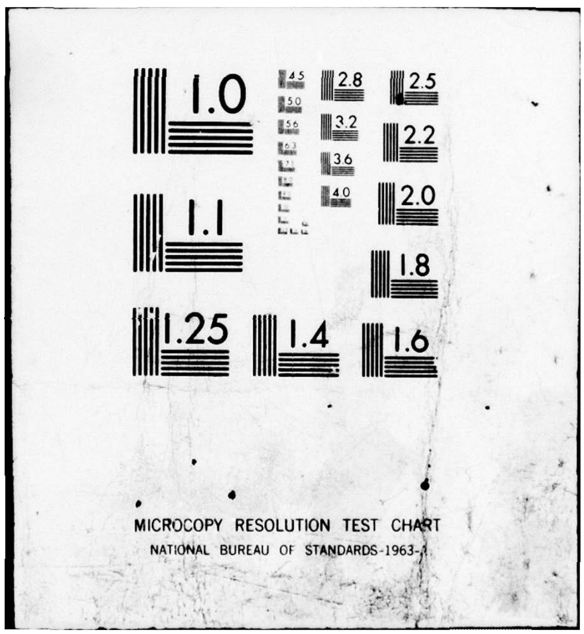
AD-A081 036 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
BASIC HARDWARE INTERCONNECTION MECHANISMS FOR BUILDING MULTIPLE--ETC(U)
NOV 79 B J CAREY, E E MCCOY
UNCLASSIFIED NPS52-79-005 NL

1 OF 1
AD
A081036

END
DATE
FILMED
3-80
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

LEVEL II

NPS52-79-005

NAVAL POSTGRADUATE SCHOOL

Monterey, California

DDC FILE COPY

BASIC HARDWARE INTERCONNECTION MECHANISMS

FOR

BUILDING MULTIPLE MICROCOMPUTER SYSTEMS

Bernard J. Carey

Earl E. McCoy

November 1979

Approved for public release; distribution unlimited.

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral T. F. Dedman
Superintendent

Jack R. Borsting
Provost

The work reported herein was supported in part by the Department of Electrical Engineering and Computer Science, University of Connecticut, Storrs, Connecticut and in part by the Department of Computer Science, Naval Postgraduate School, Monterey, California.

This report was prepared by:

Bernard J. Carey
BERNARD J. CAREY
Associate Professor of
Computer Science

Earl E. McCoy
EARL E. McCOY
Assistant Professor of
Computer Science

Reviewed by:

GORDON H. BRADLEY, Chairman
Dept. of Computer Science

Released by:

William M. Tolles
WILLIAM M. TOLLES,
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 NPS52-79-005	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Basic Hardware Interconnection Mechanisms for Building Multiple Microcomputer Systems.		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Bernard J. Carey Earl E. McCoy		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940	11	12. REPORT DATE Nov[redacted]79 13. NUMBER OF PAGES 22
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 37		15. SECURITY CLASS. (of this report) unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

9 Technical rept.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Multiplecomputers, Hardware Interconnection Mechanisms, Microcomputers, Computer Architecture Taxonomy, Digital Hardware Primitives

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents the current results of a research project which has been concerned with methods for designing and implementing multiple microcomputer systems. The design method is based upon identifying hardware interconnection primitives which may be used to construct the interconnection subsystem which characterizes a given multicomputer architecture. An actual experimental system has been constructed which will permit builing nine of the ten systems in the Anderson and Jensen architecture taxonomy.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

251450 Jw

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Basic Hardware Interconnection Mechanisms
for
Building Multiple Microcomputer Systems

Bernard J Carey
Earl E McCoy

Department of Computer Science
Naval Postgraduate School
Monterey, Calif 93940

and

Department of Electrical Engineering and Computer Science
University of Connecticut
Storrs, Ct. 06268

November 1979

Acknowledgement

The work reported in this document was performed in the department of Electrical Engineering and Computer Science of the University of Connecticut. This was a self directed research project. Financial support was provided by the University of Connecticut and the experimental systems described herein were constructed in the Microcomputer Lab in the Electrical Engineering and Computer Science department.

This technical report was written while the authors were visiting members of the faculty of the Computer Science department of the Naval Postgraduate School.

Discussion

This report presents a discussion of an approach to building and studying multiple microcomputer systems. The report initially discusses the general problem area of multiple computer systems and indicates why research and development studies need to be pursued in this area. This report presents the current results of a project which has been studying one aspect of this large problem. This particular project has limited itself to the study of local, special purpose, multiple, microcomputer architectures. The particular approach taken has been to use a generalized architecture taxonomy presented by Anderson and Jensen [Ander76] to define the types of architectures of interest. There are ten architectures in this taxonomy. Each architecture has distinguishing attributes which will characterize the functional and operational behavior of the system. The taxonomy only identifies high level attributes. Problems associated with real time performance, appropriate methods to program these architectures, reliability issues, operating systems questions, etc. still remain to be solved. It is felt that the existence of an experimental system whereby any of the (functionally representative) Anderson and Jensen architecture types might be conveniently built would provide a powerful tool to study some of these problems. This report presents the current results of the project which has been concerned with designing and building such an experimental microcomputer system. The philosophy followed was to base all systems on general pupose hardware primitives from which the various systems can be built. It is critical to identify these general concepts so that the results of the experimental work can be extended to the general case where different hardware and software implementations occur and to permit considering the newer VLSI devices which are becoming commercially available but which are not included in the present experimental system.

Present State of Knowledge

Advances in VLSI technology have been responsible for the propagation of digital techniques for implementing solutions to engineering problems which traditionally have been solved by other methods. These advances have also made it possible to consider building multicomputer systems where until quite recently digital systems utilizing a single computer (i.e. one processor in the system) were the norm. These two trends are converging so as to cause a growth of activity in considering the application of systems of multiple computers in a variety of disciplines.

Unfortunately, although it now seems possible to build a multiple computer system, it is not clear how to build the best one in terms of matching the design of the computer system to the needs of the application problem. Traditionally the hardware resource was a fixed element and the software structure was the major variable to consider. Now the form of the hardware resource is also a major variable. In fact it is not even clear what types of multicomputers may be implemented let alone making a rational choice among well understood alternatives. There are a variety of single computers with different attributes. There are a variety of methods and mechanisms for interconnecting these single computers into multiple computer systems. How does the designer choose the "best" permutation of these possibilities as a solution to a particular application problem if the permutations themselves are not well understood? It is felt that these permutations may be described in a systematic manner if the possible architectures are characterized by the attributes of the individual computers comprising the nodes of the multi computer system, the attributes of the hardware and software mechanisms used to interconnect the individual nodes, and the attributes of the total structure making up the global interconnection subsystem. With this understanding of the behavior and attributes of the possible multicomputer architectures available as an information base, it would then be feasible to progress to the next step of choosing the best architecture for a given problem. With information available which defines multicomputer architecture types, methods may be developed to guide the generation of operation systems and application programs which can then be matched to the underlying hardware structure. Any design process involves making choices among alternatives. Information describing alternative multicomputer structures is required so that those choices can be made in a systematic fashion.

The potential scope of the multiple computer architecture space is large enough that a study of an important subset of the total space will provide an important contribution. An area of current interest is concerned with special purpose architectures which are based upon microprocessors and compatible VLSI devices. For the intents of this research project a "special purpose" multiple computer architecture is one whose characteristics are fixed at design time while a "general purpose" architecture is one which may be dynamically reconfigured at run time according to the needs of the task(s) it is supporting. The subject of reliability is of sufficient importance that it will be considered as a normal, desirable attribute of special purpose systems. If a hardware module failure occurs reconfiguration of the system will be necessary for the purpose of maintaining the system characteristics which were specified at design time. Thus at least this aspect of dynamic reconfiguration is of interest in special purpose systems. The special purpose systems will encompass the spectrum from tightly coupled multiprocessors to loosely coupled systems.

