MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC

Computation Structures Group Memo No. 42

n-server m-user Arbiter

Suhas S. Patil

May 1969

Introduction

In a multiprocessing environment it is often advantageous to arrange similar resources into a pool from which the resources may be borrowed as needed[1,2,3,4]. For example n identical processors may be pooled together to be used by m distinct processes; the need of a process in this case can be met by any of the processors in the pool. When a process needs a processor it requests for one and if a processor in the pool is free, it is assigned to the process. The process uses the processor and returnes it to the pool when it is done with its use. In the above example the processors are servers and the processes are users. The act of assigning servers to users is called arbitration.

Some highly parallel computers[2,3] have functional units organized into pools of resources. In computers of this kind arbiters are important components because multiplexing is integral part of their operation. Furthermore, if possible, the arbiters should be realized in terms of macro-modular units[5] for the computers themselves may be built this way. Another advantage of the macro-modular approach is that an arbiter of arbitrary size can be built this way using a few types of modules.

The n-server m-user arbiter presented here is asynchronous and macro-modular. It is constructed from a set of macro-modules introduced here. The macro-modules are inter-connected through C-links (composite links). The C-links are like the control links introduced earlier in connection with the macro-modular design of the asynchronous circuits[5] except that C-links have more wires and are capable of concurrently carrying two distinct communications.

It should be pointed out that the solution to n-server m-user arbitration is not the end to the problem of arbitration. The solution to presented here handles a set of identical servers, but there may be situations where it may be desirable to pool together some servers which are not identical. A pool of non-identical servers is particularly interesting when the capabilities of the servers overlap. For example servers of type 1 could perform operation A, servers of type 2 could perform operation B and servers of type 3 could perform both operation. Then a user who wants to perform operation A could use servers of either type 1 or type 3 and a user who wants to perform operation B could use servers of type 2 or type 3.

The problem becomes more interesting and difficult when individual requests for multiple servers are permitted. A more general arbiter than the one presented here is needed to solve this problem.

Even though the n-server m-user arbiter presented here does not solve the general problem it is an important step in that direction.

Composite Links

The C-links form the links of communication between the macro-modules introduced here. The function performed by a C-link is analogous to that of the control links introduced earlier in connection with the macro-modular design of the asynchronous circuits[5]. The C-links are, however, more complex; they have more wires and can concurrently carry mare information (figure 1). A C-link has two sub-links: i) activation link and ii) block

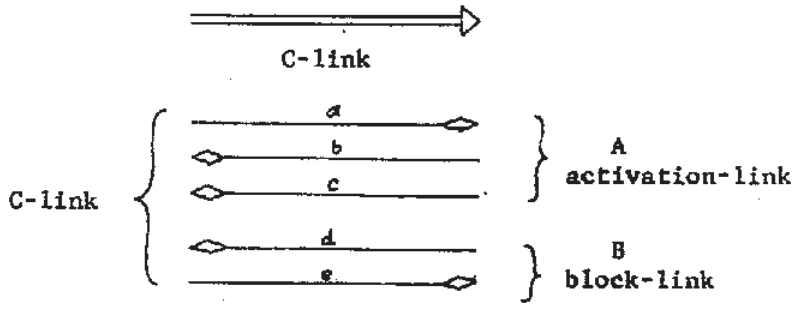
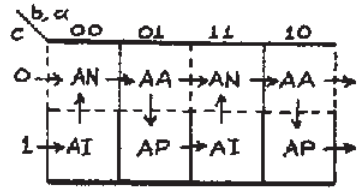
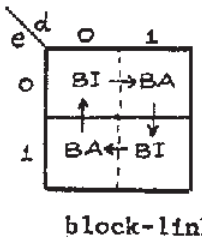


Figure 1 A Composite Link



the alternative set

{AN,AA,AP,AI}



{BA,BI}

Figure 2 The relation between the levels on the wires and the conditions of the sub-links of a C-link

link. These sub-links together form a C-link; they do not have separate existance. Associated with each sub-link is an alternative set which gives the possible conditions the sub-link can be in. At any given time a sub-link is in one and only one condition in its alternative set. The relation between the signals on the wires and the conditions on the sub-links is given in figure 2. In the Karnaugh map of figure 2 the dotted boundaries represent the transitions caused by the module on which the C-link is incident and the light boundaries represent the trandition caused by the module from which the C-link emerges. No transitions are possible across the bold boundaries. The constraints on the changes in the conditions of the sub-links, given in figure 2, can be expressed more elegantly through the Petri-nets[6] shown in figure 3. The condition of a C-link is a tuple giving the conditions of its two sub-links. The different conditions of a C-link have the following interpretation.

The sub-link A, called activation link, forms the core of a C-link. The alternative set for this cub-link is {AN, AA, AP, AI}. AN represents incative and/or negative reply condition. It is also the initial condition of the sub-link. A request for resource is made by changing the activation link from AN to AA through transition ℓ (figure 3). The action of changing the condition of the activation sub-link of a C-link from AN to AA is called activation of the C-link. An activation represents a request for a resource and the response to the activation may be either positive or negative. A positive response occurs through transition m (figure 3) which changes the condition of the activation link to AP. A

