BUS COMMUNICATIONS SYSTEMS

Robert Chia-Hua Chen

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pa. 15213
January, 1974

Submitted to Carnegie-Mellon University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

This work was supported by the National Science Foundation under grant GJ 32758X.

This is a summary of a thesis by the same title, available from the National Technical Information Service (NTIS) of the Department of Commerce under NSF-OCA-GJ32758X-RT0174.

Bus Communications Systems

Robert Chia-Hua Chen

January, 1974

Thesis Summary

This thesis addresses problems associated with the design of communications systems which have buses as their basic structure. Three major areas of study are treated. These are the problem of synchronizing the sender and receiver for each transmission of data on a single bus (bus synchronization), the problem of arbitrating requests for use of the bus by different units on the same bus (bus arbitration), and the problem of deadlocks in systems with multiple buses.

For many years, the most valuable portion of a computer system was the central processor. The design of the processor was therefore critically important to the cost and performance of the entire system. Much work went into processor design, and many schemes -- multiprogramming, for instance -- were devised to obtain better utilization of the processor. As time went by and the cost of the processor decreased more quickly than the cost of the memory, and as the need for bigger and faster memories

grew more quickly than the need for faster processors, the memory component of computer systems gained in relative importance. Interest shifted to memory system design, and ways to achieve better utilization of valuable memory components. Memory hierarchies received a great deal of attention; base-displacement addressing was introduced partly to increase address space without increasing instruction length; the cache was introduced to utilize limited high speed main memory.

As processor and memory costs decrease further, the cost of a third major component -- the switch -- takes on increasing significance. As processor and memory speeds increase further, the delays imposed by switches and communication links also become increasingly significant. At the same time, as computing needs continue a growth rate which is not matched by the improvement rate of processor and memory speed, and as decreasing computing cost further broadens the spectrum of applications, frequently increasing the desirability of decentralization, more and more of the switching and communications function is needed. As a result, the communications system -- the system of links and switches -- has become more and more important a consideration at the intra-system level of computer systems.

At the inter-system level, many computer networks have been constructed in recent years, spurred by practical considerations of resource sharing and load sharing. The quick growth of computer networks and the promise of much more future growth has also emphasized the need for

studying communications systems. This thesis studies one type of communications systems: communications systems based on the communications bus.

BUS SYNCHRONIZATION

Communication between units connected by a bus take place via messages sent from a sending unit to a receiving unit. Each message consists of a number of data items, which are sent serially; the sending of each data item involves the driving of data lines by the sender and the recording of the state of these lines by the receiver. Of critical importance is the timing of the transaction: the receiver will obtain incorrect information unless it is recording the state of the lines at the time when the lines are being driven by the sender. Successful transmission of data is therefore dependent on synchronization between sender and receiver.

Synchronization is a characteristic requirement of any set of processes that run asynchronously and interact occasionally; therefore, the problem of synchronization appears at many different levels in computing systems. Much work has been done on primitives for synchronization at the programming or software process level. Several authors have addressed the problem of synchronization at a different level: the level of line control procedures. At the root of all synchronization problems is the control of the progress of processes which interact. What differentiates one type of

synchronization problem from another is the set of assumptions made concerning these processes: the primitives which can be used to control them and the ways in which control must be exercised. At the software level, the primitives usually involve the testing and setting of various memory locations which are accessed by the processes and thus control their progress; the ways in which control must be exercised usually involve the serialization of certain types of actions by different processes which otherwise might proceed concurrently. At the level of line control procedures, the primitives usually involve binary inputs to the processes, which are viewed as finite state machines, and the control which must be exercised usually involve the determination of the state of the interacting processes in order that they will exhibit the proper sequence of actions. At both of these levels, the required communications facilities -- for testing and setting memory locations at the one level, and for presenting sets of inputs at the other -- are implicitly assumed.