From a high level architecture viewpoint, a multiple computer system may be described as a number of independent processing nodes which are connected together by an Interconnection Subsystem (ISS). Microcomputer based processing nodes may be characterized as conventional general purpose digital computers which execute sequential programs. The attributes of these nodes may be described by existing notations such as the PMS and ISP descriptive systems [Bell71,Bell78]. What is not presently understood is how to effectively describe any arbitrary multiple computer architecture. Even given the case where the same processing nodes are used, a variety of different multicomputer architectures are possible whose characteristics are dependent upon the structure of the ISS which is employed. An example of this is shown in figure 1. Further permutations are possible when it is considered that for a particular functional ISS type there are a variety of implementations which are dependent upon the particular hardware mechanisms, and the associated standards and protocols which are employed. An example of this is shown in figure 2. One must know the implementation details for the ISS before the operational behavior of the multi-computer system can be definitively described.

The philosophy of the PMS approach to describing computer systems is based upon the idea that a fundamental set of functional building blocks can be identified from which all digital systems are comprised. Evidence indicates that this is valid for the uni-computer case and studies such as that which developed the Cm* architecture [Swan77] indicates that the approach is also valid for the multi-computer case. Each element in the PMS set is a functional primitive. Corresponding hardware primitives may be generated reflecting implementation considerations. This philosophy can apply to multiple computer architectures. A set of hardware interconnection primitives (HIP), along with recognized standards and protocols, can be identified. These HIPs will form a primitve set from which any ISS, within the context of implementations based upon VLSI, are realized.

With a complete set of hardware building blocks known for the multi-computer case, a need exists for an underlying structure upon which to found the study of multiple computer architectures. Thus a rationale exists for deriving a taxonomy for multiple computer systems. At present there are a small number of existing taxonomies [Ander76] [Free79] [Kro72] [Fly72] [Cren72]. Of these taxonomies the one presented by Anderson and Jensen appears to be a conceptually sound starting point for identifying HIPs and ISSs. The nodes in the tree identify high level architecture types and provide an initial classification of ISSs. From this information a functionally complete set of ISSs and associated HIPs may be defined.

It appears that a variety of multiple computer configurations are both technically and economically feasible. However, detailed information needs to be developed which defines the total number of distinguishable architectures, and their operational characteristics before design decisions can be made in a systematic manner.

Related Research

Distributed system taxonomies have been devised by other researchers such as Freeman and Thurber [Free79]. Their work categorizes local computer networks into four classes: packet switched, circuit switched, bus structure, and I/O channels. Anderson and Jensen [Ander76] provide a more detailed taxonomy with ten architecture classifications. The architecture taxomony is based upon functional attributes of the processing elements (the nodes), paths or links, messages, and switches. Specifically not considered are communication strategies, message addressing, or deadlock situations. The primary shortcoming of this taxonomy is that some actual implementations of distributed systems are hybrids of the basic taxonomy classes. In addition the communications protocol remains as an important unconsidered attribute. Thus one attribute of the ISS which can greatly affect system performance is not included in the taxonomy. It may be argued that this is an implementation consideration rather than a general system attribute. A graphical summary of the Anderson and Jensen taxonomy is presented in figure 3.

Several research projects are underway on local computer networks. Of particular interest are the definitions of various hardware interconnection primitives (HIPs) and interconnection subsystems (ISSs), although not referred to in those terms. Rennels [Renn78] describes a fault-tolerant distributed system intended for spacecraft. A number of modular building blocks are defined specifically for the interconnection of off-the-self computer components. These are a memory interface (MI-BB), a bus interface (BI-BB), one for input-output (IO-BB), and a so-called "core module" (CORE-BB). Collectively these building blocks constitute a class of ISSs, at least for the particular architecture investigated by Rennels. His redundant global bus approach may be more fault-tolerant than is required by most earth bound networks but self-checking components are an important area of interest.

Powell et. al. [Pow78] describe a hybrid architecture with interesting ISSs. Their network consists of two levels; the higher level is an irregular network whose nodes are interfaced by an ISS composed of one or more four port HIPs that collectively make available an arbitrary number of ports to the network. Only one of the ports may be active at any moment in time, with contention control determining the particular port. The lower level consists of the node itself which implements a unique form of a bus: a loosely coupled pulse transformer. The advantage claimed for such a HIP is its fault-tolerant properties.

An example of a ring topology is the Distributed Computer System developed at UC Irvine [Loomis73]. A control token is passed around the ring for control purposes. Of special interest is their definition of an interconnection primitive called a "local network interface" or LNI. This LNI is intended to also work in networks using contention ring control and in contention bus networks as well. Their LNI is composed of two HIPs; a network interface HIP and a host-specific HIP. The first HIP controls data transmission and reception, recognizes addresses, and conditions signals. The host-specific HIP is designed to exchange data between the network and the host computer. A HIP for DMA transfer to a LSI-11 UNIBUS has been built.

An interesting variation on the IDDR architecture of Andersen and Jensen is the MICRONET system [Wit78]. In this system each node is connected to a horizontal and to a vertical bus through a HIP defined for this system. Each node consists of a local computer (a C) and a HIP which implements the bus connections and which interacts with the local C. In a given configuration the local C can be eliminated and a given node may only consist of the special HIP which maintains the ISS at a given bus junction. An interesting facet of MICRONET is that it resembles the IDDR architecture but it has an important difference in that any MICRONET node can communicate with any other node connected to a bus while in the IDDR case a given node can only communicate with a neighbor in the array. This difference affects the switching complexity of the system, its reliability, and other system attributes. Such differences must be resolved in a generally complete taxonomy.

As mentioned earlier a communications protocol is an important aspect of a distributed system because of its infuence upon the system's performance. At the "common carrier" network level much standardization is underway by various international and national institutions. A local computer network may benefit by using standard components. Standards are being discussed at seven different levels, many or most of which may be important to the context considered here. It is hoped that someday both hardware and software modules may be commercially available at relatively low cost (compared to the cost of producing "optimal" modules for a particular application problem). A recent technical journal is devoted to network protocols (September 1979 Computer, Special Issue on Network Protocols).

Current Results from Project : Modular Microcomputers

An example of a modular microcomputer system is the Microprocessor Independent Microcomputer (MIMIC) [Carey77a, Var79]. This is based upon a standardized microprocessor bus, and a set of modular hardware primitives which are compatible with any of the current generation of 8 bit microprocessors. These systems are being extended to the multicomputer case by defining hardware primitives from which a variety of systems may be realized. A first set of hardware interconnection primitives (HIP) have been designed and implemented which include (1) a shared memory, (2) multichannel parallel and serial links, (3) a multiport bus window, (5) an intelligent channel processor, and (5) extensions of the MIMIC bus standard to support shared bus systems. These HIPs are being installed in a three node MIMIC system so as to perform experiments on a variety of interconnection subsystems (ISS). The selection of the particular hardware interconnection primitives was done by an analysis of a high level taxonomy [Ander76]. The designs were a function of the characteristics of the MIMIC bus and currently available VLSI devices. It is felt that the experience gained with such a system will provide valuable insights into understanding more complex systems based upon future VLSI devices and more complex processing nodes and interconnections subsystems.

