LIBRARY
OF THE
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

D28
M414
668-73

Dewey
MASS. INST. TECH.
OCT 30 1973
LIBRARIES

WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

MESSAGE-BASED RESPONSE ROUTING WITH SELCUK NETWORKS

by

Kenan E. Sahin
Visiting Associate Professor

668-73 July, 1973

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

MASS. INST. TECH.
OCT 18 1973
DEWEY LIBRARY

MESSAGE-BASED RESPONSE ROUTING WITH SELCUK NETWORKS

by

Kenan E. Sahin
Visiting Associate Professor

668-73 July, 1973

HD28
.M414
no. 668-73

RECEIVED
OCT 30 1973
M. I. T. LIBRARIES

Abstract

Computer-to-computer communication in a very large net, parallel processing, and content-based retrieval associative memories all can be viewed in a broader context as the broadcast of a "to-whom-it-may-concern" type message and the subsequent routing of the response messages to the broadcast source. Simplicity and economies in logic and memory can be obtained if the routing does not require site information and is as decentralized as possible. We propose a procedure for such routing in a large net of modules of limited logic and memory, with connectivity primarily to near neighbors with one-way channels. The procedure is based on the Selcuk Principle which maintains that in properly constructed networks routing can be achieved with each module basing its switching decision only on the identity of its input channel(s) upon which the broadcast message first arrived - a strictly local information. One-way channel connectivity emerges as advantageous because it requires much less memory and logic then two-way arrangements, while achieving nearly the same efficiency in properly constructed nets.

071 706

Index Terms

response routing, message-based switching, parallel processing, computer communication, associative memory, content-based retrieval, pattern recognition.

There is now increasing attention being focused on computer networks which allow computer to computer communication (10, 11, 12, 14) and interactive resource sharing (18, 23, 27). Such efforts are currently well exemplified by the ARPA network (7, 13, 18, 23). The ARPANET (18) achieves intercommunication not by line switching whereby dedicated paths are established between two interacting computers but by message switching (16). Switching of the messages is really accomplished by a superimposed net of identical small processors called Interface Message Processors (IMP's). An IMP has stored in it a table showing shortest routes to all other IMP's. The table is updated frequently to take into account the level of traffic in the net. Messages arrive with a leader portion indicating the address of the computer to be reached. When an IMP receives this message, by table look up of the address it decides to which of its channels to route the message (4, 16, 23).

While in some sense the ARPANET is viewed as the prototype of computer communication it actually addresses itself to a rather restricted communication problem: one user contacting another user that it identifies (4). A more generalized situation is a member sending a to-whom-it-may-concern message to all members in a net and then receiving responses from the appropriate members. With twenty members in the net this task can easily be solved with ARPA schemes. However, as one moves to many more members the table look-up method of routing would quickly be impractical (12, 18). By the end of the decade it should not be surprising to see thousands and even hundreds of thousands of various size computers hooked into a super network of identical and small switching machines (IMP's). In such a net each computer, while specializing in the tasks of its environment, would be interested in and even depend on the activities of the other computers. Of the myriads of activities going on in the network, presumably only a few would be of use to a particular member X. If such activities of interest always originated in a particular member Y then a devoted link can easily be created between X and Y. However,

more generally activities of interest may be taking place at any one of the members. To keep posted, member X must periodically or as the situation warrants it, interrogate the remaining members. In other words, send to all "to-whom-it-may-concern" messages or "general messages." If responses are appropriate, these must then be routed to X. Of course the same must be provided for each of the other members. The situation is analogous to any telephone in a telephony net being able to ring all the other telephones and then receive responses only from a few appropriate ones. The basic problem is one of routing.

For the solution of the task let us require that the IMP's which we shall call modules remain small and simple (limited logic and limited memory) and rule out full connectivity as unwieldy. We are then left with a network of relatively simple devices (modules) connected to only few of the collection of devices. We shall also rule out location-based addresses and the table-look up schemes as utilized in ARPA (16, 23). With all these restrictions the problem boils down to the following: In a network of many many simple devices, can a device (module) initiate a general inquiry which is propagated to the entire net expediously and then have the responses routed to it without revealing its location and without the respondents indicating their location. In addition, the response must travel along a path that is reasonably close to the optimum path. If the problem can be solved with a practical scheme, then it could serve as a basis for very large computer communication networks.

