(3)

SRM/18

THE UNIVERSITY OF NEWCASTLE UPON TYNE

COMPUTING LABORATORY

S(Magnabus) A Multi-Unibus Switch for PDP-11- or PDP-11 Multiprocessing.

G. Bell

27th May, 1971

This memorandum was produced by Professor Bell at Carnegie-Mellon University, and, being complementary to the description given in SRM/16 of the switch that was designed during his visit to Newcastle, is also being distributed as an SRM.

Introduction

This memo summarizes the design which is the result of a one-week meeting at the University of Newcastle Upon Tyne*. The principals who put forth the design were G. Bell, H. Lauer and B. Randell. Various others participated in critique sessions including J. Eve, W. Lynch, J. Given, and others. At least two written versions of the meeting are planned: this one, and one by Hugh Lauer (Newcastle) which will exist in their Research Memorandum Series. Little attempt will be made to describe alternatives which were discarded, although some of the rationale for various decisions will be noted.

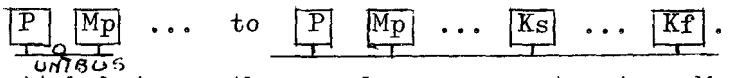
The design is a switch structure allowing a number of PDP-11 components to intercommunicate with one another to form a large, multiprocessor computer, or several multiprocessor (or uniprocessor computers). Although it is the intent of this design to have a computer structure for multicomputer and multiprocessor research, we believe it also fulfills a production role. Since the PDP-11 Unibus already fulfills the role of providing very general intercommunication among a uniprocessor structure, it was the intent of this design to extend this capability to multiprocessor. That is:

$$\frac{\text{Unibus}}{\text{Uniprocessor}} \approx \frac{?}{\text{Multiprocessor}}$$

where ? is a switch for multiprocessors, but has countless possibilities for names (e.g., Magnabus, Manybus, Superbus, Multibus, Unibus Fleet, a Roundhouse, a Bus Station, an Interchange) none of which seem quite suitable. I have arbitrarily picked a name for this memo.

S(MAGNABUS)

Basically, what we describe is the switch shown in Figure 1

where: C can be 

;
S(Magnabus) is the switch between the p and m component sets; Mp is primary (program) memory; Ks is for a slow device control (e.g., Teletype, Card reader, etc.); and Kf is fast device control requiring direct access to memory (e.g., disk, mag. tape, display); Ku is a unibus control which, for NPR requests looks like a Unibus, but otherwise cannot issue requests or acknowledge interrupts.

Magnabus is logically a switch which allows the p components busses on the right to access the m component busses on the left.

*G. Bell was guest at Newcastle, May 10-14, 1971.

The exact structure of S is a cross-point providing up to min(m,p) possible simultaneous conversations. The sole function of Magnabus is to decide which of the m x p cross-points is to be closed as a function of p and m requests to communicate. The specific switch (or switches closed is a function of the addresses of the requestor. Thus, magnabus has parameters within it which route requests to the requestee. These parameters are nominally set by manual switches (i.e., toggle and rotary). Physically, S is shown in Figure 2 where nominally $p < m$: i.e., for performance reasons $2p \approx m$ for model 20. S(Magnabus) is a centralized switch with P + M Unibus cables entering it, as opposed to being a distributed switch -- as in common in multiport memories in m or p -- which would require p x m cables. For the time being, lower cost logic (relative to cables), suggests a central organization of this type even though some modularity may be sacrificed. The modularity aspect will be discussed later. Also, a centralized switch avoids the multiple receivers and transmitters inherent in a distributed bus thereby improving performance and reliability due to noise and grounding. The reliability aspect of a centralized switch will also be discussed; however, it appears to be no worse than distributed switches.

