THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 7, September 1981
Printed in U.S.A.

A First-Come-First-Serve Bus-Allocation Scheme Using Ticket Assignments

By D. K. SHARMA and S. R. AHUJA

(Manuscript received November 13, 1980)

This paper describes a new scheme for allocating a data bus on a first-come-first-serve (FCFS) basis. When the devices connected to the bus request to become the bus-master, they are assigned distinct "ticket numbers" in the order in which the requests are generated, at which time they go into a wait state. When the bus is released by a device holding the ticket number n, it is then allocated to the device holding the ticket number n + 1. We discuss the conditions under which the scheme is a close approximation to the ideal FCFS scheme and evaluate its performance using simulation results. We also present two alternative hardware implementations of this scheme—one centralized and the other distributed. Because of its simple hardware implementation, the scheme is attractive for applications where a bus is shared, in an unbiased fashion, among a large number of devices.

I. INTRODUCTION

In computer systems, situations frequently arise where a resource is shared among several devices, but it can be used by only one device at a time. Scheduling such a resource to enforce mutual exclusion over its use is necessary if devices request the resource while it is being used or if the requests arrive simultaneously. A frequently encountered resource of this type is the data bus, which provides a communication path among the various devices connected to it. At any given time, there can be several devices receiving (or reading) information from the bus, but there can be only one device that has the privilege of transmitting information on it. Such a device is called the bus-master, and mutual exclusion among the devices wishing to become the bus-master is enforced by bus arbitration schemes.

Devices requesting the bus while it is busy are made to wait until it becomes available again. As soon as that happens, one among the waiting devices is allowed to become the bus-master. In most bus arbitration schemes, this choice is made without regard to the order in which the requests originally arrived; for example, daisy-chaining, device polling, and parallel priority resolution schemes.

Some of the commonly used bus arbitration schemes have been reviewed by Chen and Thurber et al.[1,2] Among them, polling and daisy-chaining are most commonly used. Polling is suitable only for slow devices, because the waiting times from bus request to bus grant are quite long, as the devices can access the bus only during preassigned time intervals. Daisy-chaining is extensively used in several minicomputers, such as the PDP-11s made by Digital Equipment Corporation (DEC).[3] Arbitration delay in this scheme may be quite long, since it is proportional to the number of devices connected to the bus. Furthermore, by virtue of their location on the bus, the devices are assigned fixed priorities that are used for contention resolution.

For faster bus arbitration, the recent computers designed by DEC and Honeywell, Inc. use distributed schemes.[4,5] These schemes, and those described in Refs. 6, 7, and 8, use the same algorithm with different implementations. They are fast, modular, and flexible, but they, too, allocate fixed priorities to the devices connected to the bus.

The major drawback of allocating fixed priorities to the devices is that the low priorities may have to wait indefinitely before being granted access to the bus if a few high-priority devices decide to use the bus frequently. They are effectivelly "locked out" from service. See Ref. 9 for a simulation-based quantitative analysis of these and other bus arbitration schemes.

In this paper, we present a first-come-first-serve (FCFS) scheme that allocates the bus in an order that is a close approximation to that in which the devices request the bus. This scheme does not have the above-mentioned drawback of locking out a few devices from service, and it provides an equal grade of service to all devices. We first describe the scheme and then discuss two alternative hardware implementations—one centralized and the other distributed. Before describing the scheme, we briefly discuss the advantages of following the FCFS allocation policy.

Consider a data bus that is shared among several devices, and assume that (*i*) the devices request the bus with the same statistics, and (*ii*) the bus is allocated for a fixed quantum of time for each request. The bus arbitration scheme should then have the following two properties:

(*i*) It will minimize the idle time on the bus, so that the bus throughput is maximized. This is done by arbitrating for the next bus-master concurrently with the bus usage and by reducing the arbitration time, if it happens to be longer than the time quantum for which the bus is allocated.

(*ii*) It will have the least disparity of service across requests and also across devices. The disparity of service across requests is represented by s, the standard deviation of the waiting times taken over all bus requests, and the disparity of service across devices is represented by S, the standard deviation of the average waiting times experienced by the individual devices. It is a simple matter to show that if an arbitration scheme does not prefer a device over any other, the average waiting times experienced by individual devices are all equal. Such schemes are called unbiased schemes, and S for them is zero. The ideal FCFS scheme is one such scheme. In the Appendix we show that the ideal FCFS scheme also attains the minimum value of s. Thus, under the assumptions stated above, the ideal FCFS scheme is a desirable scheme to be emulated in real systems.