Where the MIMIC multicomputer system is based upon internal University of Connecticut designs and 8 bit microprocessors, a second multiple computer based upon commercial modules, and the 16 bit LSI 11 microprocessor, has also been designed and implemented. The interconnection subsystem (ISS) is entirely based upon serial links.

Although this is a very low performance environment the particular ISS is very flexible, permits a variety of architectures to be implemented under program control, and is currently being used to explore problems with distributed software systems.

Current Results : Hardware Interconnection Techniques and Primitives

There are a limited number of ways that digital hardware modules and systems may be interconnected. This limitation is based upon the nature of digital hardware itself. Additional limitations occur if only certain types of digital systems are considered such as those which are based upon microprocessors and associated VLSI devices. In addition to hardware limits functional aspects must be considered. In the PMS system [Bell71] there are two interconnection primitives; the link (L) and the switch (S). This is a functional definition and in practice other primitives such as the memory (M), the controller (K), etc. may serve to implement an interconnection function. To say that something is an interconnection mechanism requires first stating the level of viewpoint which is being used. For this paper at least three viewpoint levels are employed. These levels are the (1) electronic hardware, (2) system functional, and (3) software functional.

At the electronic hardware level in microcomputer based systems only four primary mechanisms exist for interconnecting modules. These are (1) bit serial transmission via a serial link under the control of a processor, (2)word parallel transmission via a parallel link under the control of a processor, (3) word parallel transmission over a bus (i.e. the internal microcomputer bus), and (4) word parallel transmission through storage in a shared memory (I/O devices may be viewed as a shared memory). These are the basic possibilities with hardware mechansims of higher functionality being constructed from this set of techniques.

At the system functional level the PMS concepts appear to be complete. Therefore a given approach will be characterized as a PMS primitive. An interconnection mechanism may have a completely programmable computer within it but if the function it is performing is that of a switch then a switch it is.

At the software functional level the activities that the hardware system is supporting must be considered. McFayden [Mcfay76] suggests such a functional viewpoint by proposing the "functional layering" concept for organizing software systems in multicomputer systems. In this approach the functional responsibilities of the network nodes are divided into three major layers which are (1) transmission management, (2) node functional task management, and (3) application operations. This identifies at least three node tasks which will have a time ordered relationship with resulting implications about the potential parallelism (concurrency) of software operations associated with the node. The performance of the node can be immediately affected by the hardware resources in the node which are available to support (exploit) this potential parallelism.

A further classification of the interconnection mechanisms can be made if the system is divided into the two broad classes of internodal and intranodal connections with internodal being those connections between disjoint nodes and intranodal being those connections within a distinct node in the system. At first it might be tempting to envision that fundamentally different hardware mechanisms are necessary to realize internodal versus intranodal connections. However this is not true. As stated there are a severely limited number of basic digital hardware possibilities for implementing hardware connections and these cover both the inter and intra nodal cases. The same hardware modules may be employed to realize both. The distiction becomes a matter of functional defintion as is true with the PMS primitives.

With the above discussion as an explanation let the following hardware interconnection primitives be defined for microcomputer based systems.

General microcomputer hardware interconnection primitives:

1. Microcomputer bus (MB) : primary intranodal connection between the microprocessor and all other functional modules in the node.
2. Microcomputer bus window (MBW) : electronic mapping between MBs.
3. Shared Memory (SM) : a memory which may be accessed by more than one microcomputer or microprocessor.
4. Shared Microcomputer Bus (SMB) : an internal bus which may be dynamically time shared by two or more microprocessors.
5. Shared Communication Bus := (SCB) := an external bus which is shared by two or more microcomputers for internodal communication
6. Serial Link (Ls) : a bit serial interface whose operations are controlled by a microprocessor. Normally only used for transmission of information.
7. Parallel Link (Lp) : a word parallel interface whose operations are controlled by a microprocessor. May be used for simple message transmission or as part of a control mechanism.
8. Transmission Processor (Pt) : a microprocessor whose purpose is to

support the transmission layer within a node or between nodes. A functional division into two separate types of Pt may be necessary if internodal and intranodal transmission operations prove to be fundamentally different.

9. DMA controller (Kdma) : implement dma intranodal transfers.

There are other possibilities such as an alignment network but upon examination these seem to be constructed from the above primitives.

Given these basic hardware techniques the next level is to identify hardware interconnection primitives which may be used to realize any hardware architecture of interest. In this study the Anderson and Jensen taxonomy was employed to define such an architecture space. As mentioned in the introductory discussion other architectures seem to be possible but this taxonomy seems to be representative. Therefore the basic set of hardware techniques were used to define HIPs from which any of the Anderson and Jensen architectures could be realized. What is of interest is that only a small set of HIPs is required. These are as follows.

MIMIC hardware interconnection primitives :

1. Four channel serial link (S;Ls;4) : dual mode to support internodal or intranodal information transfers.
2. Four channel parallel link (S;Lp;4) : dual mode to support internodal or intranodal information transfers.
3. Four channel bus window (S;BW;4) : address mapping is programmable to support a variety of bus window based architectures.
4. Transmission computer (Ct;dma) : a separate computer with a Lp and a Kdma to the MIMIC bus and which may control either the

S;Ls;4 or the S;Lp;4. This will support either internodal or intranodal software "transmission layers." In this HIP the Ct;dma will support either transmission through a S;Ls;4, a S;Lp;4, or it provides a dma block transfer capability within a node. Cost modularity considerations indicated the efficiency of grouping these functions.

Each of these modules has been designed and constructed for use in the MIMIC systems. Since the MIMIC system is representative of systems such as the S-100 based microcomputers [Elm79] the results of this implementation are applicable to S-100 type systems also.

Current Results : MIMIC Multicomputer Architectures

The following figures present the particular implementations of the MIMIC HIPs defined above, some of the interconnection subsystems possible with these HIPs, and some of the resultant multi microcomputer architectures. Figure 4 presents a block diagram of the basic four channel link HIP. This diagram represents both the S;Lp;4 and the S;Ls;4. Figure 4(b) and 4(c) show a PMS diagram version of these HIPs. The PMS representation is much clearer and compact as a graphical notation. An interesting point is that although the functional representations of the the S;Lp;4 and the S;Ls;4 modules are the same their effects on the operational characteristics of the system are significantly different. As an example physical distance and transmission speeds are quite different and the S;Lp;4 may also serve in a variety of control operations which are impossible to the S;Ls;4. This is a vivid illustration of the need to examine the implementation considerations beyond the functional definitions to see the important effects of using different interconnection mechanisms.

Figure 5 presents a PMS diagram of a typical microcomputer such as a MIMIC or S-100 based machine. Figure 6 presents a PMS diagram of the transmission processor HIP with the various module features defined. Note that this is quite similar in nature to the machine of Figure 5 but the buffered bus connections and the Lp and the Kdma provide the hardware means to realize the transmission functions.

Figure 7 combines the modules of figures 4,5 and 6 to show a typical node which consists of a node microcomputer and the HIPs necessary to realize a variety of interconnection operations. This is a general purpose structure which will permit implementing a number of the architectures from the Anderson and Jensen taxonomy.

Figure 8 presents a block diagram of the four channel bus window and a corresponding PMS represntation. This HIP actually realizes four independent bus window channels in a single hardware module. This was done for reasons of cost modularity as it was reasonable to multiplex the BW control and timing functions among separate bus window channels.

Figure 9 presents two possible architectures feasible with the BW;4 HIP. In figure 9 (a) is shown a three node configuration which permits either a DDL type machine or a true IDS configuration. The address maps in the bus window are programmable which permits defining a mapping of bus window address spaces which functionally define different architectures. As an example each node might have to manage the ring messages through a store and forward mechanism if the BW address maps only permitted unidirectional information transfer. In another case the

BW adress maps may be set to permit any C in the network to access any other C indirectly through the BWs. In these cases the functional software operations and system behavior will differ.

