systems based on multistage interconnection networks have been produced, such as the BBN Butterfly [8].

1.2.3 Common Bus Interconnection Network.

A common bus interconnection network, shown in Fig. 1.4, is the least complex and most popular among manufacturers. This interconnection scheme is well known as being inexpensive and easy to implement. The shared-memory multiple-microprocessor system emerging as commercial products have avoided crossbar and multistage networks as a means of interconnecting the microprocessors to the shared memories. Instead they employ interconnection architectures based on conventional buses. The main characteristics of some commercial multiple-microprocessor systems are reviewed in Fig. 1.5 . The most attractive kind of interconnection network is the common bus.

1.3 Shared-Memory Multiple-Microprocessor System.

In real-time control applications where diversified computations are required and memory allocation is static, a shared-memory multiple- microprocessor system is one of the most suitable choices [1,2]. In such a system, each microprocessor assigned with a specific task usually communicates with its local memory and, occasionally, with a common shared memory for data exchange between the microprocessors in the system.

Shared-memory multiple-microprocessor systems have been suceeded mainly because they provide high-performance ratio for a broad range of applications. Several new shared-memory

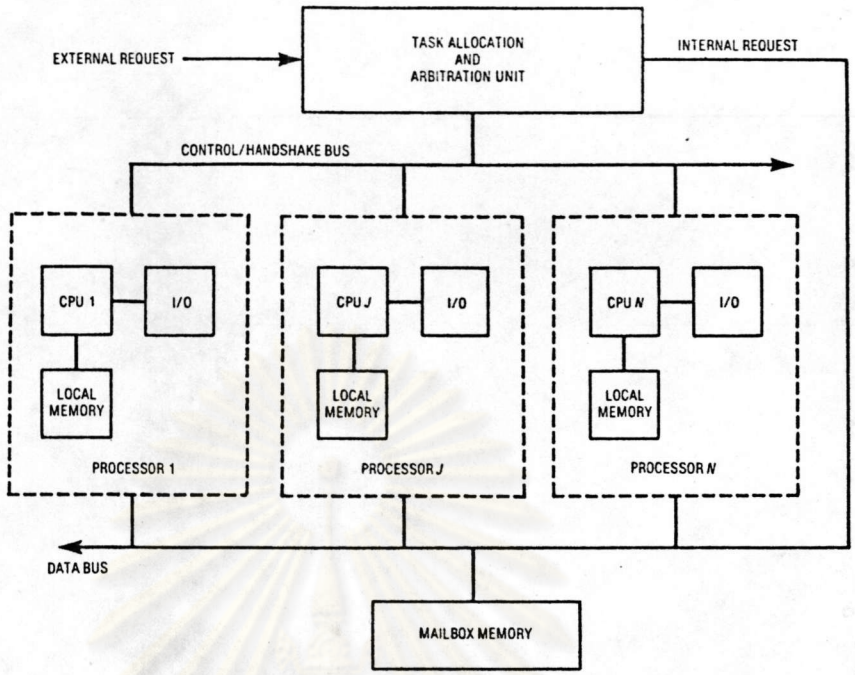

Fig 1.4 Common bus interconnection network.

System	Number of Processors	Shared Memory	Interconnection Network	Cache Memory	Cache Consistency
C.mmp	16	Global	Crossbar switch	Yes	Software tagging
CM*	50	Local	Hierarchical	No	—
RP3	512	Local	Multistage network	Yes	Software tagging
Alliant	20	Global	Common bus	Yes	Write-once
Cedar	64	Hierarchical	Multistage network	Yes	Software tagging
Butterfly	256	Local	Multistage network	No	—
SPUR	12	Global	Common bus	Yes	Ownership
Dragon	10	Global	Common bus	Yes	Write-through with update
Multimax	22	Global	Common bus	Yes	Write-through with invalidation
Balance 21000	30	Global	Common bus	Yes	Write-through with invalidation

Fig 1.5 Main characteristics of some multiple-microprocessor systems [8].

multiple-microprocessor systems have been announced every year. Representative examples are the Encore Multimax and Sequence Balance.

When shared memory is used as a data exchange medium and two or more microprocessors simultaneously require the use of shared memory, conflicts usually occur. An arbiter is normally required to solve the conflicts. Polczynski [9] proposed the FETCH-EXECUTE phase different method, but this can be used only with a specific microprocessor. Petriu [10] used an inherent property of logic gates to implement an arbiter, however, this cannot work with sophisticated priority rules and is difficult to implement. Loewer [11] described a system that used opposite phases of the clock for each microprocessor, but this limited the system to only 2 microprocessors. Hojberg [12] proposed an arbiter with a REQUEST-GRANT memory and a separate priority generator, this arbiter is relatively complicated and required two external high-speed clocks. In this dissertation, a flexible arbiter using a simplified and improved Hojberg method that requires very simple hardware and no external clock is presented. Two examples of multiple-microprocessor system with shared memory are also presented.