II. DESCRIPTION OF THE SCHEME

Let there be N devices connected to the bus. In order to ensure that the devices gain bus control one at a time and in the order in which they requested it, we propose a scheme that is very similar in essence to that used in many supermarkets. As customers walk in, they pull out a numbered ticket from a machine that dispenses sequentially numbered tickets. When the server becomes free, he or she waits on the customer with the ticket one number higher than that of the last customer served, thus, providing equitable service to all customers.

Our scheme is based upon two essential pieces of information: "next number to be served" (NNS) and "next number available" (NNA). This information can be maintained in a centralized or distributed fashion, as we discuss in Section III. In addition, each device has a register, called the ticket register, to store the ticket number assigned to it when it requests bus mastership. How these ticket numbers are assigned is discussed later; let us first see how they are used. As soon as the bus is available, each device compares its ticket number with the NNS, and the device that finds the match becomes the bus-master. As we explain in the following discussion, there can be only one device whose ticket number matches NNS. Sometime before the bus is available again, NNS is incremented by one. This incrementing is done modulo NTICKETS, so that the ticket numbers range from 0 to (NTICKETS-1). To ensure that devices have distinct ticket numbers, we must have NTICKETS $\geq N$, where N denotes the number of devices.

Now we consider how the ticket numbers are assigned. This is done using NNA. When a device wants to become the bus-master, it copies the NNA into its ticket register. Then, NNA is incremented by one modulo NTICKETS, thus, ensuring that sequentially increasing ticket numbers are "dispensed" in the range from 0 to (NTICKETS-1). Of course, while the copying and incrementing operations are being done, no other device should be allowed to copy the NNA. If the NNA is copied, either there will be two devices with the same ticket number, and confusion will ensue as they both will become bus-masters at a later time, or there will be a device with an invalid ticket number that is outside the above-specified range, and that device would never be able to access the bus, as NNS will never be equal to the invalid ticket number. The accesses to NNA to receive ticket numbers should, therefore, be mutually exclusive.

Thus, in our ticket assignment scheme, achieving mutual exclusion for the bus depends on achieving mutual exclusion at a lower level—that of NNA. The second mutual exclusion is achieved by using one of the existing arbitration schemes; for example, simple daisy-chaining (SDC), rotating daisy-chaining (RDC), modified device polling (MDP), dynamic parallel priority resolution (DPPR), etc. See Ref. 9 for a detailed description and comparison of various bus arbitration schemes.

The duration for which NNA is allocated to a device is the time it takes to copy NNA into its ticket register. This duration is very short—typically, a few gate delays. Thus, the time required to assign a ticket number is essentially the time spent in arbitrating for the use of NNA. Whenever this time is short, as compared to the time for which the main bus is allocated, our scheme would be a close approximation to the ideal FCFS scheme. This is because the devices that request the bus while it is busy are quickly assigned ticket numbers and put into a waiting state. Thus, the scheme remembers the order in which the requests arrive.

It is also possible that some devices will request the bus while NNA arbitration is in progress or while NNA is in use. In such cases, depending upon the NNA arbitration scheme, one among these devices is allowed to copy the next NNA, and they may or may not receive the ticket numbers in the temporal arrival order of their requests. Thus, for the overall scheme to be unbiased, the NNA arbitration scheme must treat the devices in an unbiased way. This is desirable because it is a necessary condition for making the overall scheme a close approximation to the ideal FCFS scheme.

To summarize, the NNA arbitration scheme should be fast and unbiased. We use the criteria in choosing the NNA arbitration scheme. In addition, to judge how close the overall scheme is to the ideal FCFS scheme, we use s and S. The smaller the value of s, the better the approximation, since s is minimum in the ideal case. Similarly, the smaller the value of S, the better the approximation, since S is zero in the ideal case.

We now examine the SDC, RDC, MDP, and DPPR schemes mentioned earlier with regard to their desirability as NNA arbitration schemes.

In SDC, the central arbiter sends out a daisy-chained NNA-grant signal. If a device does not want to access the NNA, it lets the signal pass through; otherwise, it stops the signal and then accesses NNA. Thus, the devices closer to the arbiter are preferred over those farther away from it. This tends to make the overall scheme, named FCFS/SDC, a poorer approximation to the ideal FCFS scheme.