Figure 10 presents a four node ring implementation of the structure originally shown in figure 2(b). In this case the configuration of figure 7, which employed the Ct and S;Ls;4 HIPs, is used to build a higher performance DDL architecture. Other basic structures are also possible from this fundamental node.

Present studies show that nine of the ten possible Anderson and Jensen architectures are possible from these hardware primitives. Only the ICSB structure is not possible using these mechanisms.

These systems reflect the possible multicomputer hardware structures possible from a fundamental set of building blocks. The next phase of the study will define system level characteristics which are of interest. These system attributes will then be examined in light of a particular HIP/ISS implementation to draw conclusions about the effects of particular hardware techniques upon system level behavior. From this information additional studies will explore developing methods for matching the best hardware architecture to the needs of a given design problem.

Current Results : Commercial Multimicrocomputer systems

Where the MIMIC hardware primitives reflect specialized design whose primary prupose is a consideration of hardware architectures, another facet of the project being presented here is concerned with distributed software issues. It is not presently clear what are the relationships between the structure of a distributed software system and the underlying distributed hardware structure. To study this and to permit focusing upon the distributed software issues an eight node LSI11 distributed microcomputer has been designed and is being constructed. This system has a general purpose ISS whose structure may be varied under program control. The ISS is constructed from serial link HIPs to provide full point to point communication between any node in the system or any permutation of interest. The software system defines the particular ISS structure which is to be used. This permits an easy and flexible shifting between different fundamental architectures as the interests of the software research efforts specify a change.

Some examples of this system are shown in figures 10 through 12. The basic system is shown in figure 10 (b) which consists of eight LSI11 nodes, as defined by figure 10 (a), a full point to point serial link ISS, and a PDP 11/60 for program development support. An example of a full point to point ISS is shown in figure 11 (c). Any node can directly communicate with any other node through dedicated serial links. This is a completely general scheme with some of the possible permutations shown in figures 11 (a), 11(b), 11(d), 11(e), and 12. The software system defines the dedicated serial links which it chooses to use and that defines the ISS. As can be seen this permits DDL, ICDS, ICDL, DDC, IDDR, and IDDI architectures which are six of the ten basic Anderson and Jensen structures. This will provide a great flexibility to support experimental work with distributed software.

To show other features of this system figure 12 presents a dual DDL system. Basically this is a system with an alternate path to compensate for a failure of a given dedicated link or the node which supports that link. Such a system provides an experimental vehicle for considering reliability and software recovery issues. In this system two DDL loops are physically defined. During normal operations only an active DDL path will be employed. In case of a link failure an alternative link will be activated. In case of a node failure the node tasks need to be tranferred to an active node for a degraded mode of operation and associated alternative links will become active.

Additional system architectures are possible from this general purpose serial link ISS.

Conclusions

It seems to be possible to identify a set of architecture types which may be used to characterize and study multiple computer systems. It also seems possible to identify the fundamental hardware techniques which are employed in implementing hardware interconnection mechanisms. These two results have been combined in this project to identify an initial set of hardware interconnection primitives (HIPs) which may be used to build functional representative interconnection subsystems (ISSs) which result in a distinguishable architecture type. An initial set of these HIPs have been designed and constructed to permit the convenient construction of the various multiple microcomputer architectures of interest.

There are a number of major issues which remain to be studied prior to the development of well ordered design procedures concerning multiple microcomputer systems. Can the types of all possible architectures be identified and characterized in a meaningful way? The Anderson and Jensen taxonomy indicates that this is possible but a more complete taxonomy is still necessary which will also consider implementation attributes. Can an encompassing set of functional, modular primitives be identified which support the implementation of any multiple computer system? This project has accomplished this goal within a limited context. This successful result indicates that such an objective may be generally possible. The PMS system provides a conceptual base for identifying primitives in unicomputers. It remains to be seen whether this can be done for multiple computers. Does an appropriate description language exist which can be used to describe, simulate, and manipulate the architectures of multiple computer systems as a support for the design process? In a unicomputer the architecture is often characterized by either the structure of the processor (stack, register oriented, etc.) or by the processor instruction set itself [Amd64, Brown76]. What is the proper way to characterize the multiple computer architecture? Perhaps the best way is to characterize such systems in terms of their interconnection subsystems which must eventually encompass both the hardware subsystem and the software subsystem. What are the important attributes of this overall interconnection subsystem? What are the tradeoffs between hardware and software mechanisms? What is the relationship between the distributed hardware system and the distributed software system? These and other such questions remain to studied and to be answered. It is felt that the identification of the hardware primitives and the existence of the experimental system described in this report will provide an effective experimental means to examine some of these issues.

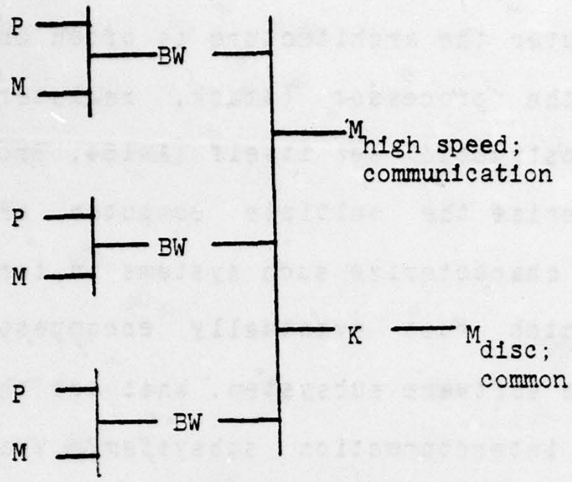


Figure 1 (a) An example of a high speed interconnection subsystem using bus windows and a shared memory

System characteristics
1. Tightly couple
2. High performance ISS
3. Nodes are physically close
4. Expensive ISS
5. Higher reliability
6. Good modularity
7. Central OS kernel
8. etc ?

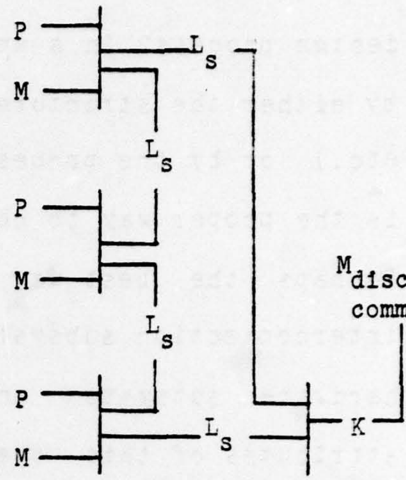


Figure 1(b) An example of a low speed interconnection subsystem using serial links.

System characteristics
1. Loosely coupled
2. Low performance ISS
3. Nodes may be physically far
4. Inexpensive ISS
5. Low reliability
6. Good modularity
7. Distributed OS kernel
8. etc. ?