Another context for the same problem is parallel processing most notably exemplified by ILLIAC IV (3,8,17,26). In its current realizations there are sixty-four processors (PE's) each connected to immediate four neighbors thus resulting in an array of eight by eight (3, 8). Each PE is also connected to a central processor (CU). Instructions and data streams emanate from CU to all PE's and all PE's execute the instructions in a lock-step fashion (3).

Transfer of data from one PE to another PE is possible but is mediated by the CU and routing worked out by the same (26). As such ILLIAC IV achieves significant speed increases in some classes of problems (3).

If the number of PE's were to increase severalfold, very likely the present architecture would remain functional. However, if PE's were to increase radically, say to hundred thousand or so, the CU would both be saturated, oversized and vulnerable. The reasonable way out would seem to be further decentralization including that of routing. Once again we encounter a situation similar to that which would arise if the ARPANET membership were to increase significantly. In other words, a reasonable way of achieving the advantages of parallel processing might be to equip each PE with the capability to interrogate all the others with "to-whom-it-may-concern" type messages, and then receive the appropriate response messages be they data, status reports, new instructions, executions of instructions, etc. If we impose the same restrictions on the "net" of PE's as we did on computer communication nets we emerge with the problem stated earlier; namely, achieving message-routing of responses to general inquiries in a large net of limited-logic limited-memory devices without a central router, without complete connectivity and without making the devices complex (which rules out table look-up schemes for switching) yet achieving close to optimum paths for the response.

Yet another context for the same problem is content-based retrieval where one desires to retrieve segments using some set of key words relayed to all locations, matched with contents and correct matches retrieved. Memories built to accomplish this task are often called associative memories (1,2,5,6, 9,15,19,20,21). Typically memory is organized as a matrix of mxn (6,20). The key word is stored in a register of m bits. Either all the bits of the key word or only some of the bits are sent through the corresponding vectors of the memory, either serially or in parallel (2, 6, 20). The advantages of

associative memories basically derive from not having to organize memory contents and from the parallelism of the search process. In tasks that require frequent interaction with memory contents these advantages would be particularly apparent (9, 15).

The powers of associative retrieval can be considerably enhanced by allowing for entry to and retrieval from memory at any site in memory by allowing each site to have limited logic as well as its existing limited memory. Operationally the problem would then be achieving the propagation of "key words" corresponding to general messages from any site to all other sites and then routing the "responses" to the sites from where the searches originated.

Even a more general context for the problem to be investigated is dividing a large network dynamically into subnetworks on the basis of task characteristics. The task would be introduced from any site available. A message containing task characteristics would then be broadcast from that site. Sites that accept the characteristics would send "responses" to the broadcast source and through the response paths be linked to it. Thenon the subnetwork thus formed, while still participating in the activities of the full network, could be particularly concerned with the task at hand.

In the abstract, all these contexts of computer communication and resource sharing, parallel processing with decentralized control, associative memory with entry provided at every site in the memory, and task-wise dynamic partitioning of a network reduce to the broadcast of a <u>general message</u> (GM) in a network and the subsequent routing of responses to the source of GM wherever the source may be in the network. We shall address the problem in the framework of multitude of limited-logic, limited-memory "modules" connected to adjoining neighbors as in Figures 2 and 3. We shall make the connections one-way since a two-way channel is in some sense two one-way channels and since as it will emerge in the paper the one-way arrangement is superior. We shall then seek a solution to

the problem of response routing in such networks without resorting to central switching and without requiring information on the location (site) of GM sources or respondents. The basic response routing (switching) idea to be explored will be to allow for the propagation of the general message to the network as rapidly as possible and then to have each module switch a response it may receive only on the basis of how the associated general message had arrived.

To obtain fundamental results without undue complication we will assume transmission time of a message between any pair of modules to be the same. This assumption will be realized in the network representation by making the channels the same length. Consequently, we will be dealing with geometrically regular networks. Furthermore, only planar networks with modules having the same number of channels will be considered.