The level treated in this thesis is that of a single data transmission. The control that is exercised at this level is exercised by the driving and receiving of voltage or current levels on synchronization lines, and controls the recording of voltage or current levels on data lines as data transmitted from the sending unit to the receiving unit. The primitives for synchronizing data transmissions are identified: they are the SR (sender ready), RR (receiver ready), DR (data ready) and DA (data accept) synchronization primitives, and acknowledgments to these. Different

synchronization protocols are constructed by using different combinations of these synchronization primitives and acknowledgments. The pulse, level and transition signalling modes are each considered. The maximum rate of data transmission, in terms of several parameters governed by the physical characteristics of the system, is determined for the different implementations of each synchronization protocol. Two possible sources of error are described -- errors due to levels which are maintained for too short a time, and errors due to levels which are maintained for too long. A method is given for identification of portions of a synchronization protocol which are susceptible to such errors, and possible corrective measures are described. In this way, a complete set of alternatives is described and analyzed so that an intelligent choice may be made to satisfy a given speed or error rate requirement. The identification of synchronization primitives and the systematic discussion of all the synchronization protocols which can be constructed with these primitives is carried out for the first time in this thesis.

BUS ARBITRATION

A communication bus serves a number of units, any number of which may be senders or receivers of messages. At any one time, any number of potential message senders may require the bus for the sending of a message. On the other hand, the bus can only carry one message at a time: in other words, the bus may at any one time be actually allocated to

only one of the message senders requiring it. It is necessary, therefore, to institute a mechanism to perform this allocation. The allocation of the bus is referred to as bus arbitration, and the mechanism for bus arbitration is referred to as the bus arbiter.

Bus arbitration is a matter of assigning a single resource to one of a number of requesters, a subject that has been addressed by the theory of scheduling and studies on scheduling algorithms. Much of this related work has come about as studies of the allocation of the processor in single processor multiprogramming computing systems. Bus arbitration differs from most of the scheduling situations studied (such as multiprogramming) in that the cost of communicating requests and other information required for arbitration, in terms of both time delay and physical material costs, is more critical because of the low costs involved in the use of the bus itself. Because of these reasons, it is important to examine the costs of various implementations of various arbitration algorithms.

While no complete set of alternatives can be identified for the bus arbitration problem as can be identified for the bus synchronization problem, this thesis presents a methodology, based on identifying the tradeoffs possible by moving through the design space defined by dimensions of variability of the implementation method, which may be used to guide the search for a bus arbiter suited to the requirements of a particular implementation. The input information required by the arbitration algorithm must be communicated to the bus arbiter unless it is generated at the

arbiter location, and therefore may impose a communication cost (in terms of both equipment and delay). To reduce such costs, the arbiter may be implemented as a dynamic arbiter, located at different units at different times, or as a distributed arbiter, distributing the arbitration algorithm in such a way that the communication costs are minimized. Additionally, the priority sorting method may be chosen to optimize the communications cost or the time required for the arbitration process, the addressing scheme may be chosen to reduce the task of the arbitration algorithm or to eliminate the need for resolving ambiguities, and the information passing method may be chosen to reduce the communication lines needed or to accomodate simultaneous transmission by different units or reduce hardware requirements at certain units. Several arbitration algorithms are implemented using each of five different implementation methods to illustrate these different considerations.

DEADLOCKS

In the past, deadlocks have not been a major issue in the design and analysis of computer communications systems, because the occurrence of a deadlock was always a very rare if not an impossible happening. This was due to the nature of the data traffic in the computer systems. Almost exclusively, data traffic was generated and controlled by processors, of which there were usually not more than a few to a computer system. Also, data flow was usually between a processor and some other device

which was not a processor. Therefore it was practical to construct a separate bus for each processor where high speed was needed and dense traffic was expected (e.g., the processor to main memory links) or to construct a single shared bus for all processors where speed was not as critical and traffic was lighter (e.g., the processor to peripherals links). Such systems were inherently deadlock free, because the resources were requested by the processors always in the same order, as in a resource ordering scheme, and in fact the processors usually requested either completely different or exactly the same resources.