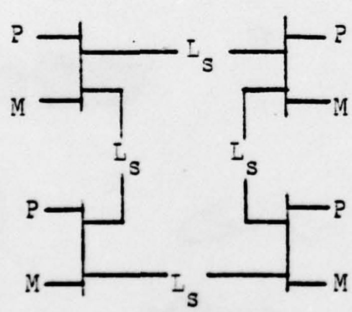


Figure 2 (a) DDL architecture(Ander76) in which node P must support both ring message transmission and local tasks

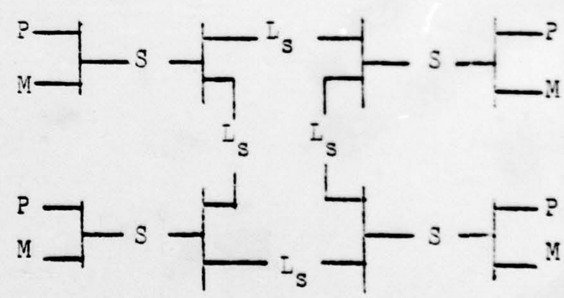
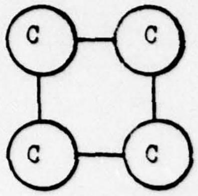
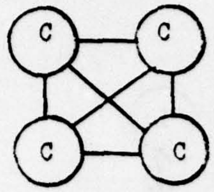


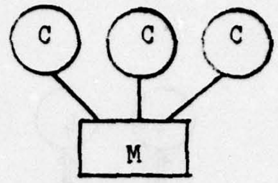
Figure 2 (b) DDL architecture (Ander76) with switch S supporting ring message transmission and node P supporting local tasks.



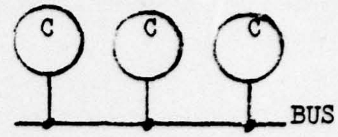
a) DDL : Direct Dedicated Loop



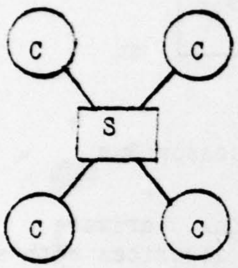
b) DDC : Direct Dedicated Complete



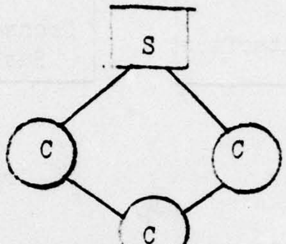
c) DSM : Direct Shared Memory



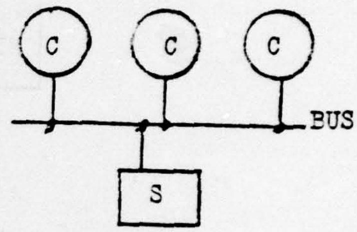
d) DSB : Direct Shared Bus



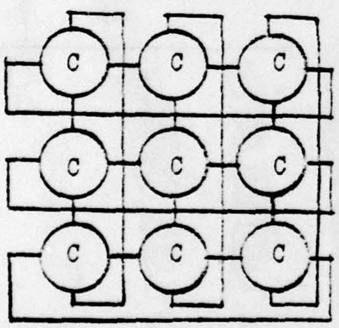
e) ICDS : Indirect Centralized Dedicated Star



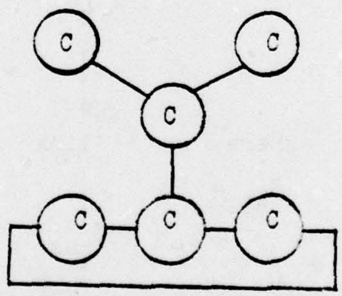
f) ICDL : Indirect Centralized Dedicated Loop



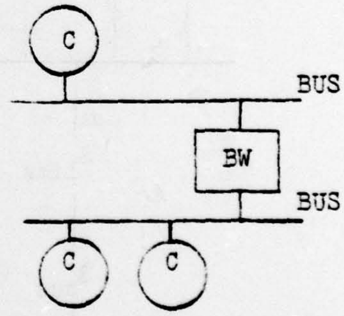
g) ICSB : Indirect Centralized Shared Bus



h) IDDR : Indirect Decentralized Dedicated Regular



i) IDDI : Indirect Decentralized Dedicated Irregular



j) IDS : Indirect Decentralized Shared Bus -Bus Window

Figure 3 Graphical summary of the ten architecture types of the Anderson and Jensen taxonomy Ander76 .

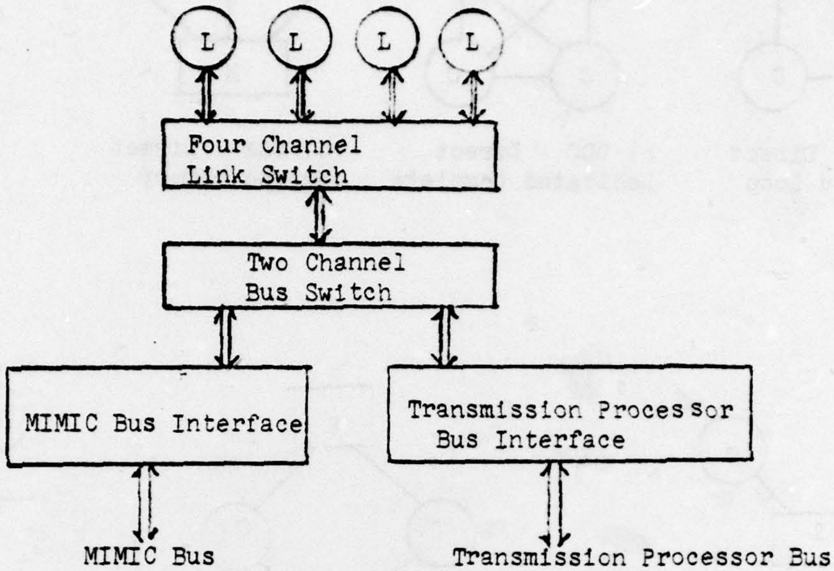


Figure 4 (a) Block diagram of two mode, four channel link, hardware interconnection primitive. Above diagram describes both serial and parallel link HIPs.