In RDC, the device that accessed NNA last acts as the arbiter for the next arbitration cycle and sends out the NNA-grant signal. On the average, all the devices are given equal treatment, and the average arbitration time is the same as that for SDC. Therefore, the overall scheme, FCFS/RDC, is expected to be a better approximation of the ideal FCFS scheme than the FCFS/SDC.

In MDP, there is no central arbiter, and the daisy-chained NNA-grant signal keeps travelling from device to device in a cyclical fashion. Devices wishing to access the NNA wait for the grant signal to arrive, stop the grant signal temporarily, access the NNA, and then release the grant signal. All the devices receive unbiased treatment. The performance of FCFS/MDP is, therefore, expected to be similar to that of FCFS/RDC, and their hardware implementation is also quite similar.

In DPPR, devices are assigned priorities which change after each NNA arbitration cycle. As the arbitration starts, all the devices that need to access the NNA put their priorities on a common priority bus. Then, each device removes itself from the contention if its priority is lower than the composite priority on the priority bus. This eliminates all but the highest priority device, which then accesses the NNA.[9] The dynamic assignment of priorities in this scheme can be done in a variety of ways, but here we assume it is done so that the order in which devices win arbitration is essentially the same as that of RDC (the same priority assignments emulate MDP also). Initially, the ith device is given the priority i, and after each arbitration, priorities are cyclically rotated so that the device that won the last arbitration gets the priority one. All the devices are treated equally; however, the average arbitration time for DPPR is much smaller than that of RDC or MDP, because there is no daisy-chained signal involved that gets delayed while passing through each device (by as much as 4 gate delays per device). Thus, FCFS/DPPR is a better approximation to the ideal FCFS scheme than FCFS/RDC and FCFS/MDP. The disadvantage is that FCFS/DPPR requires more hardware than FCFS/RDC or FCFS/MDP.

The conclusions drawn above are supported by the values of s and S obtained through simulation, which are shown in Tables I and II, respectively. (These statistics have been borrowed from Bain and Ahuja.[9]) Both tables include two cases: (*i*) when the schemes discussed above are used for NNA arbitration in the ticket assignment scheme, and (*ii*) when they are used to arbitrate for the main bus itself. Note that the values of s and S for ticket assignment schemes (column 1) are smaller than for the others (column 2); therefore, they are better approximations to the ideal FCFS scheme. Similarly, among the ticket assignment schemes, FCFS/DPPR is the closest approximation to the ideal FCFS scheme. Through simulations, we also observed that as the number of devices is increased, the performance of FCFS/DPPR rapidly converges to that of the ideal FCFS scheme, but the performance of other schemes diverges significantly from that of the ideal FCFS scheme. Therefore, FCFS/DPPR is an attractive scheme when a large number of devices (approximately 16 or more) share a common bus.

III. IMPLEMENTATION OF THE TICKET ASSIGNMENT SCHEME

In this section, we describe and compare two implementations of the ticket assignment scheme. In the first, the NNA and NNS are centralized, and in the second, they are distributed. We consider only the FCFS/MDP scheme, since the implementations with different NNA arbitration schemes are quite similar.

Figure 1a shows an implementation of FCFS/MDP in which the NNA and NNS counters are centralized, and the devices access them through the NNA and NNS buses. If a device requests access to the main bus, it waits for NNA-GT, the cyclically daisy-chained NNA grant signal, to

Table I—Table of s, the standard deviation of the weighting times taken over all requests to the bus. X denotes the schemes in the first column and FCFS/X denotes the ticket assignment using X for NNA arbitration.

X	s for FCFS/X (μs)	s for X (μs)
SDC	19.32	30.26
RDC	1.476	3.214
MDP	1.512	3.230
DPPR	1.368	3.159
Ideal FCFS		1.112

Note: Simulations were carried out for 32 independent devices, each device requesting the bus with uniformly distributed interrequest times between 0.4 and 19.6 μs, with the average interrequest time of 10 μs. The bus was allocated for 0.4 μs for each request.

Table II—Table of S, the standard deviation of the average weighting times experienced by the individual devices. Simulations were carried out under the same conditions as shown in Table I.

X	S for FCFS/X (ns)	S for X (ns)
SDC	524000*	132600†
RDC	68.0	199.9
MDP	63.7	155.8
DPPR	45.0	153.9

* Devices 26 through 32 did not get service.
† Devices 23 through 32 did not get service.

arrive. The device then holds the NNA grant signal, transfers the data on the NNA bus to its ticket register, signals on the INC-NNA line to increment the NNA counter, and then releases the NNA grant signal. (See Fig. 1b for a detailed circuit diagram.) After the device has a ticket number, it waits until the contents of its ticket register are the same as the data on the NNS bus and the bus busy line, BB, is negated. When that occurs, it asserts BB, becomes the bus-master, and signals on the INC-NNS line to increment the NNS counter. After using the bus, it simply negates the BB line. The BB line permits incrementation of NNS to proceed while the main bus is being used.