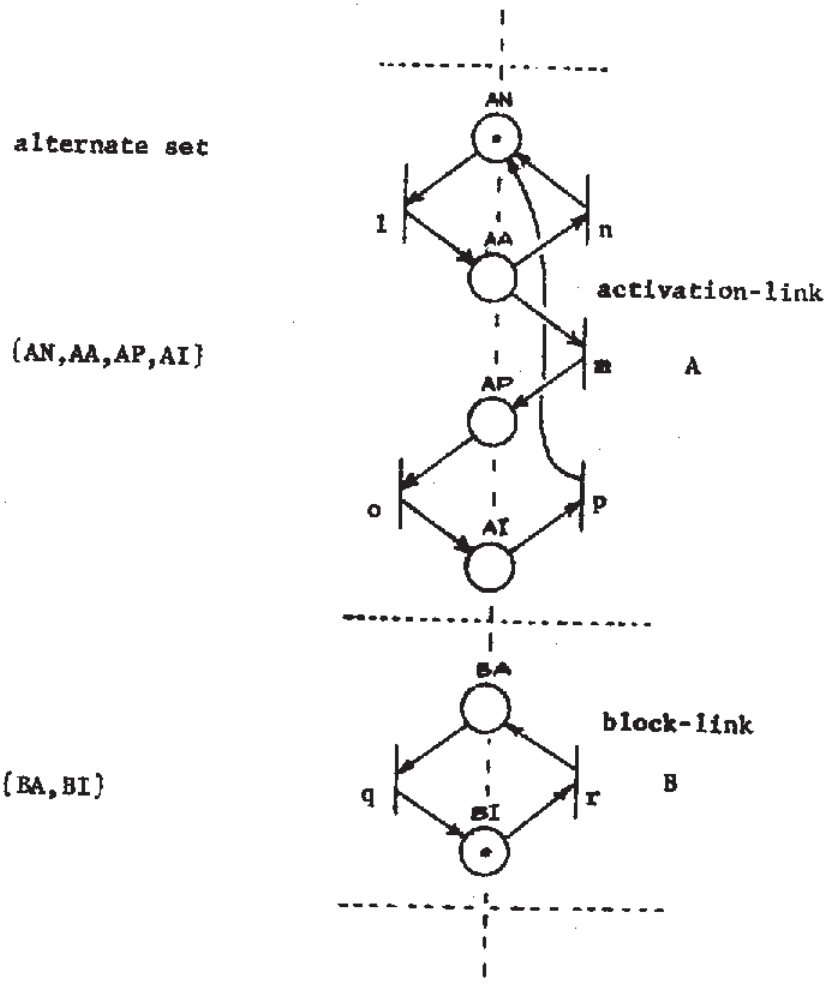


Figure 3 The Alternate Set for the Sub-links of a C-link

negative response occurs through transition n which immediately returnes the link to the inactive condition AN.

The condition that results through a positive response indicates that the requested resource has been granted to that link. On completing the use of the resource the user changes the condition of the link from AP to AI through transition o. The resource is then released and the link is returned to the inactive condition AN through transition p.

The alternative set for the sub-link B (block link) is {BA, BI}. BA and BI refer to the block active and block inactive conditions respectively. The condition BA indicates that the resource has been granted to some other C-link. This information is valuable to some macro-modules as will be seen later.

Elementary Arbiter

An elementary arbiter is an elementary macro-modular unit from which larger arbiters can be constructed. It has two incident C-links and one emerging C-link. The function of an elementary arbiter is to logically connect the emerging C-link to an active incident C-link when any of the incident C-links are activated. A logical connection between two links means that the changes in the condition of one link are transferred to the other. This connection continues until the incident link connected to the emerging link reaches the condition AI. The connection is then broken and a new connection is established possibly to the other incident link (if it is active). The act of connecting the emerging link to an

incident link is interpreted as granting access to that incident link. When an elementary arbiter has granted access to an incident link and that incident link is making use of that access, the requests on the other incident link are answered negatively. The detailed operation of an elementary arbiter is explained below with the help of the Petri-net of figure 4.

When one of the incident links is activated, say link 1, the elementary arbiter acts through transition c_1. With this transition the elementary arbiter is engaged by link 1. If both incident links become cative simultaneously the arbiter arbitrarily picks the link it wants to be engaged by. During the time an elementary arbiter is engaged by one incident link say link 1, the only possible immediate response to the activation of the other link is a negative answer through transition b_2. The following actions follow the engagement of the arbiter by link 1.

Transition c_1 causes transition i and transition i activates the emerging link. The activation of the emerging link may be answered either positively or negatively. A change in the condition of the emerging link from AA to AP corresponds to a positive answer while a change from AA to AN corresponds to a negative answer. A positive answer on the emerging link is passed to link 1 through transitions j and d_1, and a negative answer through transitions g and a_1. In case of a negative answer the elementary arbiter is immediately reset by transition n. A positive answer returned on link 1 is a signal that the resource has been granted to the user corresponding to link 1. The user uses the resource and signals the completion by