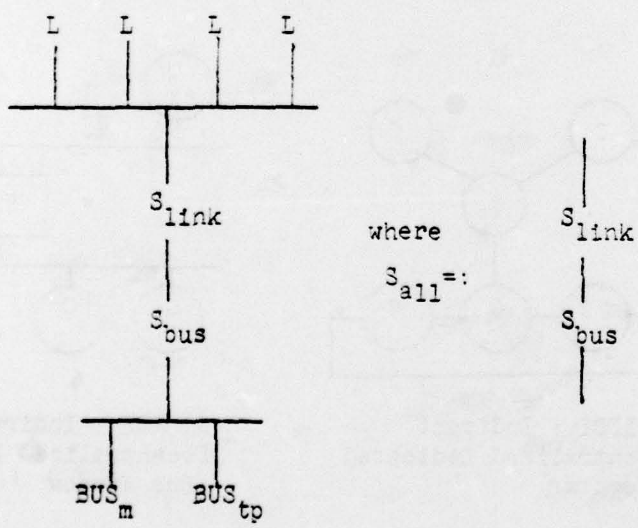


Figure 4 (b) PMS diagram of two mode, four channel link HIP

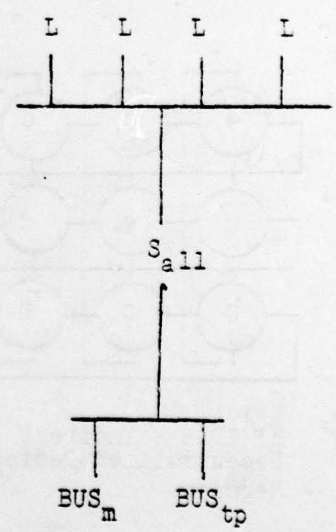


Figure 4(c) Reduced PMS diagram of two mode, four channel link HIP

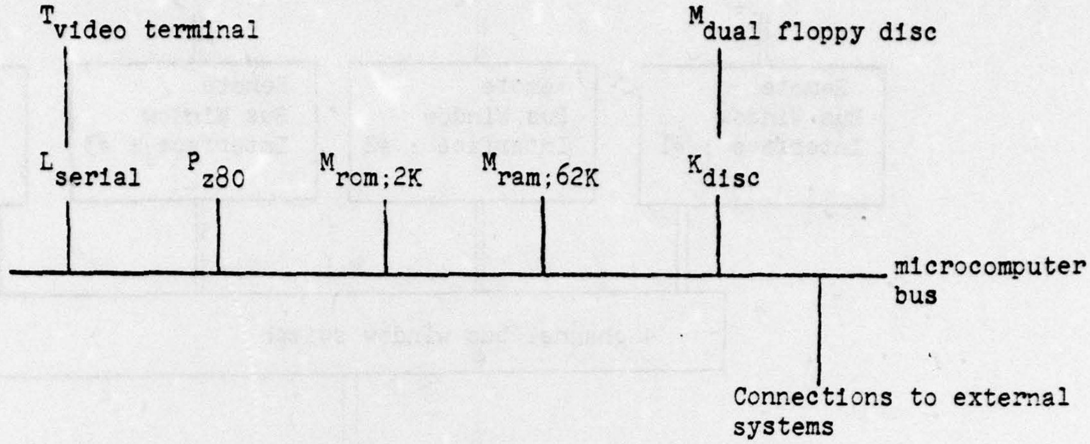


Figure 5 PMS diagram of a typical microcomputer system

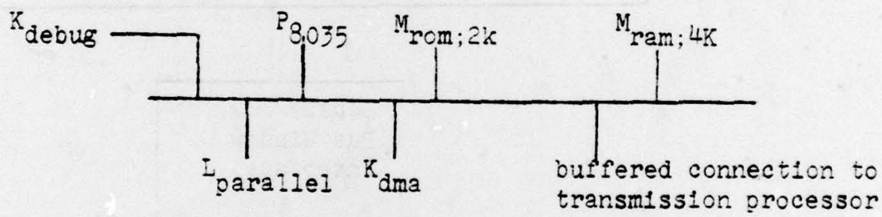


Figure 6 PMS diagram of Transmission Processor HIP

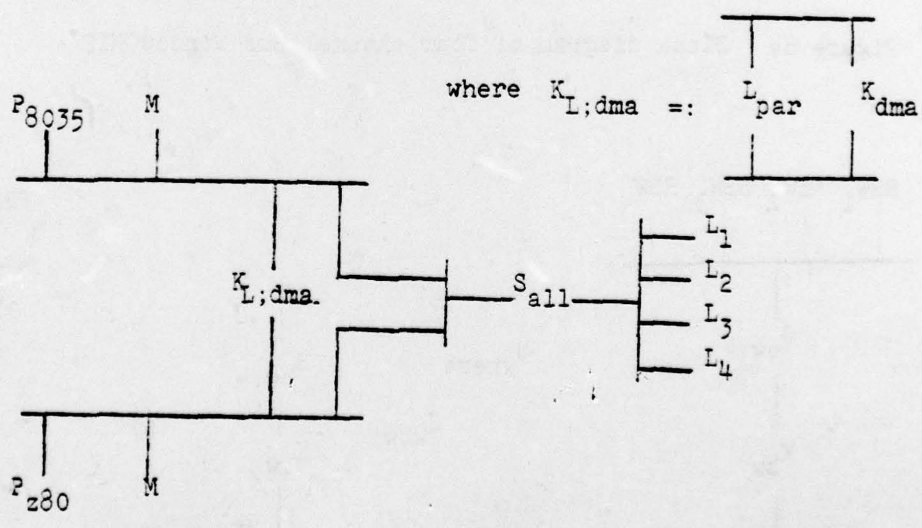


Figure 7 PMS diagram showing the interconnection of a node microcomputer, a transmission processor HIP, and a 4 channel link HIP.