Now that the reader has some idea of what the switch is, several aspects of the design will be discussed. These are:

Ground rules for S(Magnabus) Design

Addresses in the Multiprocessor on Multicomputer structure

Communication dialogues via Magnabus

Performance

Reliability, Repairability

Magnabus Parameters: Manual Control of Structure; and Component Names

Other communication in a Multiprocessor

Logical and Electrical Design

Extensions and Modularity

GROUND RULES FOR S(MAGNABUS) DESIGN

The principal constraint on the design was that it would use standard PDP-11 Uniprocessor components (i.e., p, Mp, Ks, Kf, etc.) without modification. This also includes standard housing of components, <u>except</u> S(Magnabus). Similarly it was assumed that components of a given type would be completely interchangeable (i.e., all 4K word memories). Furthermore, we assumed that subsequent hardware mapping devices, and processors could also be compatible with S(Magnabus).

Secondary ground rules included the use of standard logic and electrical components, although such a design could no doubt

use LSI technology.

Philosophically, we have adapted the view that components are addressed anonomously, and that control resides with a program (as opposed to a particular processor). Although this view is present in the design, it has the capability of being configured to operate in a restricted fashion (e.g., all interrupts go to a particular processor).

Although we would have liked to keep the current device names, (addresses), we have allowed these to change, in order to decrease the cost and avoid the problem of introducing content addressable memories into the switch. Such a change only necessitates reassembly of any source code to run on one of these structures.

Also, we are attempting to have a modular design -- one which could be extended by either identical or similar modules to larger computers.

[As an aside comment, the design of controllers should be such that they cannot be overwritten by data arriving from disks and other NPR devices.]

ADDRESSES (NAMES) IN THE MULTIPROCESSOR OR MULTICOMPUTER STRUCTURE

Uniprocessors

An especially interesting view that one might take of a computer is the addresses (names) which it uses. In a PDP-11 Uniprocessor there are:

accessed by Pc	2^{16}* - 2^{12}	for Mp addresses	
	2^{12}	for Ks|Kf addresses	
accessed in a second private address space	8	for Pc - Registers	addressed via the console as part of Mp|Ks|Kf (ignore)
	1	for Pc - Processor state word	
	4	for interrupt level names	

In this design, we are able to provide two kinds of structures depending on whether multiprocessors or multicomputers are configured.

Multiprocessors

1. There is one public address space of 2^{16} + 4. All components have unique names. Each Pc has its 9 private registers.

2. There is one public address space of less than 2^{16} + 4. Each Pc has access to a large private space, plus public address space, thus total space to any Pc is 2^{16} + 4 + 9 registers.

---

* or 2^{18} depending on the Pc

Private addresses mean that no two processors or sets of components share the addresses. For example, a public disk would normally not be able to access private memory.

Multicomputers

There can be multiple public address spaces of $2^{16} + 4$ words plus the 9 times p. Each computer has its own address space. Each computer is either a uniprocessor or a multiprocessor (above).

COMMUNICATION DIALOGUES VIA MAGNABUS

Basically, we have all the dialogues present in a uniprocessor. In effect, it is the purpose of Magnabus to connect sets of two Unibusses together for dialogues: one from the processor controlled set (1 of p); and one from the other set (1 of m). Up to min(m,p) simultaneous connections are made at a time e.g., at a given instant, we might expect to see the intercommunication of Figure 3.

0. C. All local communication via 1 of p unibusses connected to Magnabus; these include local: Mp ← Pc; Kf|Ks ← Pc; Kf|Ks → Pc; and Mp ← Kf dialogues. None of these appear as traffic within Magnabus.

1. Mp ← Pc; Pc accesses data from an Mp which is connected to 1 of m Unibus parts of Magnabus. The particular Mp address control is under the control of Magnabus parameters.

2. Kf|Ks ← Pc; Pc sends control information to a fast or slow control. This appears to be the same kind of traffic as Mp ← Pc traffic (1 above).