Planar geometric regularity confines us to square and hexagonal "packing". And the equality of input output channels limits us to five classes of modules in terms of channel arrangements as shown in Figure 1. Labelling of channels is in clockwise fashion and "i_1" denotes input channel 1 whereas "o_1" indicates output channel 1 etc.

Any network with one-way channels in which response routing without location-based addressing is achieved will be called a Selcuk Network - a term coined by us. Selcuk Networks were first reported in (24) and their application to shape discrimination and angle detection discussed in (25).

Square Class C Network:

Consider the Square Class C Selcuk Network shown in Figure 2. Note that throughout only Class C modules are used and that the network is connected. Let us allow Module I which appears in the center of the network to initiate the general message. It is only for convenience that we are choosing this module; excepting for boundary conditions the results to be shown will be true for any module in the network.

Our task can be restated as follows: while the general message emanating from Module I propagates, it is desired that another module, upon receiving the general message, be able to emit a response message such that when this is routed by successively adjoining modules a response message will get to Module I. Routing rules are to be applied locally by each module and independently of others and without site information.

Let us more graphically show that there is a problem here before proceeding to a solution. Consider Figure 2.

Arrows show the direction of channels. The general message, at the end of Time 1 will reach modules a and b and at the end of Time 2 will reach Modules c, d, e and f.

Since an addressing scheme is not used, Module a has no way of knowing that the general message came from Module I. For all it knows, it might have come from Module g or even f. Not knowing the source, if it wanted to respond, it wouldn't know, therefore, where to send its response. Even assuming that it knew the source to be Module I, it could not send its response to this source since the connecting channel is one-way towards Module a. Therefore, the response must be sent to either Module c or d. However, c or d again not being directly connected to I, but more importantly not knowing where I is would again be unable to relay the response to I. The problem, therefore, is to find some routing rules which, although used locally and independently, will allow a collectivity of modules forming a path to bring the response to the source.

Before the response routing rules the general message propagation rules must be identified. As the previous discussion also indicates, upon receiving a general message a module sends it out on all out-going channels. Due to circular paths a general message would never die out. Hence, the need for a

stopping rule. We therefore impose: a general message having been received once will be ignored upon subsequent arrivals.*

In Figure 2 the propagation of a general message based on the above rules has been shown. Heavy elliptic dots mark the first arrival of the general message; subsequent arrivals have not been shown, since they are to be ignored. Note that some modules such as III in Figure 2 receive the general message simultaneously on two channels.

Defining unit time as the time it takes a message to go from one module to a neighboring one including processing time, we can study the number of modules reached by the general message as a function of time.

Referring to Figure 2 we see that in Time 1 two modules have been reached. In Time 2 four additional modules are covered. It can easily be shown by enumeration that for $t > 4$ it is:

$$tpr = \text{total number of modules covered} = 2t^2 - 2t + 1$$

where t is time that has elapsed since the initiation of the general message.

Our assertion is that only remembering the channels on which the general message first arrived is all that is needed for routing in the Square Class C Selcuk Network.

Let the channels of a Class C Module be labeled as in Figure 1. That is when going clockwise, the first input channel will be i_1, the second i_2 and similarly for the output channels.

---

*Second arrivals are important for adaptation to local malfunctions since failure to receive the second arrival of a G.M. would be indicative of failures of neighbors and such information can be used to use a different set of response routing rules.

The response routing rules then are the following:

General Message First Arrival Channels	Response Route
i_1	0_2
i_2	0_1
i_1 and i_2	0_1

For example, if the GM had first come on i_1, then the response when and if it comes is to be sent on 0_2 no matter what channel the response arrived on.

The proof is by construction. In Figure 2 consider Modules II and III. It can be easily seen that the application of the above rules will route the response messages as indicated by dotted lines. Application of the response routing rules to all the modules in the network shows a definite pattern as indicated by heavy lines the examination of which pictorially shows the convergence of response paths upon the source, Module I.