1.4 Multiple-Microprocessor System Performance Analysis Methods.

One intuitively expects that more computing power can be obtained by increasing the number of microprocessors in a multiple-microprocessor system. However, the entire system may perform less satisfactorily since more microprocessors can cause more conflicts for the common resources. A thorough performance analysis is needed before the advantages offered by multiple-microprocessor system can be fully utilized.

There are three general methods that are traditionally used to analyze the performance of multiple-microprocessor system :

simulation, benchmarking, and analytical modeling [13].

1.4.1 Simulation.

Simulation has been very popular form of computer modeling for years. To simulate the system means we used a program which generates the bahavior of a specified model. It enables the analyst to model the system at a much greater level of detail than is practicable with analytical model. However, simulation models are difficult and costly to construct, validate, and run. Simulation projects tend to become large, long-term, and very difficult to control.

1.4.2 Benchmarking.

The benchmarking approach to performance evaluation is probably the oldest and most widely used technique, although its use has been limited primarily for new hardware selection. A computer benchmark is definded as a program that, through a series of computations, operations, and actions, produces a relative figure of merit of a system's performance. A benchmark program is typically written in a high level language, and the frequency and type of statements that appear reflect the characteristics of the intended application. Thus it is not a promising technique for designing new system. There are many well known benchmark program for single processor system e.g. Wheatstone, Linpack. However, there has been little effort in the area of developing benchmark program for multiple-microprocessor system.

1.4.3 Analytical Modeling.

Analytical modeling involves the mathematical description of a system's operation and the subsequent analysis. It has become widely accepted as being a cost effective evaluation technique for estimating

the performance of computer systems. Analytical models are cost effective because they are based on efficient solutions to mathematical equations. However, in order for these equations to have a tractable solution, certain simplifying assumptions must be made. It is generally believed that carefully constructed analytical models can provide estimates of average throughput to within 10 % accuracy and estimates of average response time to within 30 % accuracy. There are three areas where this level of accuracy is usually considered to be sufficient and where analytical models have had substantial impact, namely capacity planning, I/O subsystem performance evaluation, and as a preliminary design aid in development of new systems.

Computer systems can generally be characterized as consisting of a set of both hardware and software resources and a set of tasks competing for an accessing those resources. It is therefore natural to represent the system by a network of interconnected queues. The purpose of the model is to predict the performance of the system by estimating characteristics of the resource utilizations, the queue lengths, and the queueing delays. Analytic performance models are queueing network models for which these characteristics may be found mathematically. Therefore, research in performance modeling methodology has essentially been research in queueing theory. Queueing theory has attained new relevance because of the computer performance modeling application. Furthermore, to a great extent, the direction of queueing theory has been influenced and driven by this application.

1.5 Queueing Model for Multiple-Microprocessor Systems.

There are two basic modeling approaches for analyzing the performance of the multiple-microprocessor systems. The first approach is based on the problem of hardware contention for buses and

memory. This approach usually models the multiple-microprocessor system with the closed queueing network [14]. The performance of the multiple-microprocessor system according to this approach has been studied by several authors[15,16,17,18]. This modeling approach focuses on performance analysis of the large-scale general purpose multiple-microprocessor systems. The second approach is based on the flow equivalence technique [17]. The analysis in this approach treats each microprocessor as a server which takes its input from a given queue. This approach has been used by many authors for predicting the performance, in terms of system response time, of the multiple-microprocessor systems for specific applications where the microprocessor tasks are not dynamically reconfigurable [19,20,21]. However, most of the papers published using this approach focused on the symmetrical multiple-microprocessor system i.e. the system that consists of identical microprocessors. The unsymmetrical multiple-microprocessor system; or the system that consists of unidentical microprocessors or microprocessors with different processing rate, is usually required in industrial control applications. In this dissertation the performance analysis is extended to study the multiple-microprocessor systems that consist of microprocessors with different processing rate or the unsymmetrical multiple-microprocessor systems.

1.6 Objectives of the Dissertation.

To gain a better understanding of the multiple-microprocessor system for real-time applications, simple technique for implementing the shared-memory multiple-microprocessor systems will be presented first. The proposed shared-memory multiple-microprocessor system is then applied to two real-time application examples. By using queueing theory, a simplified performance model of a shared-memory multiple-microprocessor system with unidentical microprocessors is presented. The proposed model is based on the flow equivalence

technique [17]. Two systems are discussed : the system with equal processor priority and unequal priority. The model is verified with the experiment on the proposed shared-memory multiple-microprocessor system. The model is then applied to analyze the performance of two multiple-microprocessor systems for real-time applications.

In summary, the objectives of this dissertation are as follows:

1. To present an efficient technique for implementing the shared-memory multiple-microprocessor systems.

2. To develop a queueing model for the shared-memory multiple-microprocessor systems that consist of microprocessors with unidentical processing capability.

3. To demonstrate the applications of the proposed model for performance analysis of a multiple-microprocessor based communication communication controller and the SCADA systems.

The proposed analysis method can be used in the design stage of a multiple-microprocessor system. It can be helpful in selecting a suitable system for a given application. The performance of multiple-microprocessor system can be achieved prior to hardware implementation without the use of a costly simulation program.