3. Kf|Ks → Pc; Interrupts are sent to a Pc. A K requests interrupt attention. The particular Pc it is routed to is a function of the Magnabus parameters. In this case, the possible connections are selectable (at the Magnabus) for each processor:

   a. Never interrupt a particular Pc.

   b. Allow interrupts to go to a Pc depending on the interrupt level Pc is operating at. Here, all interruptable Pc's are looked at, and the one operating at the lowest level is chosen by Magnabus.

4. Mp ← Kf; Direct memory accesses (NPR) are sent to an Mp from a fast control. In this case two crosspoint switches of Magnabus are closed (sequentially). First, 1 of the p Unibusses is selected to carry the address of the NPR; and second 1 of the m switches is closed corresponding to the selected Mp address. Note, after the first switch is closed, an NPR request appears exactly like a Pc request to Mp.

In the case of Mp ← Kf requests, the trunk used to transfer the data is selected on a basis which is identical to that used for interrupts to Pc. (see 3 above) Here, if there is lots of traffic in the system, it would be advisable to use 1 or more Ku's. The Ku is a control that looks like a processor to acknowledge NPR requests. (It cannot issue requests nor can it acknowledge interrupt requests).

PERFORMANCE

The performance of the structure can be computed almost exactly, given m, p, and the following parameters:

tp -- a distribution of processor times from a memory request until the next request for the program.

ta, tc -- the access and cycle times for Mp.

the location of the programs in the various memories, including whether interleaving is used.

tio -- the rate of transfers from Kf's which interfere with p requests.

Strecher gives closed form solutions in terms of these parameters. If, however, we assume tp = tc - ta (i.e., the processing time is equal to the memory rewrite time), the number of memory cycles rate is:

$$UER = \frac{m}{tc} \times \left(1 - \left(1 - 1/m\right)^{p}\right)$$

and if we look at the second factor, as the interference, for p = m, then the second term and UER are:

m = p:	Interference:	UER:
1	1.0	$1/tc$
2	.75	$1.5/tc$
3	.71	$2.1/tc$
4	.69	$2.8/tc$

Thus, it is likely that m should be larger than p, also since there will be Kf, using the set of m lines, we are assured that m should be greater than p.

Strecher also gives other formulations for tp >tc - ta and tp < tc - ta, but to a first approximation the above formula can guide us. Namely, because of interference due to Kf, it seems appropriate to plan that $2p \leq m$.

As a second consideration, the switching time, ts, of Magnabus acts almost directly to increase ta (and tc). Here, preliminary estimates for ts is 100 ns.

MAGNABUS PARAMETERS: MANUAL CONTROL OF STRUCTURE AND COMPONENT NAMES

Ideally (perhaps) we would like content addressable memories each component each time an address is placed on a Unibus. Unfortunately, such a structure would present control problems and be expensive. What we have done is provided the ability to name (address) various device classes which attach to each of the m- unibus ends emanating from Magnabus. Basically, the reader should think of the control for Magnabus as being separated into 2 x m - disjoint sets of addresses corresponding to the m-busses. That is the control is organized in row modules for each of the m unibusses in Figure 4.

The control portion of a given row of m has to decide whether p requests are to this particular row (i.e., devices on its controlled unibus); and if it also has to route NPR and priority interrupt requests to a particular processor. The means it choses for both these are on the basis of control parameters set in it. Nominally these control parameters would be set by manual switches, but they could also be registers. The control parameters are:

1. Memory-space address -- specifies the values of memory. Address on the attached unibus. For multiple memories, memory size would also be specified, and odd-even addressing (possibly).

2. Device-space address -- specifies the io device addresses of the attached Unibus. A parameter would indicate the size of the address part to be looked at.

3. Ignore processor to memory requests (p x 1-bit switches) which disconnect the processors from this Unibus.

4. Ignore processor to device requests (p x 1-bit).

5. Interrupts to processor (p x 1-bit switch). Ignore or alternate requests to this processor.

6. NPR requests to processor (p x 1 bit). Ignore or alternate requests to this processor.

OTHER MULTIPROCESSOR COMMUNICATION METHODS

It may also be necessary to provide other means of intercommunication in addition to the shared memory and shared i/o devices implicit in the above scheme.