If we allow the channels to be bidirectional, the total number of modules covered by a GM becomes $2t^2$ for all t. For large t this is also the propagation in unidirectional Square Class C network. In a bidirectional network the response follows the shortest path. Call this the shortest response time. Actual response time in a Selcuk Network of one-way channels equals the number of modules along the path actually taken by response. Delay is defined as the difference between these two response times. For instance for Module II shortest response time is nine whereas actual is eleven giving a delay of two. In Figure 2 note the spiralling of response paths only in the vicinity of the source module. Elsewhere the response paths are straight. The response

delays must, therefore, be independent of the size of the network. In fact the maximum delay is 10, minimum 0 and the approximate average delays for the four quadrants starting with upper right and going clockwise are 0.5, 8.5, 4.5, 2.5, or an overall average delay of 4. When dealing with thousands of units this delay is of course negligible.

Clearly in the case of Square Class C Network three simple routing rules that are strictly based on local information and applied independently suffice for a response from any module to reach the source of a GM. Location information is entirely immaterial. Furthermore nothing would be gained by making channels two-way since propagation and response times would remain about the same. However, two-way channel networks would require much more memory and logic. The efficiency of the one-way channel network is obvious.

Hexagonal Class K Network

Figure 3 shows a Hexagonal Class K Network and the propagation of a GM. Note that: in the Class K Module input and output channels alternate; except along the principal axes (with respect to GM source) the GM arrives simultaneously on two channels; in a region defined by two adjacent axes the arrival channels remain the same; and if we connect modules that receive the GM at the same time we obtain concentric triangles. Counting the modules along triangular propagation fronts we see that the total number of modules covered is $3/2\ t(t+1)$ or for large t $3/2t^2$. Surprisingly despite a 50% increase in the number of channels propagation speed is down by about 30%.

Response routing rules are: $i_1 \rightarrow o_1$; $i_2 \rightarrow o_2$; $i_3 \rightarrow o_3$; i_1 and $i_2 \rightarrow o_2$; i_1 and $i_3 \rightarrow o_1$; i_2 and $i_3 \rightarrow o_3$. The rules can be summarized by two statements since the Class K Modules is so symmetric: 1) If the GM had first arrived on a single channel, the response when and if it comes is to be routed on the right adjacent output channel and 2) If the GM had arrived simultaneously on two channels then

the response is to be routed on the output channel between the two input channels that had carried the GM in. Figure 3 shows the application of the rules to three modules. Note the stepstone behavior of response paths at the principal axes. As to be expected from this zigzagging response delay increases with the size of the network. So compared to Square Class C, the Hexagonal Class K not only is slower in propagation but also in response routing despite a 50% increase in channels and the resultant increase in the rules, memory, and logic. Is the alternation of input and output channels to be blamed? Is making input channels adjacent important? More evidence can come by examining networks of Square Class D (input channels alternate with output channels) and of Hexagonal Class D (adjacent input channels). We will summarize the behavior of these two networks.

Square Class D

In the network constructed with Square Class D modules in which input channels alternate with output channels we observed that a) the total number of modules covered by the GM is $t(t+1)$ or for large t only t^2, clearly the slowest network so far, b) response routing is possible only if one takes into account the channel on which the response arrives in addition to GM arrival channels and c) response delays are very much a function of the size of the network. On all counts this is an inferior network.

Hexagonal Class D

The Hexagonal Class D module has all input channels adjacent. However, when a network of Class D modules is constructed it turns out that connectivity in all regions is not achievable. Consequently we constructed a network using both class D and class K modules, making class D as numerous as possible. The result was a network with three class D per class K. In this hybrid hexagonal network if one lets a class K type module initiate the GM a) the total number

of modules covered rapidly approaches $3n^2+3n$, the fastest so far and is equal to propagation in a two-way channel hexagonal network b) response routing rules that strictly depend on how the GM had arrived exist and c) response delays do not increase with the size of the network and the average is about 3/8 of a time unit, very negligible.

Clearly making input channels adjacent has dramatically increased efficiency. Even making the channels two-way will not increase speed but will appreciably increase the complexity and the logic and memory requirements.

Conclusions

We have shown that in networks properly constructed with one-way channels, while a general message (GM) is propagating a recipient module can generate a response which when routed in accordance with rules that depend strictly on local information, will get to the source of the GM. Neither the respondent nor the GM source need know where each other is located. The GM can be initiated by any module and any module can be the respondant. Furthermore some Selcuk Networks of one-way channels rapidly approach the efficiency of two-way channel networks with approximately half the memory and logic requirements. In the efficient Selcuk Networks responses follow approximately the shortest path. The channel arrangement appears to have a profound impact. Making input channels adjacent dramatically improves performance.