With the appearance of systems involving more processors, and especially with more complex systems, the possibility of deadlocks increases to very significant levels. A simple case where deadlocks can be very significant is that of the PDP-11 Unibus window [Digital Equipment Corp., 1973]. Deadlocks can also occur in the ARPA network [Frank et al, 1972], and in the new minicomputer/microprocessor for the ARPA network [Heart et al, 1973], where it is reduced in likelihood only because the system is special purpose and has been programmed in such a way as to achieve the reduction.

This thesis considers three major ways of dealing with deadlocks: deadlock prevention, deadlock avoidance, and deadlock detection and recovery. Both message (packet) switched and circuit (line) switched systems are treated. Deadlock prevention aims at the design of communications systems which, through the use of message routing rules,

can be certified to be deadlock free regardless of the number of messages in the system. This type of deadlock prevention has not been treated in previous work. Such deadlock prevention methods may require the inclusion of communications resources which are not efficiently utilized for the sake of eliminating the possibility of deadlocks, and may allow less flexibility in the use of resources, but possess the advantage that no overhead is imposed during the operation of the communications system -- any allocation of a resource to a message may be made under any circumstances, without incurring the danger of a deadlock. The message routing rules have been expressed as a number of message types, where each message type, when occupying certain resources, will only request one of a known set of "successor" resources. Two types of routing rules are considered: fixed routing, where the set of successor resources is always either a singleton set or the null set, and alternate routing, where the sets of successors may have more than one element. In each case, a number of algorithms are given, of differing permissiveness and also differing computational complexity, each of which will reject all systems which are not deadlock free, but only the most permissive of which will accept all systems which are deadlock free. Examples are given of both an existing message switched system (the ARPA network) and circuit switched systems, in which deadlocks can occur; the results of the thesis are applied to these examples to determine routing rules or system configurations which render the resulting systems deadlock free. Alternate routing represents an extension of previous work on deadlocks in that a set of alternate resource types are given for each

request for resources, and resources of any one of these types can satisfy the request.

Also considered are deadlock algorithms used during the operation of the communication system to avoid deadlocks or to detect and recover from deadlocks. Two types of deadlock avoidance algorithms are used. Global deadlock avoidance algorithms, where the algorithm is aware of the total current state of resource allocation, are special case applications of previous work on deadlocks. Local deadlock algorithms, where the algorithm is not aware of the total current state of resource allocation, are also considered: for two classes of local deadlock algorithms, methods similar to the deadlock prevention algorithms are given for determining which communications systems will be deadlock free under the application of a particular local deadlock algorithm of that class. Deadlock detection and recovery algorithms are briefly discussed. Global deadlock detection algorithms are also special case applications of previous work on deadlocks. One type of widely used local deadlock detection -- the time-out -- is discussed.

[Digital Equipment Corporation, 1973] "PDP-11 peripherals handbook". Digital Equipment Corporation, Maynard, Mass., Sept. 1973

[Frank et al, 1972] Frank, H., Kahn, R. E. and Kleinrock, L. "Computer communication network design -- experience with theory and practice".

Proc. AFIPS Conf. Vol. 40 (SJCC 1972), pp 255-270.

[Heart et al, 1973] Heart, F. E., Ornstein, S. M., Crowther, W. R., Barker, W. B. "A new minicomputer/multiprocessor for the ARPA network". Proc. AFIPS Conf. Vol. 42 (NCC 1973), pp 529-537.

BIBLIOGRAPHIC DATA SHEET	1. Report No. NSF-OCA-GJ32758X-RT0174	2.	3. Recipient's Accession No.

4. Title and Subtitle
BUS COMMUNICATIONS SYSTEMS

5. Report Date
January, 1974

6.

7. Author(s)
Robert Chia-Hua Chen

8. Performing Organization Rept. No.

9. Performing Organization Name and Address
Carnegie-Mellon University; Department of Computer Science
Pittsburgh, Pennsylvania 15213

10. Project/Task/Work Unit No.

11. Contract/Grant No.
GJ32758X

12. Sponsoring Organization Name and Address
John R. Lehman
Program Director, Computer Systems Design
National Science Foundation
Washington, D. C. 20550

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstracts This thesis addresses problems associated with the design of communications systems which have buses as their basic structure. Three major areas of study are treated. These are the problem of synchronizing the sender and receiver for each transmission of data on a single bus (bus synchronization), the problem of arbitrating requests for use of the bus by different units on the same bus (bus arbitration), and problem of deadlocks in systems with multiple buses.