The simplest scheme would rely on either timers and/or clocks, followed by polling and communication via primary memory.

Some sort of interprocessor interrupt may also be desirable which is used to signal for assistance among the tasks. At least two schemes of this type can be used: one -- a bit would be assigned to each p, and all processors would have the capability to write the bit. Presumably only the processor being interrupted could clear the bit. The address space for this device would be partially public (for setting) and partially private (for reading and clearing). Alternatively, each p could have a bit which could be set for each processor indicating the processor wanting attention (p x (1-p) bits).

One final means of communication seems appropriate for highly reliable computers, the use of a computer controlled console. Using this method, a conventional manual operated console would be replaced by an interface to a Unibus. In this way, a processor could be started, stopped, single stepped, etc. Also, the processor's address space including registers would be accessible via such an interface.

RELIABILITY AND REPAIRABILITY

In building a single, central switch the reader should question whether the reliability will be lower than with a switch distributed on an element along the row. We currently believe the answer is definitely not for two reasons:

a. There are significantly less receivers, transmitters and cables (only n + p instead of n x p). Also there is significantly less logic.

b. The switch can be organized such that a failure will always look like a failure in either of the row or column Unibusses (i.e., one of m or one of p).

Normally, we would expect switch failures to affect 1 of m lines because the control is organized for all p for each m. (The design is modular in this respect). Therefore what can be done when a failure occurs?

a. Consider that m or p Unibus off; wait and then

b. Repair it.

The design is such that the weak link of a single-central component does require turning power off while a board is replaced.

There are several ways of making the switch repairable during its operation.

1. Building it in a redundant fashion so that it functions during failure. This would require redundant data on the Unibus data, address and control lines. Second, the control logic to decide which points were to be closed.

2. Use a second complete switch for standby. A structure of this type would look like that of Figure 5.

This would cost m + p more cables m + p switches and a second Magnabus.

3. Partition the Magnabus into two independent parts such that a failure only occurs on 1 of the parts. These parts could be repaired independently.

For the present, let us assume we can either repair the switch at off peak times, or are willing to pay the price (> 200%) for two switches. In the section on extensions, we will examine partioning as an alternative.

LOGICAL AND ELECTRICAL DESIGN

Physically, and logically a single Magnabus would be organized around only a few board types:

1. Unibus interfaces on the basis of m and p. Failure of these boards would be identical to a failure in either m or p.

2. Control modules for all p inputs to select the appropriate 1 of m outgoing connections. These m-modules would be identical.

3. Data and address connection modules. Most likely the best arrangement would be to put the m x p simple switches for a single bit (or several bits) on a single board. Thus, the m-controls would select which one of the p possible single switches were to be closed. The main reason to put all p x m switches on a single board would be so that p + m Unibusses could be driven and terminated at the board. Also, logic for multiplexing and demultiplexing is most efficient for this organization. (This type of organization would keep all Unibus current switching on a single board).

Another way of looking at the logical design is to consider the types of signals. These are shown in Figure 6.

Considering the dialogues on the Unibus, the following actions would be taken:

1. Mp|Ks|Kf ← P; the appropriate address switch would be closed, and depending on the direction of data transferred, the data switch and direction would be set up. The master/slave control lines would be set up.

2. P ← Kf ; NPR request; The appropriate address switch would be closed, and depending on the direction of data transferred, the data switch and direction would be set up. The Master/slave control lines would be set up. On the basis of NPR grant from P, subsequent lower level (lower priority) grants would be rejected.

3. P ← Ks|Kf; Interrupts; The interrupt requests would be almost permanently connected to a given P from a particular one of the m-requesting unibusses. That is, a request along the incoming Unibus would request, on an appropriate 1 of p Unibusses, service. The granting of a Unibus request by the appropriate p, would inhibit the grants from going to a higher level 1 of m Unibusses in the switch. Thus for both interrupt and NPR requests the grant signals would be in a hardwired sequence on the basis of the number of m-Unibusses.