The salient features of Selcuk Networks are the following: a) Response routing not only is achieved without location-based addressing but also does not require a central switchboard; b) consequently if the network were sectioned and each section successively removed the remaining sections would operate as a processing network and in time can conceivably take over the specialization of the removed sections; c) if there is local failure, second and subsequent arrivals of the GM can be used to modify routing rules so as to bypass the

failed units; d) a GM can be a string of operations and its propagation can initiate local processes the results of which can then be collected as "responses" to the GM source which can be any site in the network. In some sense this is a summation or an integration process; e) some collection of the modules through response routing, can form a sub-network such that after the initial propagation of a GM the respondent modules and the source can communicate among themselves along directed pathways. This has been shown for the Hexagonal Class K Network although the results are not reported here. Note that such task partitioning of the network would allow for plasticity as well as specificity; f) finally in due time the desired response routes can be made permanent allowing the taskwise defined subnetwork to remain permanently connected and specialized in that task yet at the same time remaining a part of the total network and interacting with it.

Such features would seem to make Selcuk Networks potent for content-based retrieval in large arrays, for parallel processing machines, for large communication networks including computer nets and also for pattern recognition (25).

Figure Titles

Figure 1. Admissable channel arrangements.

Figure 2. General message propagation and response routing in Square Class C Selcuk Network.

Figure 3. General message propagation and response routing in Hexagonal Class K Selcuk Network.

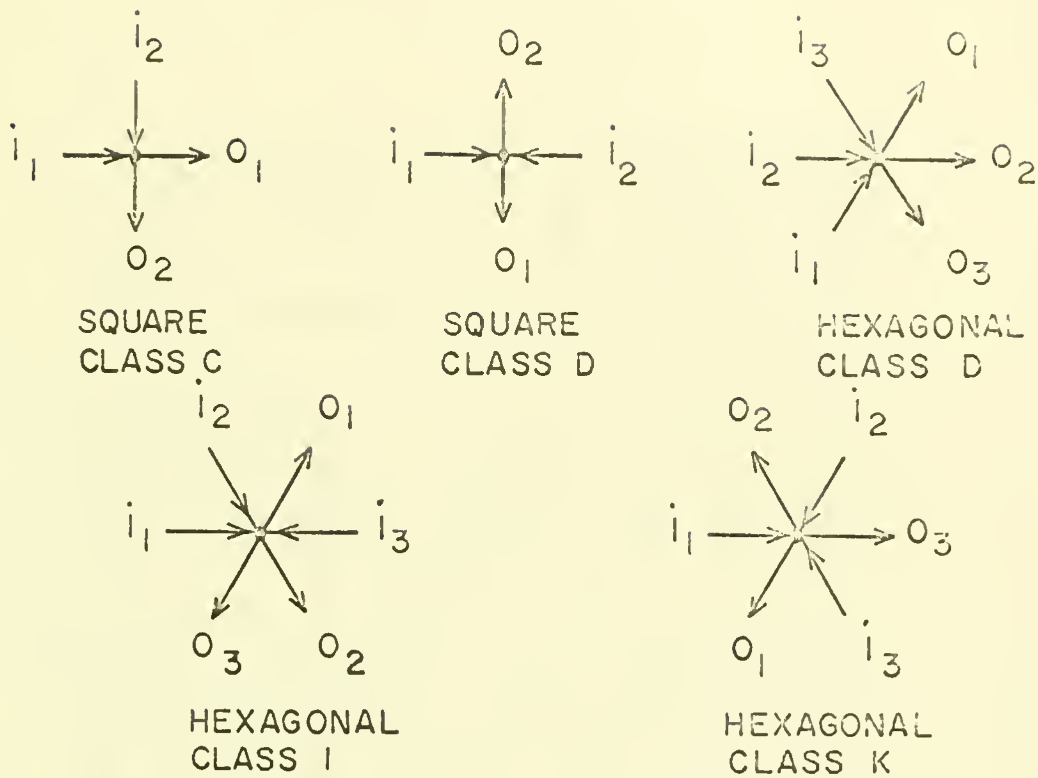
i2
i1
o1
o2
SQUARE
CLASS C
o2
i1
i2
o1
SQUARE
CLASS D
i3
o1
i2
o2
i1
o3
HEXAGONAL
CLASS D
i2
o1
i1
i3
o3
o2
HEXAGONAL
CLASS I
o2
i2
i1
o3
o1
i3
HEXAGONAL
CLASS K