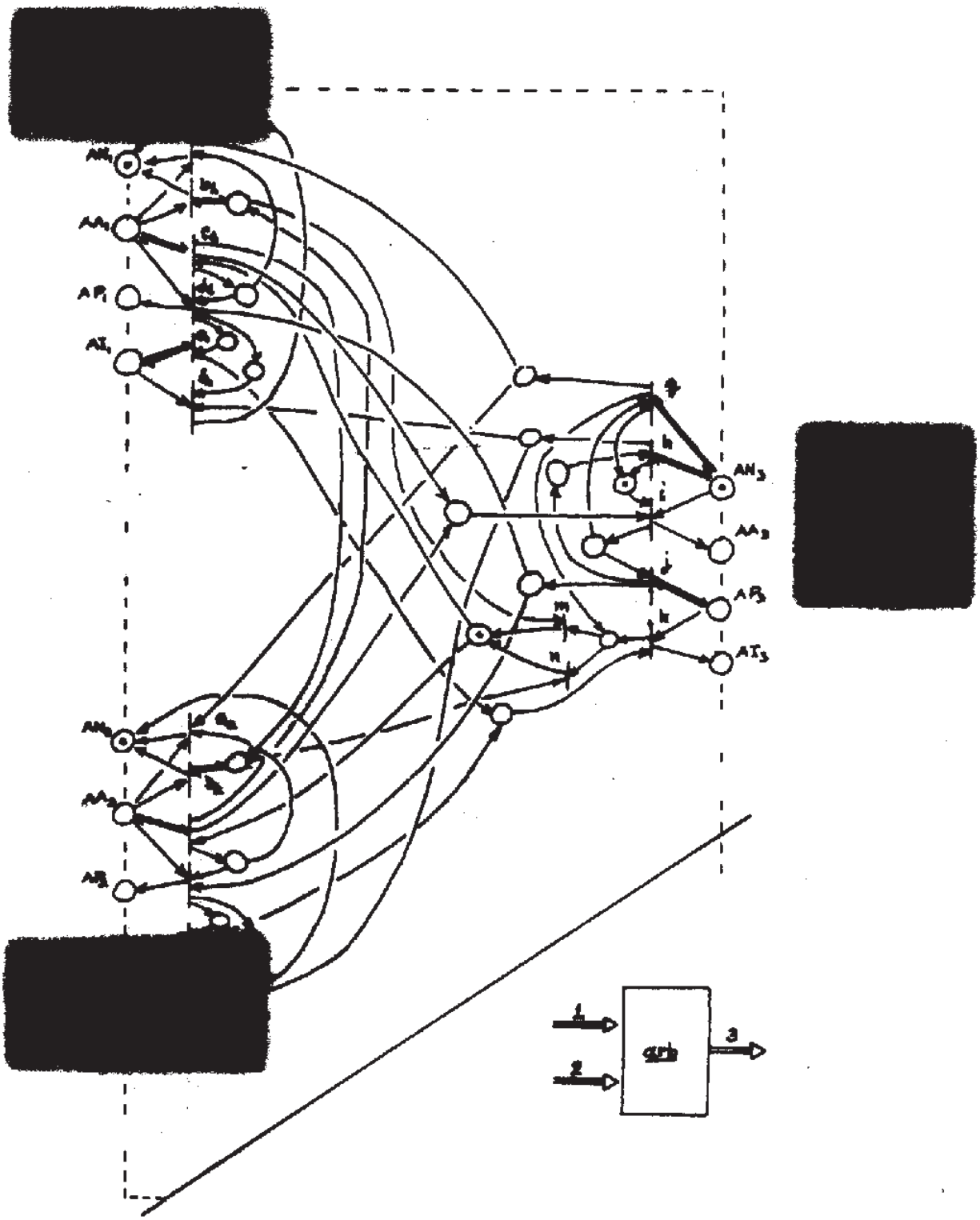
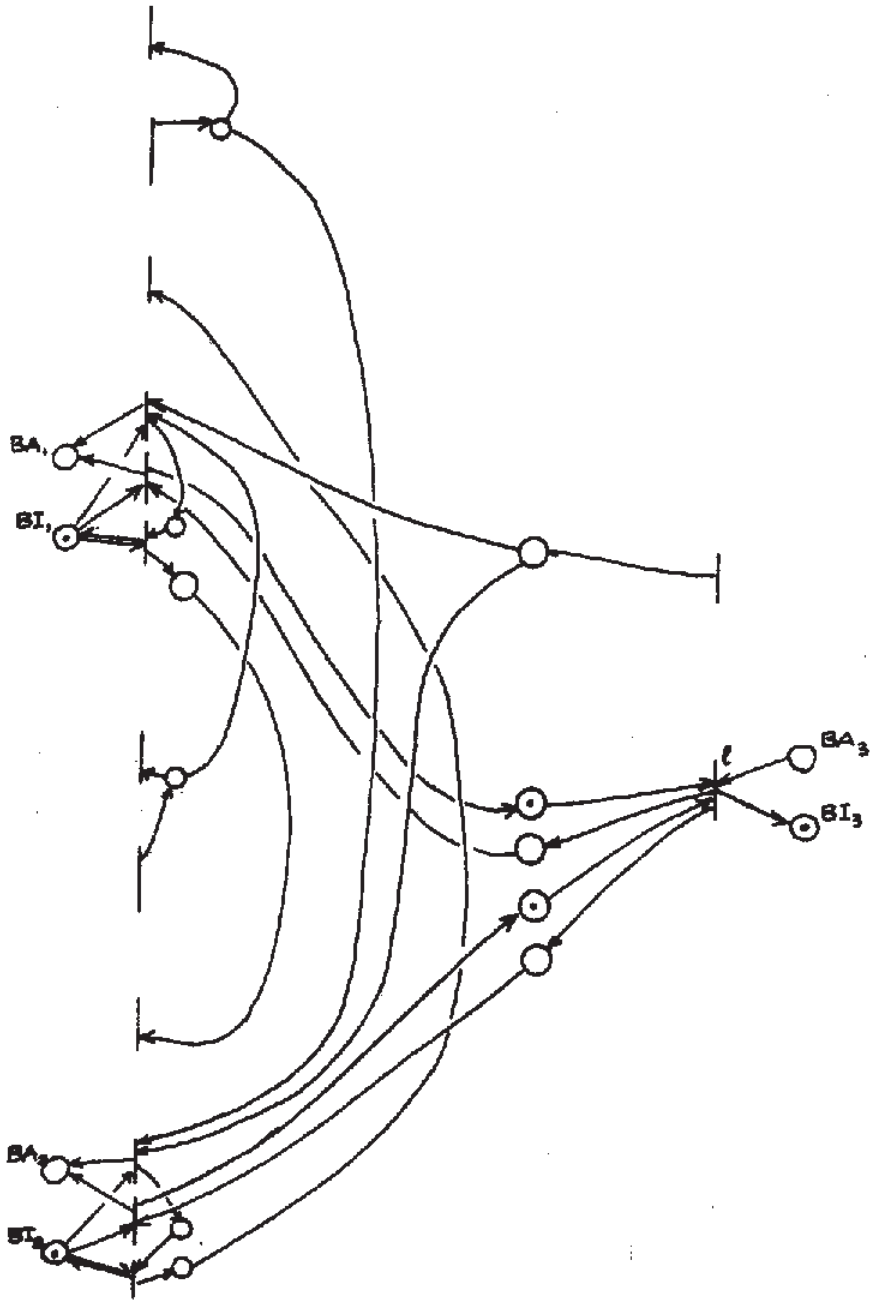


Figure 4 A Petri-net for the Elementary Arbiter


BA1
BI1
BA3
BI3
BA2
BI2

changing the condition of the link to AI. The elementary arbiter responds to this by changing the condition of the emerging link to AI through transitions e_1 and k. This releases the resource and the emerging link returns to the condition AN. In response to this the elementary arbiter turns link 1 inactive by changing its condition to AN through transition f_1 and resets itself through transitions h and n. The elementary arbiter is now ready to be engaged once again.

The function of the block link in relation to the operation of an elementary arbiter is as follows. Activation of the block sub-link of an incident C-link of an elementary arbiter (by the elementary arbiter) indicates that either the other incident link of the arbiter has been granted the resource or that a block signal has been received on the emerging link of the elementary arbiter. In any case a block signal indicates that the resource has been granted to some other link. This information is important for the proper operation of the n-server m-user arbiter presented later. The need for such information arises because many servers may respond to the request from a user; in this case the user picks the first server and informs the rest that their services, even if granted, are no longer needed. Note that a block signal propagates in direction opposite that of the C-link.