As a further goal for the logical design it would be highly desirable to allow any component on the m and p Unibusses to be pluggable during operation, provided the power is turned off at the bus being attached.

EXTENSIONS AND MODULARITY

In the preceding discussions we have described Magnabus as a single monolithic switch. Obviously, we would like a modular switch such that it could be extended by increasing either or both p or m. That is, we would like a Magnabus structure which would allow extensions of the form shown in Figure 7.

Obviously, such a structure is possible, although we must ask what is the cost and performance for such a structure.

There are several problems in extending the Magnabus in both the p and m directions. Among them are:

1. The nice electrical properties of Magnabus being at the terminus are no longer possible, and the switch is now another connection to the Unibus.

2. Extensions in the p-direction have to be slow, since p + 1,...2p requests are made first, followed by a sideways request to the switch on the left for the appropriate m. Requests for a given m-Unibus have to be decided in several physical locations (Magnabusses).

3. There appears to be cases of deadly embrace, although these can no doubt be resolved by enough time.

The cabling problems and cost will undoubtedly favour building separate switches for the various sized switches. For example, the Unibusses into certain sized Magnabusses would be:

p,m	Cables in a single Magnabus	Cables in a p x m extendable Magnabus	Cables in an extendable Magnabus
2,4	6	12	8
4,4	12	24	16
4,8	24	48	32
8,16	8	16	12

Although the attendant electrical logical cost and cabling problems initially seems to suggest not building modular switches, it might be possible to build switches easily which could be extended in the m-direction. For example, these would allow structures of the form to be built. (Figure 8)

These are essentially like the multiport memory that has been a standard for extending memory systems (e.g., PDP-10, B-5500, 360/65). Extending in this direction appears to have less problems:

1. $p < m$

2. It allows system balance by extending m

3. It can provide increased reliability

4. Control is on the basis of each of the m (or 2 * m in the figure) Unibusses.

SUMMARY

The above Magnabus structure proposed allows virtually any multiprocessor and/or Multicomputer structures to be configured by simple manual switch selection given that enough components are connected to the Magnabus. With these structures system performance and reliability can be determined at configuration time in a very flexible way.

FIGURE 1.