Figure 1

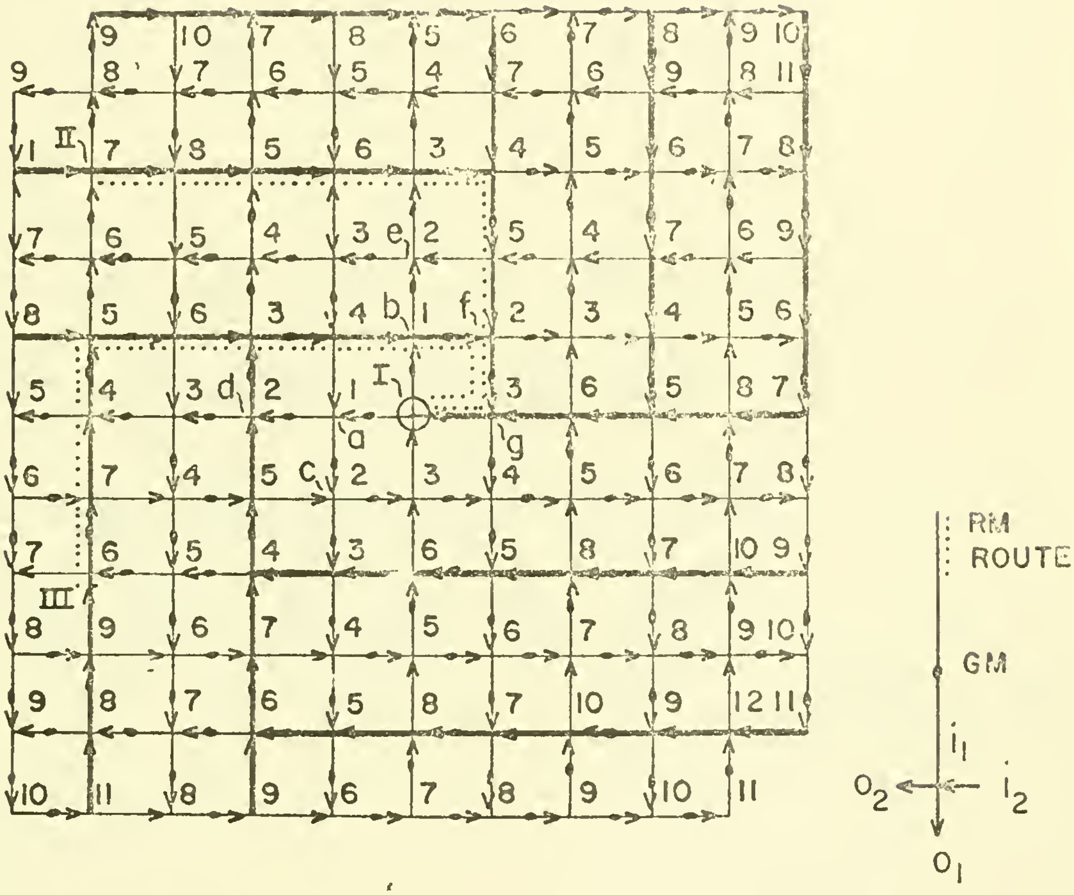

Figure 2

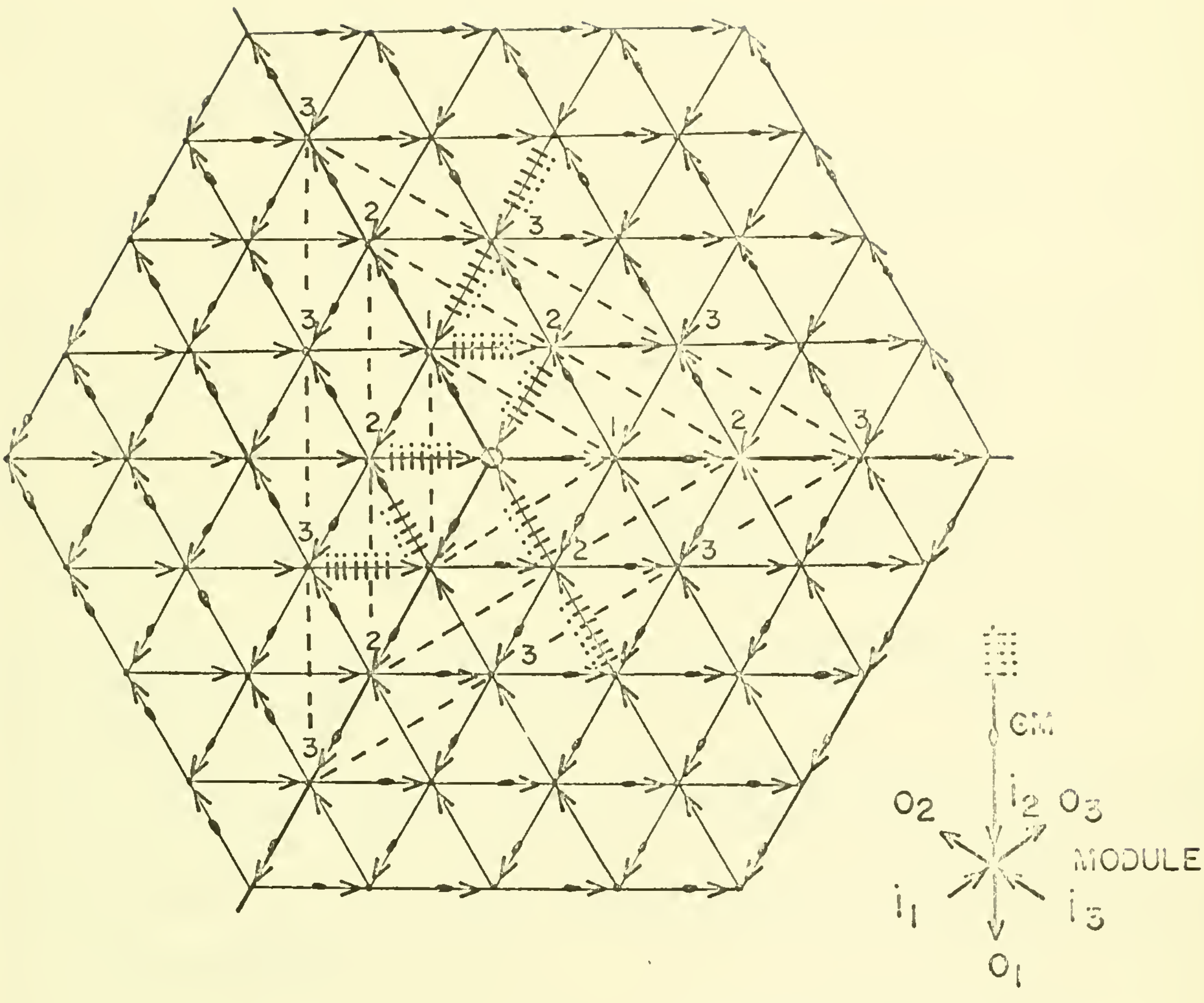

Figure 3

Kenan E. Sahin, Ph.D., Member, IEEE
Associate Professor
University of Massachusetts
Amherst, Massachusetts 01002

REFERENCES

(1) D. Aspinall et. al., "An Integrated Associative Memory Matrix" IFIP Congress Proc., August 1968.

(2) R.C.M. Barnes and I.N. Hooton, "Associative or Content-Addressed Stores" H.M. Stationary Office, England, 1966.

(3) W.J. Bouknight, et. al., "The ILLIAC IV System" Proc. IEEE, Vol. 60, No. 4, April 1972.

(4) S. Carr, S. Crocker and V. Cerf, "HOST-HOST Communication Protocol in the ARPA Network," SJCC AFIPS Conf. Proc., 1970.

(5) W.F. Chou, "Content-Addressable Memory Techniques" Sperry Rand Corp. Quarterly Progress Report, 1966.

(6) A. Corneretto, "Associative Memories," Electronic Design, 11, Feb. 1963.