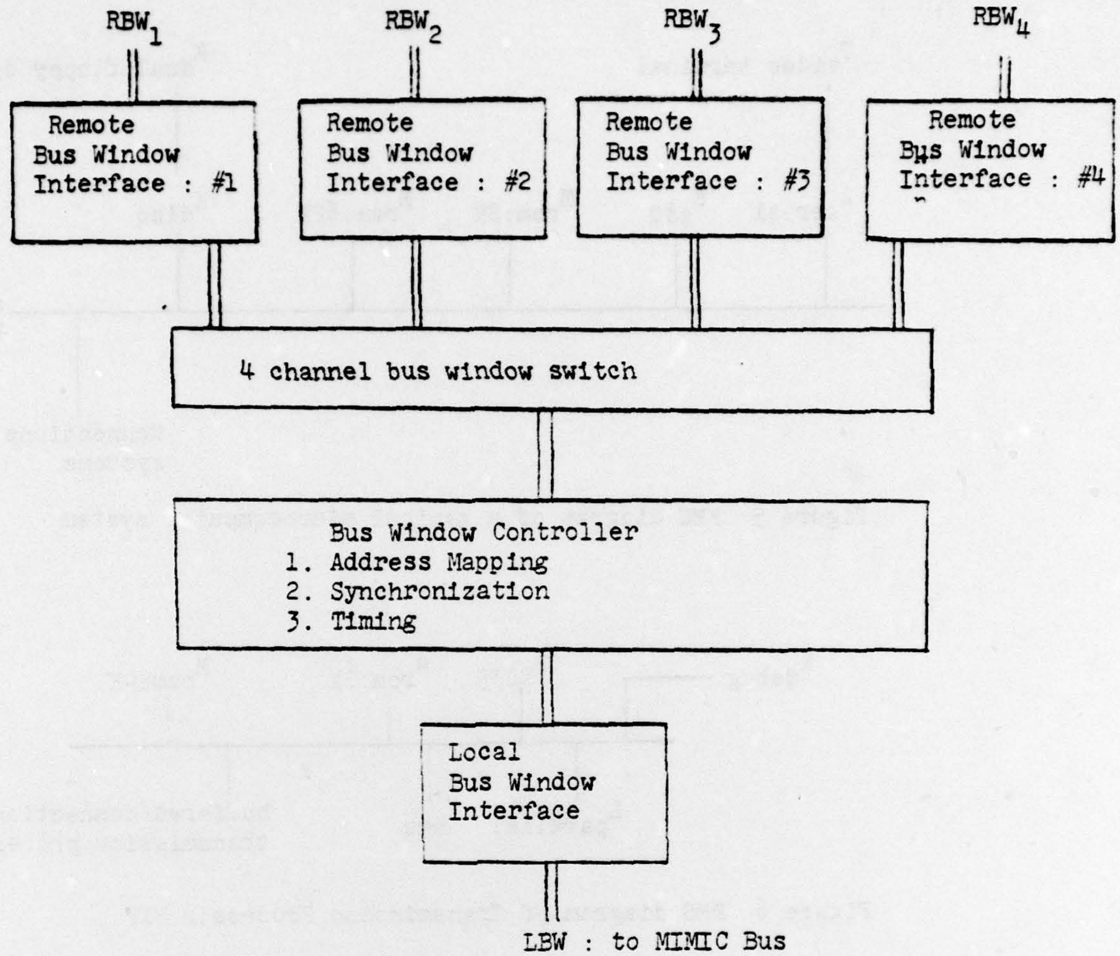


Figure 8a Block diagram of four channel Bus Window HIP

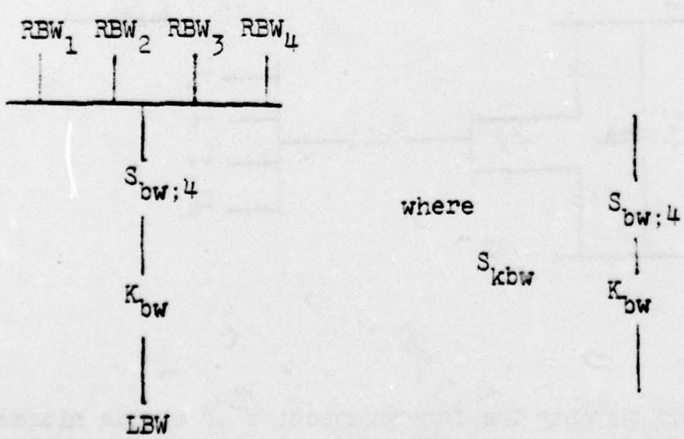


Figure 8b PMS diagram of 4 channel bus window HIP

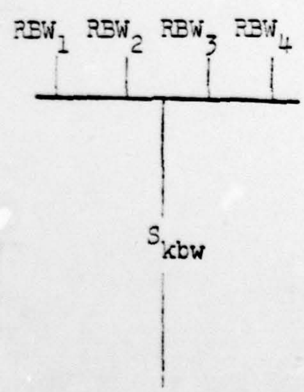


Figure 8c Reduced PMS diagram of 4 channel bus window HIP

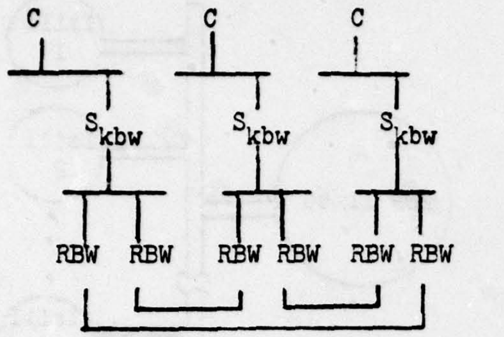


Figure 9(a) Three node system with Bus Window HIP implementing ISS. Possible architectures are DDL if window addresses are set so must do "store and forward" to pass messages, and IDS if window addresses are set so any C may access part of address space of any other C

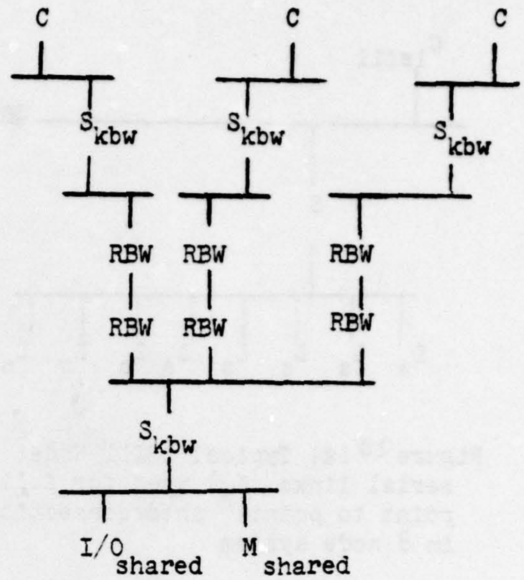


Figure 9 (b) Three node system with Bus window HIPs acting to establish a DSM architecture. Window addresses are set so each C can access any part of the shared M.

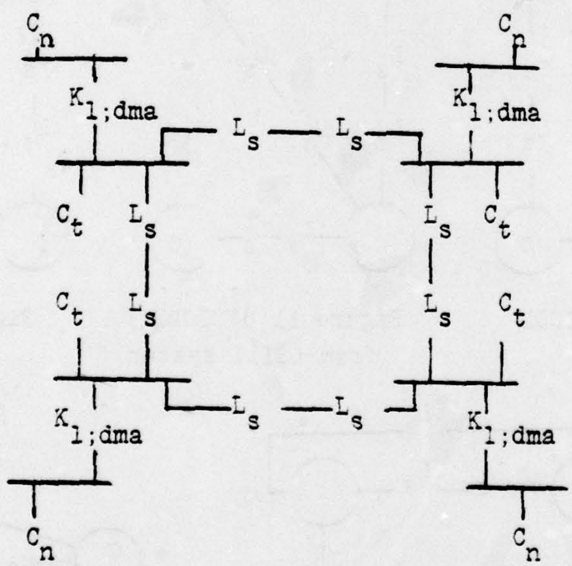


Figure 10 Four node DDL architecture with the two HIPs (1) Transmission Processor, and (2) Four channel serial link being used to implement the ISS. This is a specific implementation of the general structure shown in figure 2 (b). Note that each C_t/L_s HIP combination results in two unused serial links pernode. These may be used to support other system features or provide backup links for reliability purposes.

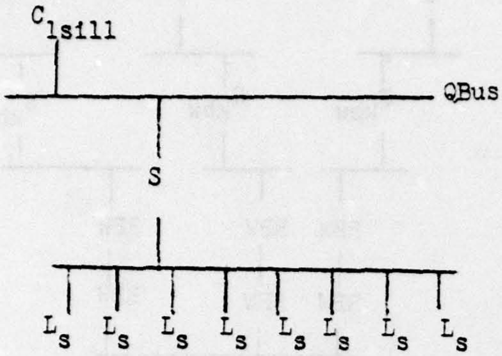


Figure 10 (a) Typical LSI11 Node; serial links (L_s) used for full point to point interconnection in 8 node system

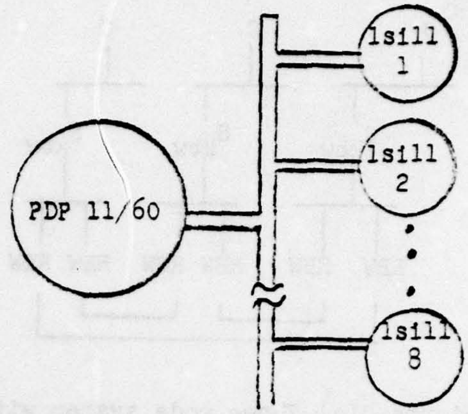


Figure 10 (c) Distributed LSI11 System with full point to point interconnection as per figure 10 (c)