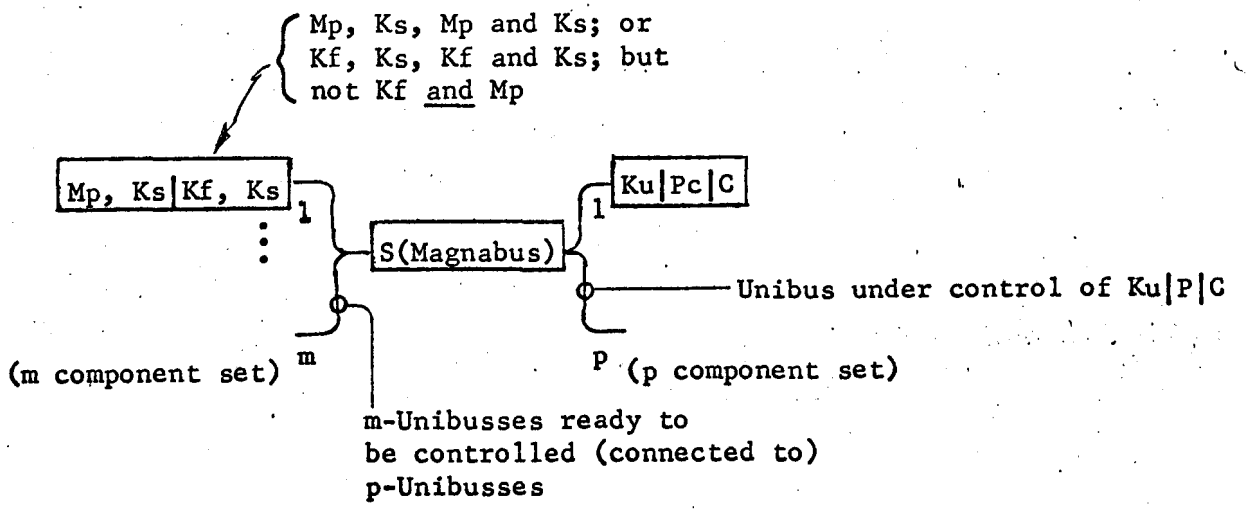


FIGURE 2.

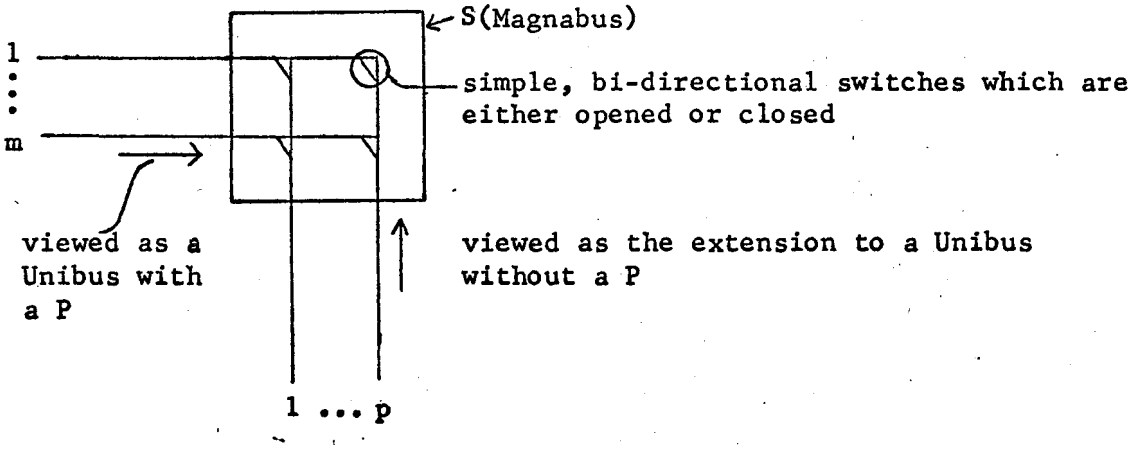


FIGURE 3.

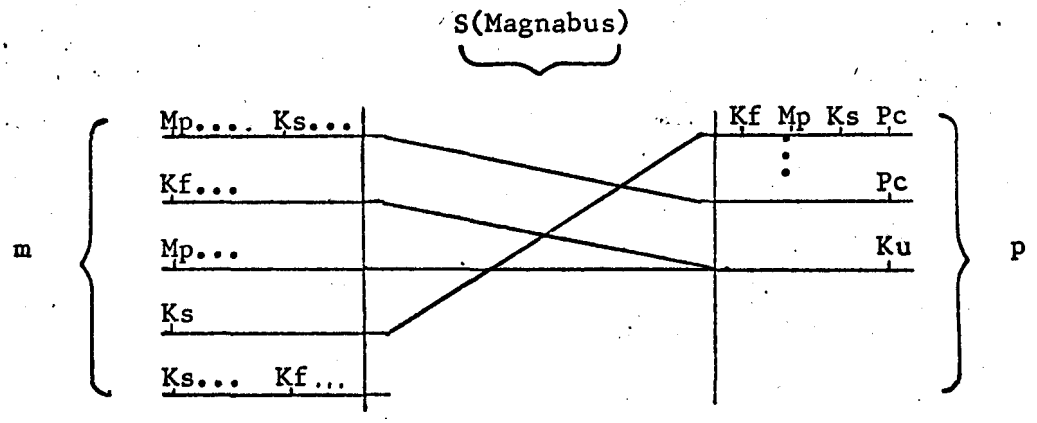


FIGURE 4.