(7) S. Crocker et. al., "Function Oriented Protocols for the ARPA Network," SJCC, AFIPS Conf. Proc. 1972.

(8) R.L. Davis, "The ILLIAC IV Processing Element," IEEE Trans. on Comp., VC18 No. 9, September 1969.

(9) R.G. Ewing and P.M. Davies, "An Associative Processor," FJCC AFIPS Conf. Proc., 1964.

(10) D.J. Farber and K. Larson, "The Structure of a Distributed Communication System," presented at Symp. Comp. Comm., April 4-6, 1972.

(11) D.J. Farber and K. Larson, "The Architecture of a Distributed Computer System - An Informal Description," Dept. Inform. Comp. Sci., Univ. of Calif., Irvine, Tech. Rep. 11, 1970.

(12) H. Frank and W. Chou, "Topological Optimization of Computer Networks" Proc. IEEE Vol. 60, No. 11, November 1972.

(13) H. Frank, I.T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network: SJCC, AFIPS Conf. Proc., 1970.

(14) H. Frank, R.E. Kahn, and L. Kleinrock, "Computer Communication Network Design - Experience with Theory and Practice," SJCC, AFIPS Conf. Proc., 1972.

(15) R.H. Fuller, "Associative Parallel Processing," SJCC, AFIPS Conf. Proc., 1967.

(16) F. Heart, et. al., "The Interface Message Processor for the ARPA Computer Network" SJCC, AFIPS Conf. Proc., 1970.

(17) ILLIAC IV Research Documents Abstracts, ILLIAC IV Project, Univ. of Illinois at Urbana, Urbana, Ill., July 1, 1971.

(18) R.E. Kahn, "Resource-Sharing Communication Networks" Proc. IEEE Vol. 60, No. 11, November 1972.

(19) C. Lee, "Content Addressable and Distributed Logic Memories," in Applied Automata Theory Tou (Ed.) New York: Academic Press, 1968.

(20) B.I. McKeever, "Associative Memory Structure" FJCC, AFIPS Conf. Proc. Vol. 28, 1965.

(21) J. Minker, "An Overview of Associative or Content-Addressable Memory Systems and a KWIC Index to the Literature: 1956-1970" Computing Reviews, Vol. 12, No. 10, October 1971.

(22) K. Nakano, "Association - A Model of Associative Memory" IEEE Trans. on Sys., Man, and Cyb., Vol. SMC-2, No. 3, July 1972.

(23) L.G. Roberts and B. Wessler, "The ARPA Computer Network Development to Achieve Resource Sharing," SJCC, AFIPS Conf. Proc., 1970.

(24) K.E. Sahin, "Intermodular Communication Without Addressing in Planar Arrays," Ph.D. Thesis, M.I.T., February 1969.

(25) K.E. Sahin, USP Application, April 1969.

(26) D.C. Slotnick, "The Fastest Computer," Sci. Amer., Vol. 224, No. 2, February 1971.

(27) D.C. Walden, "A System for Interprocess Communication in a Resource Sharing Computer," Comm. ACM, Vol. 15, No. 4, April 1972.

BASEMENT

Date Due

~~NOV 7 '75~~		
~~DEC 16 '75~~		
APR 14 '77	'89	
FEB 13 '78	APR 15 1994	
JAN 10 '80	1997	
DEC 07 '81	MAY 10 1997	
AP 14 88	AUG 07 2002	

Lib-26-67

MIT LIBRARIES
3 9080 003 827 638

MIT LIBRARIES
3 9080 003 796 577
663-73

MIT LIBRARIES
3 9080 003 796 593
664-73

MIT LIBRARIES
3 9080 003 796 619
665-73

MIT LIBRARIES
3 9080 003 796 635
666-73

MIT LIBRARIES
3 9080 003 796 650

MIT LIBRARIES
3 9080 003 796 676
668-73

MIT LIBRARIES
3 9080 003 796 692
669-73

MIT LIBRARIES
3 9080 003 796 718
670-73

MIT LIBRARIES
3 9080 003 827 653
671-73

MIT LIBRARIES
3 9080 003 827 679
672-73

MIT LIBRARIES
3 9080 003 827 695
672-73A