17. Key Words and Document Analysis. 17a. Descriptors bus synchronization, bus arbitration, multiple buses

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for public release; distribution unlimited	UNCLASSIFIED	14
	20. Security Class (This Page) UNCLASSIFIED	22. Price

INSTRUCTIONS FOR COMPLETING FORM NTIS-35 (10-70) (Bibliographic Data Sheet based on COSATI Guidelines to Format Standards for Scientific and Technical Reports Prepared by or for the Federal Government, PB-180 600).

1. **Report Number.** Each individually bound report shall carry a unique alphanumeric designation selected by the performing organization or provided by the sponsoring organization. Use uppercase letters and Arabic numerals only. Examples FASEB-NS-87 and FAA-RD-68-09.

2. Leave blank.

3. **Recipient's Accession Number.** Reserved for use by each report recipient.

4. **Title and Subtitle.** Title should indicate clearly and briefly the subject coverage of the report, and be displayed prominently. Set subtitle, if used, in smaller type or otherwise subordinate it to main title. When a report is prepared in more than one volume, repeat the primary title, add volume number and include subtitle for the specific volume.

5. **Report Date.** Each report shall carry a date indicating at least month and year. Indicate the basis on which it was selected (e.g., date of issue, date of approval, date of preparation.

6. **Performing Organization Code.** Leave blank.

7. **Author(s).** Give name(s) in conventional order (e.g., John R. Doe, or J.Robert Doe). List author's affiliation if it differs from the performing organization.

8. **Performing Organization Report Number.** Insert if performing organization wishes to assign this number.

9. **Performing Organization Name and Address.** Give name, street, city, state, and zip code. List no more than two levels of an organizational hierarchy. Display the name of the organization exactly as it should appear in Government indexes such as **USGRDR-I.**

10. **Project/Task/Work Unit Number.** Use the project, task and work unit numbers under which the report was prepared.

11. **Contract/Grant Number.** Insert contract or grant number under which report was prepared.

12. **Sponsoring Agency Name and Address.** Include zip code.

13. **Type of Report and Period Covered.** Indicate interim, final, etc., and, if applicable, dates covered.

14. **Sponsoring Agency Code.** Leave blank.

15. **Supplementary Notes.** Enter information not included elsewhere but useful, such as: Prepared in cooperation with . . . Translation of . . . Presented at conference of . . . To be published in . . . Supersedes . . . Supplements . . .

16. **Abstract.** Include a brief (200 words or less) factual summary of the most significant information contained in the report. If the report contains a significant bibliography or literature survey, mention it here.

17. **Key Words and Document Analysis. (a). Descriptors.** Select from the Thesaurus of Engineering and Scientific Terms the proper authorized terms that identify the major concept of the research and are sufficiently specific and precise to be used as index entries for cataloging.
(b). Identifiers and Open-Ended Terms. Use identifiers for project names, code names, equipment designators, etc. Use open-ended terms written in descriptor form for those subjects for which no descriptor exists.
(c). COSATI Field/Group. Field and Group assignments are to be taken from the 1965 COSATI Subject Category List. Since the majority of documents are multidisciplinary in nature, the primary Field/Group assignment(s) will be the specific discipline, area of human endeavor, or type of physical object. The application(s) will be cross-referenced with secondary Field/Group assignments that will follow the primary posting(s).

18. **Distribution Statement.** Denote releasability to the public or limitation for reasons other than security for example "Release unlimited". Cite any availability to the public, with address and price.

19 & 20. **Security Classification.** Do not submit classified reports to the National Technical

21. **Number of Pages.** Insert the total number of pages, including this one and unnumbered pages, but excluding distribution list, if any.

22. **Price.** Insert the price set by the National Technical Information Service or the Government Printing Office, if known.