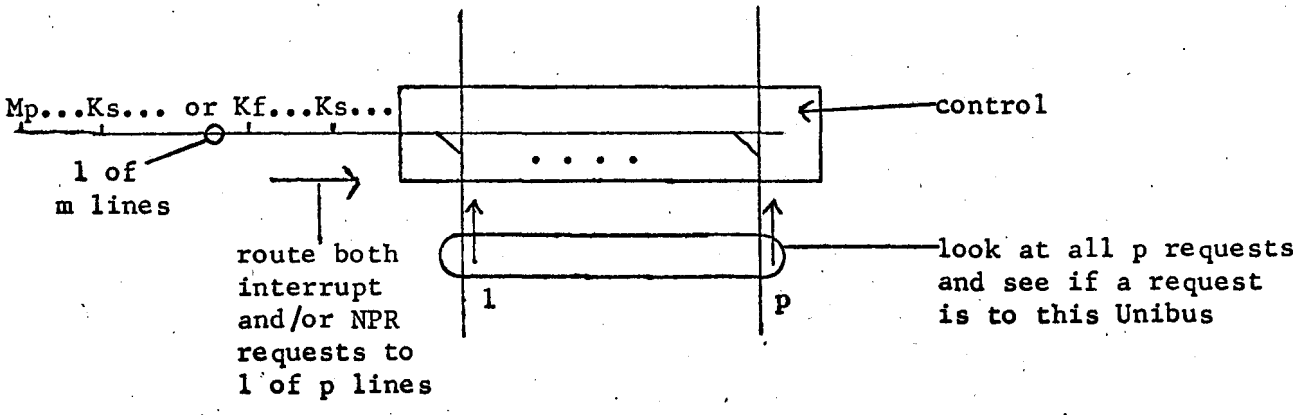


FIGURE 5.

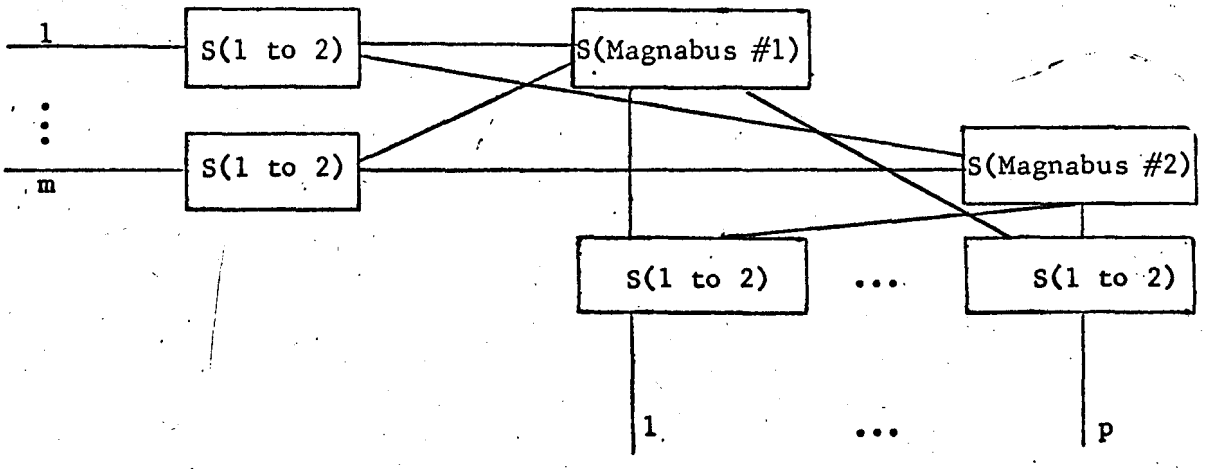


FIGURE 6.