Termination Module

A termination module (term) is used to terminate a C-link. A Petri-net for this module is shown in figure 5. The function of a term module is to reply positively to each activation of the incident C-link and to return

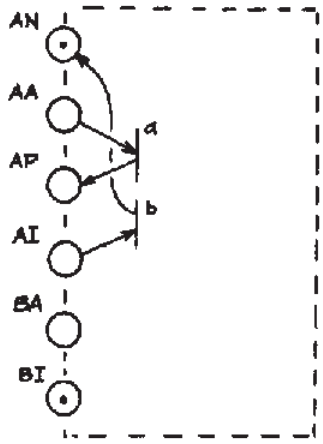


Figure 5 A Petri-net for the term module

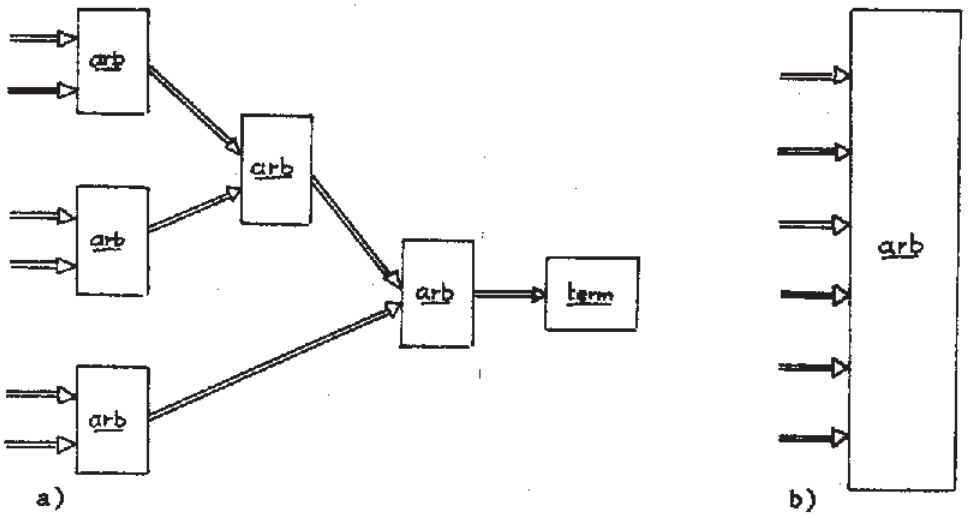


Figure 6 A Composite Arbiter

the C-link to the inactive condition AN when the C-link condition becomes AI.

The term module should more appropriately be called a positive-term as it always answers positively. A negative -term would be one which always answers negatively. Since a negative-term is not required in the construction of an n-server m-user arbiter, only the positive-term is prenented and is called term.

Composite Arbiter

A composite arbiter is an arrangement of elementary arbiters and a term module as shown in figure 6a. The difference between an elementary arbiter and a composite arbiter is that an elementary arbiter has only two input links while a composite arbiter has many input links. An advantage in constructing a composite arbiter from the elementary arbiters is that a composite arbiter of arbitrary size can be built this way. A schematic representation for a composite arbiter is given in figure 6b.

Transform Module - T

A transform module provides an interface between the ordinary control links and the C-links. A transform module has three control links and one C-link. A request for a resource is sent to a transform module on control link 1 (figure 7). The transform module passes the request to the C-link by making it active. In case of a positive answer on the C-link the transform module activates control link 2. Activation of control link 2

indicates that the resource has been granted. The user then uses the resource and signals the completion by returning an acknowledge signal on control link 2. In respose to the acknowledge signal the transform module changes the C-link condition to AI to signal that the resource is no longer required. This starts the action to release the resource. When the resource is released the C-link condition changes to AN. Following this the transform module returns an acknowledge signal on control link 1 to signal the completion of the above actions.

On the other hand, in case of a negative answer the transform module first activates control link 3 and on receiving an acknowledge signal on control link 3, returnes an acknowledge signal on control link 1.

The detailed operation of a transform module is explained below with the help of the Petri-net figure 7.

The external circuit requests resource by activating control link 1. On activation of control link 1 transitions a and i activate the C-link. If the C-link condition changes to AP (indicating a positive answer) transitions j and c activate control link 2. This is a signal to the external circuit that the resource has been granted. The external circuit uses the resource and signals the completion by returning an acknowledge signal on control link 2. The transform module then, acting through transitions d and k, puts the C-link in condition AI. This releases the resource and the C-link condition changes to AN. In response to this the transform module returns an acknowledge signal on control link 1 through transitions h and b.