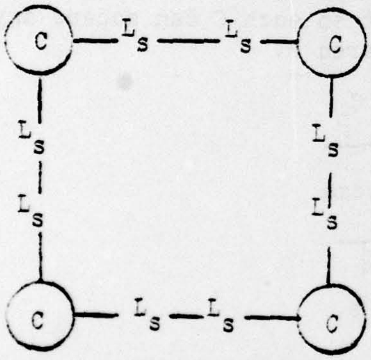


Figure 11 (a) DDL or ICDL from LSI11 system

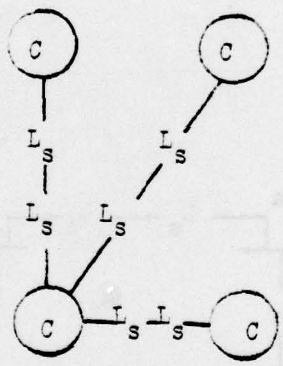


Figure 11(b) ICDS from LSI11 system

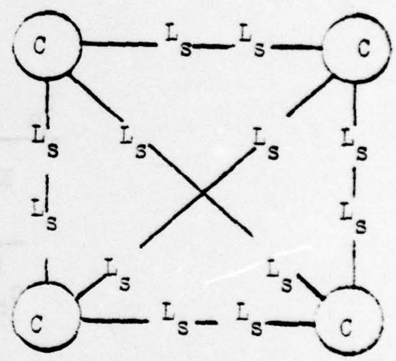


Figure 11 (c) DDC from LSI11 system

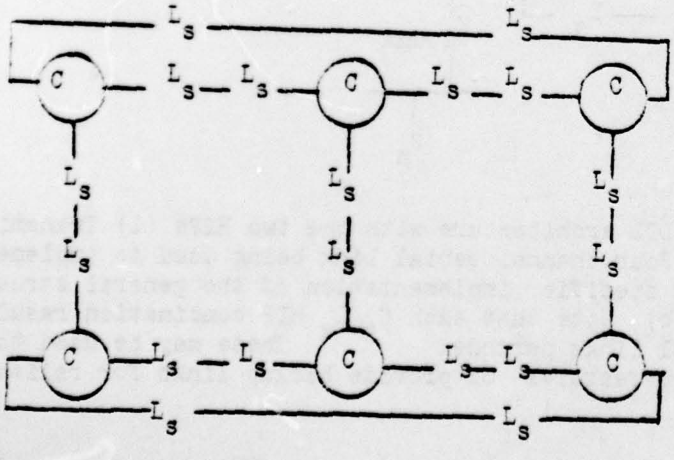


Figure11 (d) IDDR from LSI11 system

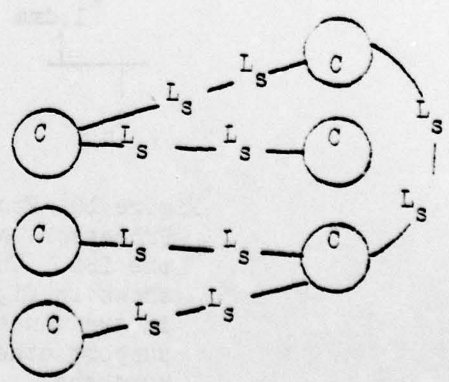


Figure11 (e) IDDI from LSI11 system

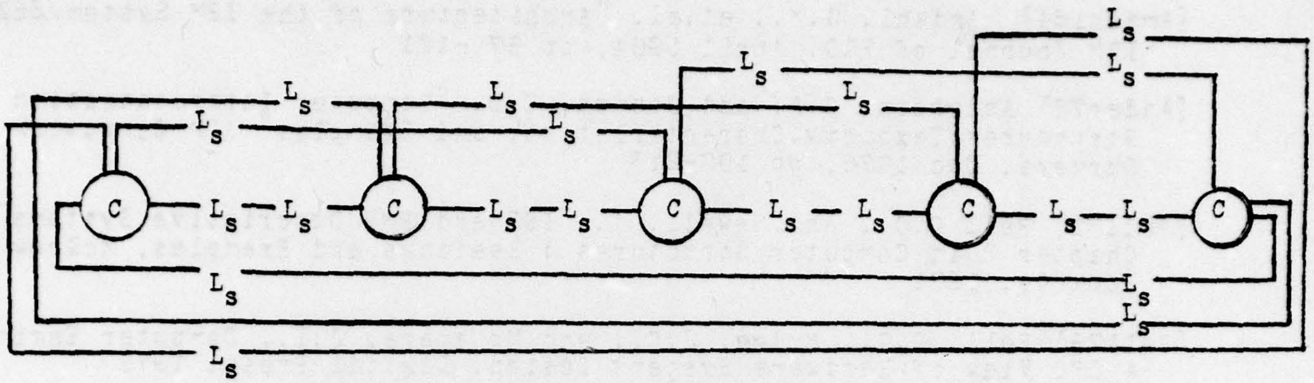


Figure 12 Dual DDL architecture with LSI11 system. Each connection in the ring has an alternate path in case of a node failure. Simple extension of basic DDL architecture to improve reliability sensitivity.

BIBLIOGRAPHY

[Amdahl64] Amdahl, G.M., et.al. "Architecture of the IBM System/360" IBM Journal of R&D, April 1964, pp 87 -121

[Ander76] Anderson, C.A. and Jensen, E.D. "Computer Interconnection Structures:Taxonomy,Characteristics, and Examples" ACM Computing Surveys, Dec 1976, pp 197-213

[Bell71] Bell C.G., and Newell, A. "ISP and PMS Descriptive Systems", Chapter 2 in Computer Structures : Readings and Examples, McGraw Hill Book Co. 1971

[Bell78] Bell, C.G., Mudge, J.C., and McNamara, J.E., Computer Engineering: A DEC View of Hardware Systems Design, Digital Press, 1978

[Burr77] Burr W. E. and Smith, W.R. "Comparing Architectures" Datamation, Feb 1977, pp 48-52

[Carey77a] Carey, B.J. "A Microprocessor Laboratory for a University Environment" IEEE Computer, Jan 1977, pp 40-46

[Cren72] Crenshaw, J.H. "Federated vs Integrated Computer Systems" in Computers in the Guidance and Control of Aerospace Vehicles, NATO, Feb 1972, pp 9-22

[Elm79] Elmquist, K.A., et.al., 'Standard Specification for S-100 Bus Interface Devices", IEEE Computer, July 1979, pp 28-52

[Flynn72] Flynn, M.J. "Some Computer Organizations and their Effectivenes" IEEE Transactions on Computers, Vol C-21, no 9, Sept 1972, pp 948-960

[Free79] Freeman, H. A., and K. J. Thurber, "Issues in Local Computer Networks", Proc. Int'l Conf. on Communications, Vol. 2, pp. 20.3.1-20.3.5, 1979.

[Kro72] Kropel, W.J. 'An Experimental Data Block Switching System' The Bell Systems Technical Journal, July 1972, pp 1147-1165

[Loom73] Loomis, D.C. 'Ring Communication Protocols,' University of California at Irvine, Department of Information and Computer Science Tech Report No. 26, Jan 1973

[Mcf76] Mcfayden, J.H. 'Systems Network Architecture,' IBM Systems Journal, Vol 15, No. 1, 1976

[Powell78] Powell, D., et. al., 'RHEA: A System for Reliable and Survivable Interconnection of Real-Time Processing Elements", Proc. 1978 Int'l Symposium on Fault-Tolerant Computing, Toulouse, France, June 1978, pp. 117-122.

[Renn78] Rennels, D. A., A. Avizienis, and M. Ercegovac, "A Study of Standard Building Blocks for the Design of Fault-Tolerant Distributed Computer Systems", Proc. 1978 Int'l Symposium on Fault-Tolerant Computing, Toulouse, France, June 1978.

[Swan77] Swan, R. J., A. Bechtolsheim, K. Lai, and J. Ousterhout, "The Implementation of the Cm* Multi-microprocessor", AFIPS Conf. Proc., Vol. 46, NCC 1977, pp. 637-644.

[Var79] Varanka,M. and Carey, B.J. "MIMIC : The Microprocessor Independent Microcomputer : Definition of the MIMIC Bus" University of Connecticut Computer Science Technical Report CS-79-9, May 1979

[Wit78] Wittie,L. "MICRONET: A Reconfigurable Microcomputer Network for Distributed Systems Research" Simulation, Nov 1978, pp 145-153

INITIAL DISTRIBUTION LIST

Defense Documentation Center Cameron Station Alexandria, VA 22214	2
Library Code 0142 Naval Postgraduate School Monterey, CA 93940	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93940	1
Professor Bernard J. Carey Code 52Ck Naval Postgraduate School Monterey, CA 93940	10
Professor Earl E. McCoy Code 52My Naval Postgraduate School Monterey, CA 93940	3
Professor G. H. Bradley Chairman Department of Computer Science Naval Postgraduate School Monterey, CA 93940	20
Professor D. E. Kirk Chairman Department of Electrical Engineering Naval Postgraduate School Monterey, CA 93940	1