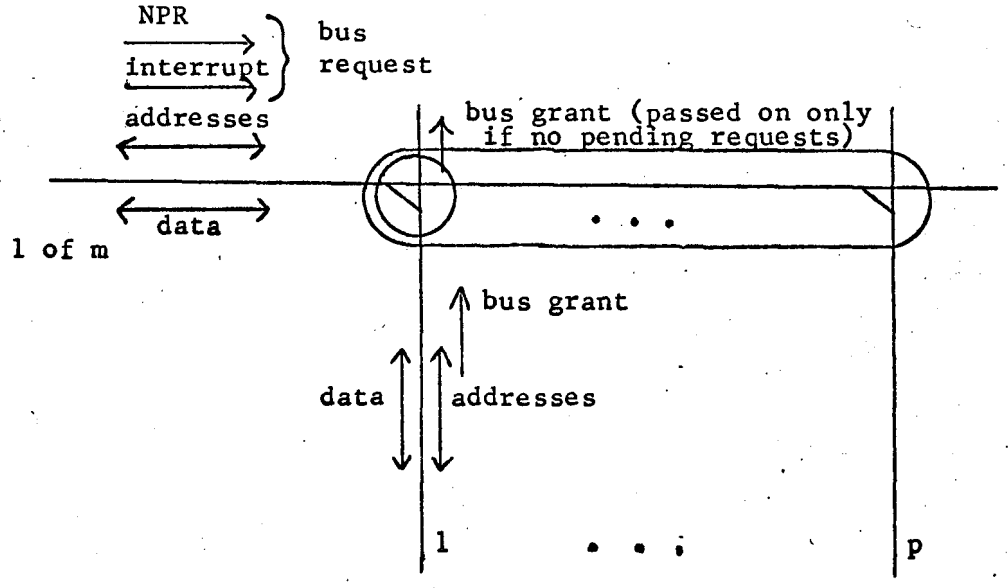


FIGURE 7.

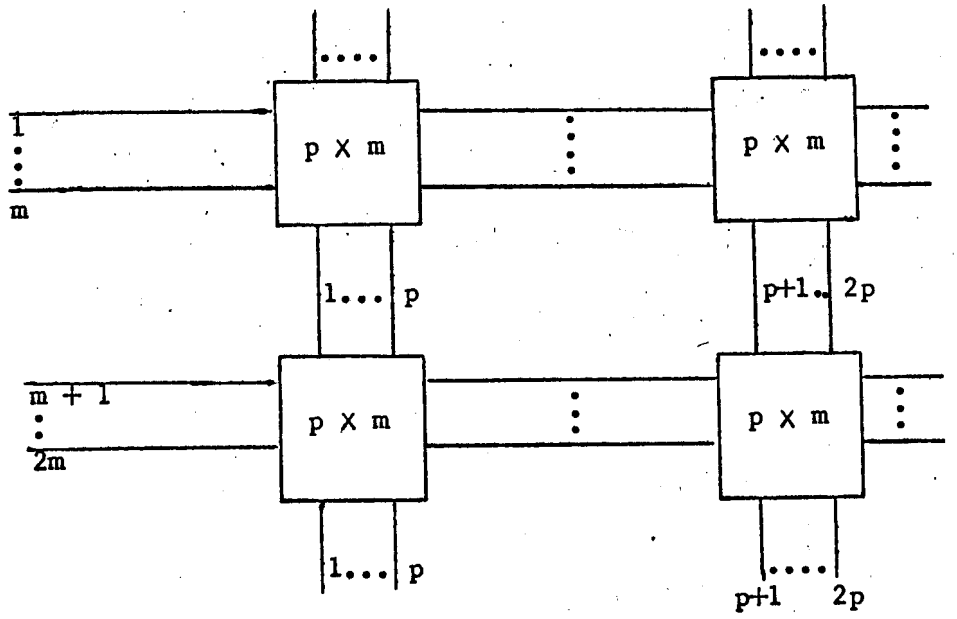


FIGURE 8.