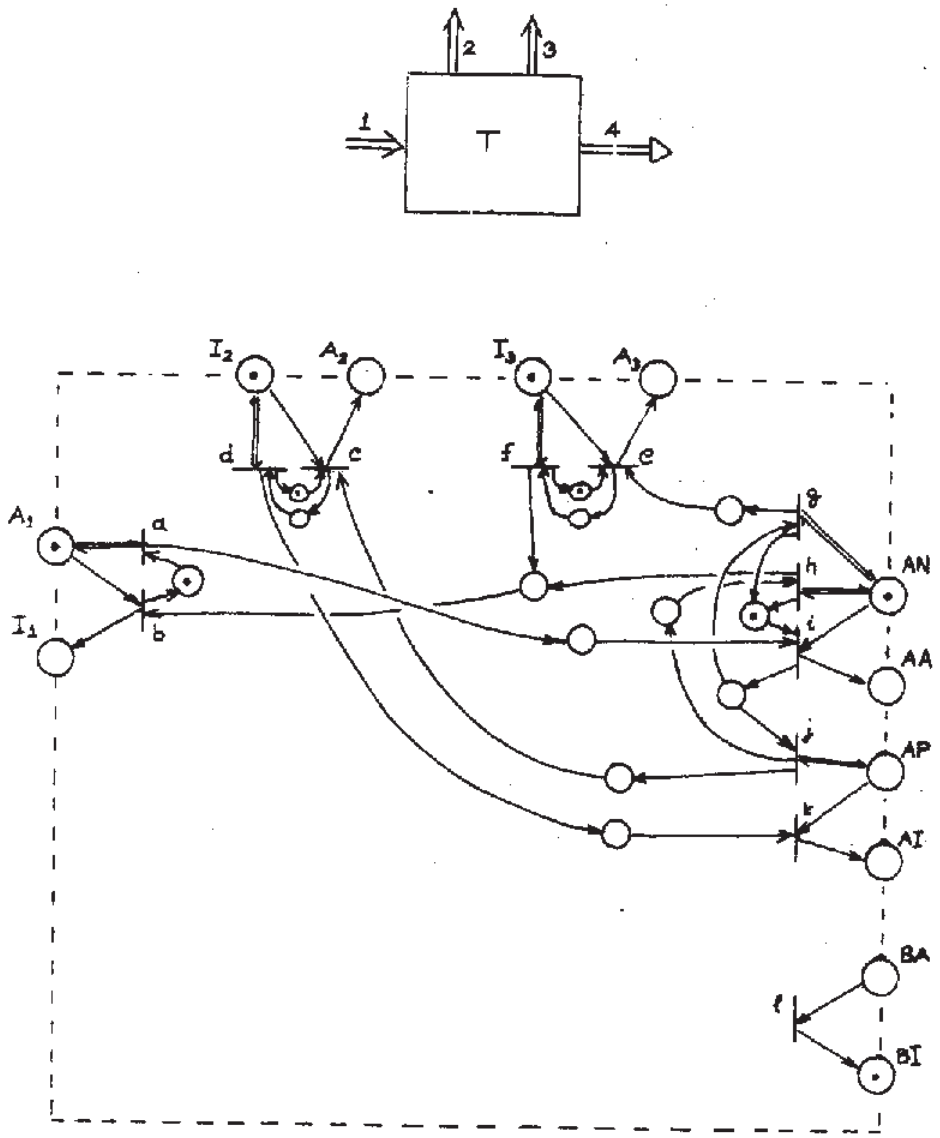


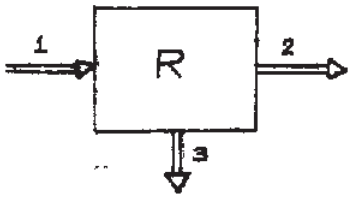
Figure 7 A Petri-net for the Transform Module

In case the C-link condition changes to AN (a negative answer) the transform module activates control link 3 through transitions g and e, and when the external circuit acknowledges this activation, the transform module returns an acknowledge signal on control link 1 through transition b and returnes to its initial condition.

Repeater Module - R

A repeater module has one incident C-link and two emerging C-links. The function of a repeater module is to repeatedly ask for a resource on link 2 (figure 8) until either the resource is obtained or a busy signal is received on link 3. The detailed operation of a repeater module is explained below with the help of the Petri-net of figure 8.

A request for a resource is made by activating link 1. When this occurs, transition b places a stone at place x, and a stone at place x causes transition i which in turn activates link 2. A negative answer to this activation leads to transition g. Transition i is then enabled and through transition i link 2 becomes active once again. this process continues till either a positive answer is obtained or the stone at place x is removed by transition o following a block signal on link 3. On the other hand, a positive answer to the the activation of link 2 leads to the activation of link 3 through transitions j and r. If the answer to the activation of link 3 is positive, a positive answer is returned on link 1 to indicate that the resource has been granted. The repeater module then waits until link 1 signals the completion of the



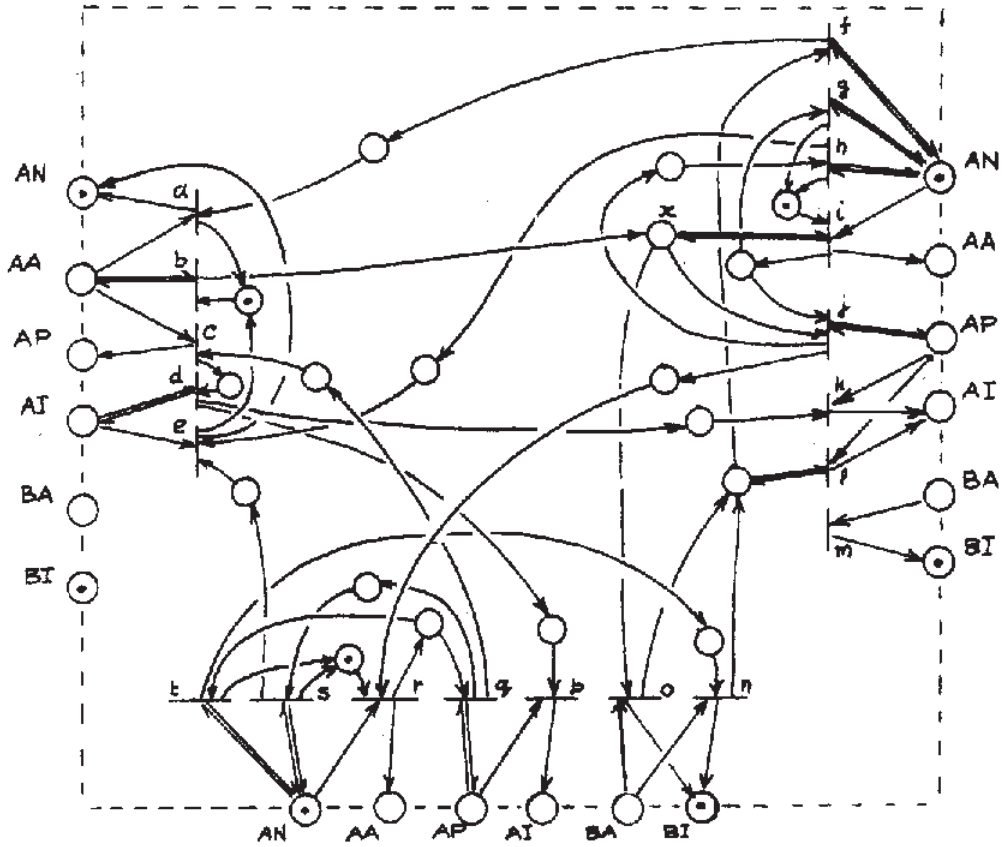


Figure 8 A Petri-net for the Repeater Module

use of the resource by changing link 1 condition to AI. In response to this change in the condition of link 1, the repeater module changes the conditions of the links 2 and 3 to AI through transitions d, p and k. The resource is then released and links 2 and 3 conditions change to AN. Following this the repeater module changes link 1 condition to AN to signal the completion of the above actions.

If the answer to the activation of link 3 is negative, a block signal must have either already arrived on link 3 or will soon arrive on link 3. Transition t follows the negative answer, and when the block signal comes transition n deactivates the block link and causes transition ℓ to occur. Transition ℓ changes the condition of link 2 to AI to release the resource for it is not required (as is indicated by the negative answer and the block signal on link 3). The resource is then released and link 2 condition is changed to AN. Following this the repeater module changes the condition of link 1 to AN .

A block signal may arrive on link 3 even while the repeater module is trying to obtain a resource on link 2. In this case the stone at place x is removed by transition o. If link 2 is inactive when the transition o occurs, the repeater module is forced to immediately give up the search for the resource and return a negative answer on link 1. If link 2 is active when the transition o occurs one of two things may happen: either condition of link 2 may change to AP or to AN. If the condition of link 2 changes to AP, transition ℓ changes its condition to AI and following condition AI, link 2 condition changes to

AN. In any case link 2 condition eventually becomes AN. When link 2 condition becomes AN the repeater module returnes a negative answer on link 1 through transitions f and a.

In any case the affect of a block signal on link 3 is to force the repeater module to give up the search for a resource and to return a negative answer on link 1.

Single Server m-user Arbiter

A single server m-user arbiter is simple compared to the n-server m-user arbiter and understanding the operation of a single server m-user arbiter will help understand the operation of the n-server m-user arbiter.

Figure 9 shows a macro-modular circuit for a single server m-user arbiter. In that circuit user 1 through user m share operator f. Sharing use of the operator involves a request for the use of the operator, moving the inputs to the operator, the application of the operator to the inputs and moving the results to the user. The arbiter thus has two parts: one that decides which user should use the operator in case of conflicting requests and the other for moving data to and from the operator. The first part is the control structure of the arbiter and the second part is the data structure of the arbiter. The most important and interesting part of an arbiter is its control structure.

The data structure of the single server m-user arbiter is implemented using the macro-modules introduced earlier in cinnection with the macro-modular design of the asynchronous circuits[5]. The control-structure,

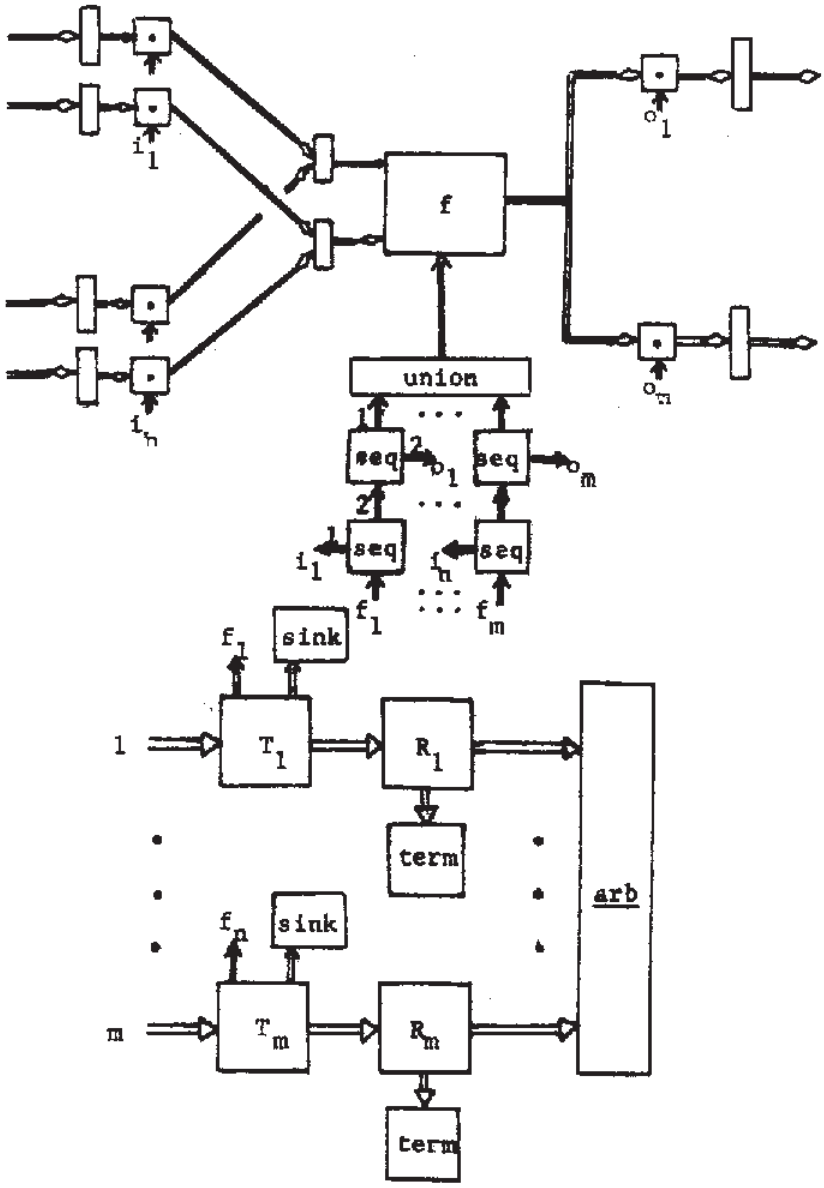


Figure 9 Single server M-user Arbitration

however, needs the macro-modules introduced in this paper. The operation of the single server m-user arbiter is explained below with the help of figure 9.

A user requiring the use of the operator sends a request to the arbiter by activating the corresponding control link i.e., user i sends a request by activating control link i. When control link i becomes active, transform module T_i sends a request for the resource (the operator f) to repeater module R_i by activating the C-link interconnecting them. The repeater module then starts the task of obtaining the operator for user i. To do this the repeater module R_i sends a request for the operator to the composite arbiter by activating the C-link interconnecting them. If the arbiter replies negatively (because the operator is being used by some other user) the repeater module sends a fresh request to the arbiter. This continues till the arbiter, by returning a positive answer, grants the operator to user i. When the arbiter grants the operator this way, repeater R_i activates link 3. Since link 3 is terminated by a term module it receives a positive answer in response to the activation. (Note that link 3 on the repeater module does not perform any useful function here, but it will in the case of an n-server m-user arbiter.)

On receiving a positive answer from the term module, repeater module R_i returns a positive answer to the transform module T_i. Transform module T_i then sends a ready signal on control link f_i. In the data-structure a ready signal on control link f_i causes operator f to apply on behalf of user i as is explained below.

The ready signal on control link f_i first causes the identity operators i_i to place the inputs from user i into the input registers of the operator. The ready signal then reaches operator f through the union module and causes it to operate on the inputs from user i. After the operator application the identity operator o_i returns the result of the operation to user i. When this is done the seq module returns an acknowledge signal on control link f_i.

When transform module T_i receives an acknowledge signal on control link f_i it turns the C-link condition to AI in order to release the operator. Condition AI propagates to the composite arbiter through the repeater module R_i. The arbiter is then released and the C-link condition is changed to AN. The C-link condition propagates to the transform module T_i and the composite arbiter goes on to serve some other user (if there is one waiting).

When the C-link condition AN reaches the transform module T_i, T_i returnes an acknowledge signal to user i to inform that the requested operator application has been completed.

n-server m-user Arbiter

The n-server m-user arbiter presented here is for a pool of n functionally identical operators. Since the operators are functionally identical, a request from a user can be met by any of the operators in the pool.

Just as the single server m-user arbiter, this arbiter has

two parts: i) a control-structure and ii) a data-structure. The control-structure of the arbiter is for deciding which operator should serve which user and the data-structure is for moving the inputs to the operators and returning the results to the users. The operation of an n-server m-user arbiter is explained below with the help of the 2-server 3-user arbiter shown in figure 10a and 10b.

The data structure of the arbiter is such that a ready signal on control link f_{ij} transfers the inputs from user i to operator j, applies operator j to the inputs and returns the results to user i. When the above action is completed an acknowledge signal is returned on the control link f_{ij} to signal the completion.

The control-structure of the arbiter has the following form. There is an arbiter associated with each server (operator) and there is an arbiter associated with each user. The arbiter associated with a server has one C-link from each user (a_{ij} is the C-link from user i to the arbiter corresponding to server j). C-link a_{ij} originates at transform module T_{ij} and reaches the arbiter corresponding to server j through repeater module R_{ij}. The transform modules $T_{i's}$ correspond to user i. The control links reach these modules from user i through a fan out of wyes. The link 3's from the R modules associated with user i go to the arbiter associated with user i. The operation of the control-structure is as follows.

A request for an operator is sent on control link i by user i. The request is simultaneously sent to all servers through the fan-out of wyes.

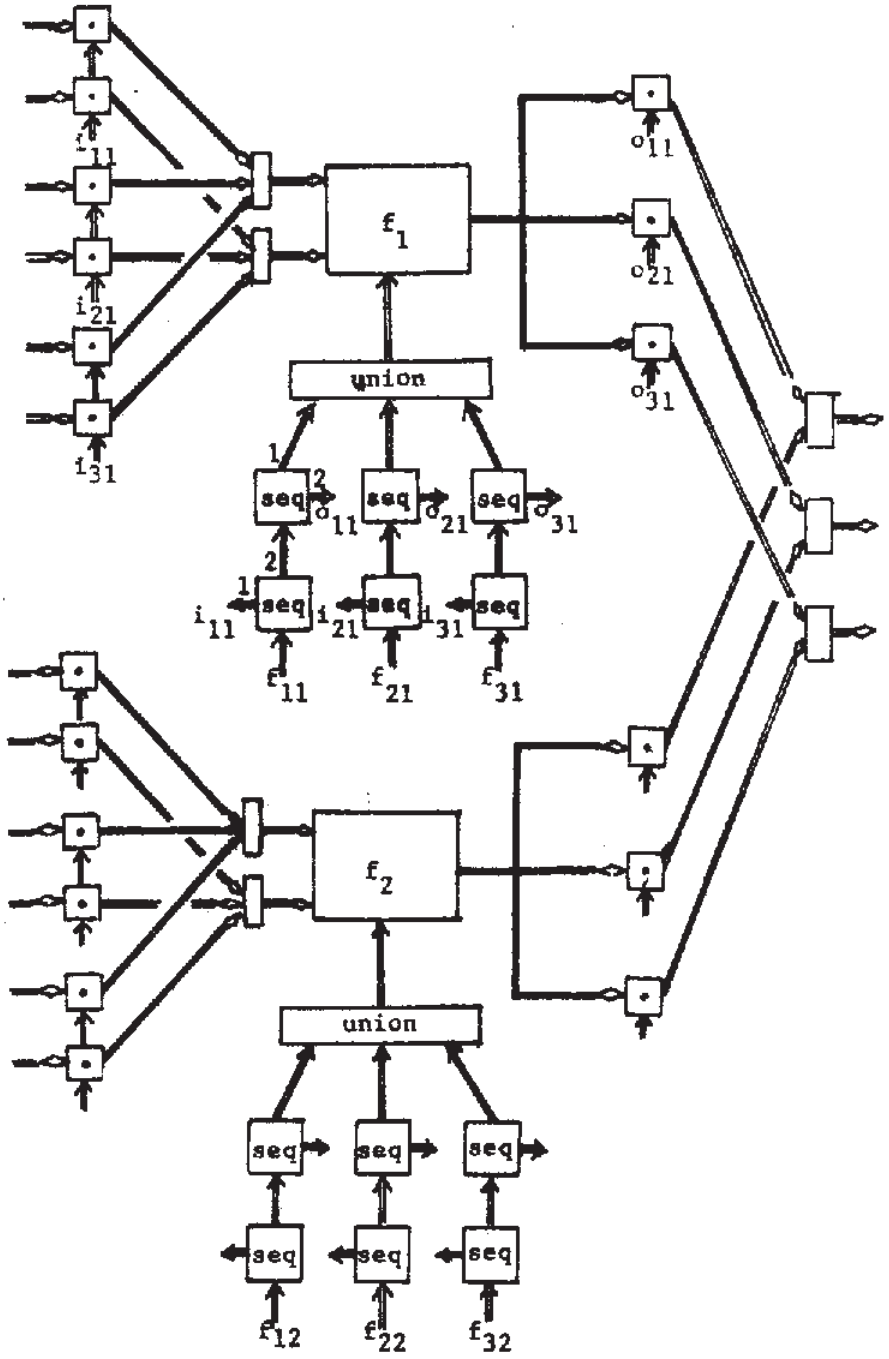


Figure 10a The Data-structure of the 2-server 3-user Arbiter

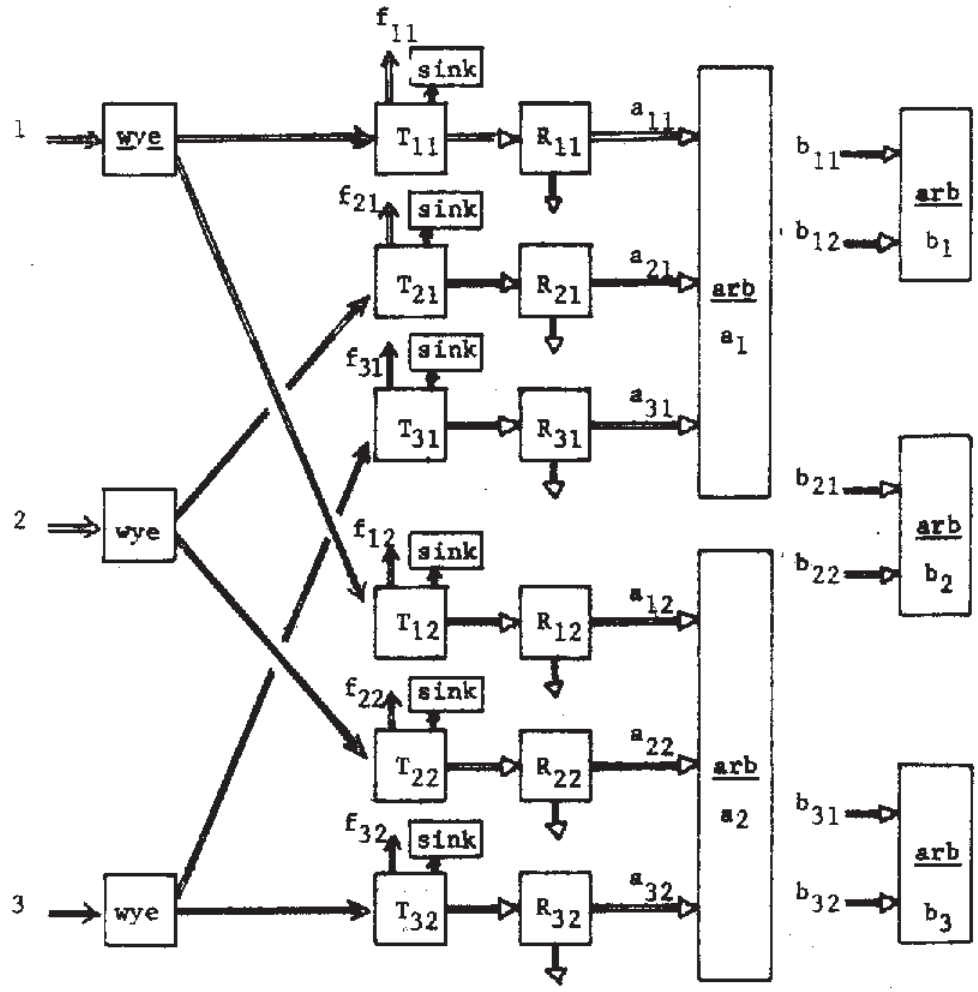


Figure 10b The Control-structure of the 2-server 3-user Arbiter

On receiving a signal on the incident control link each T_{ij} activates the C-link going to R_{ij}. On activation of the incident C-link each R module starts the task of obtaining the corresponding server for the user. For this the repeater module R_{ij} activates the C-link going to arbiter j. If the arbiter answers negatively, repeater module R_{ij} renews its request for server j by reactivating C-link a_{ij}. When arbiter a_j answers positively to an activation of a_{ij}, repeater module R_{ij} activates C-link b_{ij}. The activation of C-link b_{ij} tells arbiter b_i (the arbiter corresponding to user i) that server j is ready to serve user i. If arbiter b_i has not been engaged by some other server who offered its services, it is engaged by C-link b_{ij}. Arbiter b_i then sends a positive answer on C-link b_{ij} and block signals on the other C-links. A block signal on C-link b_{ik} tells the repeater module R_{ik} that it should give up search for operater k as one operator has already been obtained by the user, and if it has already obtained operator k it should release it immediately.

On receiving a positive answer on C-link b_{ij} repeater module R_{ij} returns a positive answer on the C-link coming from module T_{ij}. Transmit module T_{ij} then sends a ready signal to the data-structure on control link f_{ij}. The ready signal on control link f_{ij} applies operator f_j on behalf of user i as explained earlier. When the application is completed an acknowledge signal is received on control link f_{ij}. Transmit module T_{ij} then starts the action to release operator f_j by changing the C-link condition to AI. Repeater module R_{ij} then changes the conditions of C-links a_{ij} and b_{ij} to AI. Condition AI on C-link a_{ij} releases arbiter a_j

i.e., the server j, and condition AI on C-link b_{ij} releases arbiter b_i. The arbiters change the conditions of a_{ij} and b_{ij} to AN as they are released. Repeater module R_{ij} then changes the condition of the incident C-link to AN and finally transform module T_{ij} returns an acknowledge signal to user i on the incident control link. The acknowledge signal tells user i that the requested operation has been performed.

The structure of the n-server m-user arbiter is such that requests from more than one user can be processed concurrently. The ability to concurrently process more than one request is very important for the efficiant operation of an n-server m-user arbiter when n and m become large.

Since neither the form of the structure nor the modules used in constructing the n-server m-user arbiter depend on the number of users or the number of servers, arbiters of any size can be built using the modules introduced here.

BIBLIOGRAPHY

1. Malhtra, Ashok "Asynchronous Control of Computer Operations", Sloan School of Management, S. M. Thesis, February, 1967.

2. Patil, Suhas S. "An Abstract Parallel-processing System", Department of Electrical Engineering, M.I.T., S. M. Thesis, June, 1967

3. Dennis, Jack B. "Programming Generality, Parallism and Computer Architecture", Project MAC, Computation Structures Group Memo No. 32.

4. F. J. Corbato and V. A. Vyssotsky, "Introduction and Overview of the Multics System", FJCC 1965.

5. Patil, Suhas S. "Macro-modular Design of Asynchronous Circuits", Project MAC, Computation Structures Group Memo No. 41, May 1969.

6. A. W. Holt, R. M. Shapiro, H. Saint and S. Warshall, "Final Report for the Information System Theory Project", Applied Data Research, Inc., February 1968 ADR Ref. no. 6608