Figure 2 shows a distributed implementation of the above scheme in which each device has its own NNA and NNS counters. The NNA and NNS buses are eliminated, and the INC-NNA and INC-NNS lines are used to keep the various NNA and NNS counters up to date. The initial values of all the ticket registers, NNA counters, and NNS counters are 0, 1, and 1, respectively.

When a device wants a ticket number, it executes the following sequence of steps:

(*i*) Waits for NNA-GT to arrive, and captures it on arrival.
(*ii*) Initiates steps (*iii*), (*iv*), and (*v*) when INC-NNA becomes false, and does nothing before then.
(*iii*) Shifts the contents of NNA counter into the ticket register.
(*iv*) Signals all the other devices to increment their NNA counters by asserting INC-NNA line. The device also signals itself to do the same. The INC-NNA line is negated after a long enough time to allow the NNA counter to finish incrementing.
(*v*) Releases the NNA-GT signal.

Notice that only steps (*i*) and (*v*) depend on the NNA arbitration

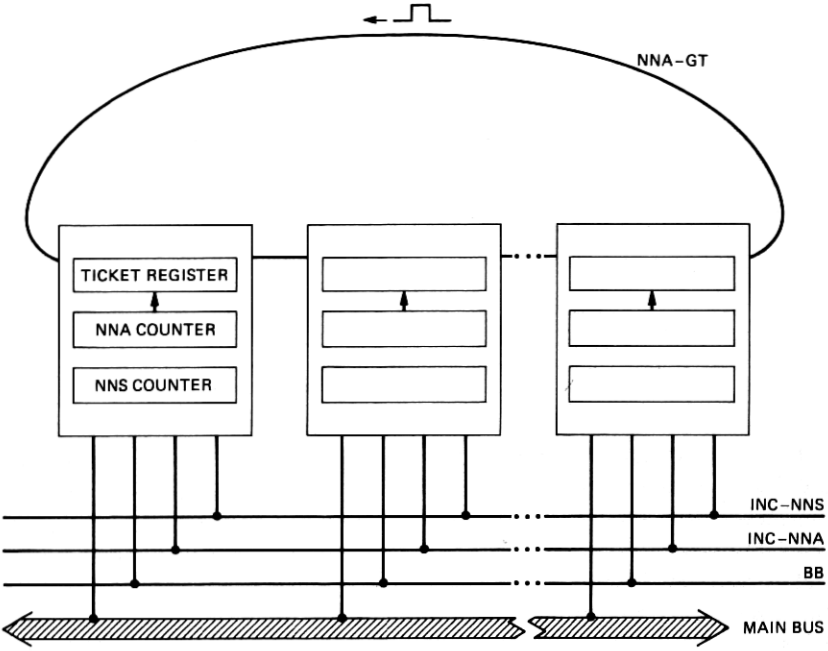

Fig. 1a—An implementation schematic for FCFS/MDP where NNA and NNS are centralized. INC-NNA and INC-NNS are used by the devices to increment NNA and NNS, respectively. NNA-GT is the cyclically daisy-chained grant signal for accessing NNA.

scheme being MDP; they are replaced by a different set of steps for different NNA arbitration schemes. All other steps, including those given below, are independent of the NNA arbitration scheme used. When a device receives the INC-NNA signal, it simply increments the NNA counter.

In the distributed implementation, gaining control of the main bus is similar to that in the centralized implementation:

(*i*) After receiving the ticket number, wait until the contents of the NNS counter and the ticket register are the same, and INC-NNS and BB are false. When that occurs, initiate steps (*ii*) through (*v*); do nothing before then.
(*ii*) Set BB to true.
(*iii*) Signal all other devices to increment their NNS counters by asserting the INC-NNS line. The device also signals itself to do the same. The INC-NNS line is negated after a long enough time to allow the NNS counter to finish incrementing.
(*iv*) Use the main bus.
(*v*) Release the bus by setting BB to false.

As a device receives INC-NNS, it increments its NNS counter. The detailed circuit diagram for this is similar to Figure 1b, except that each device has NNS and NNA counters of its own.

The distributed implementation has two advantages over the centralized implementation.

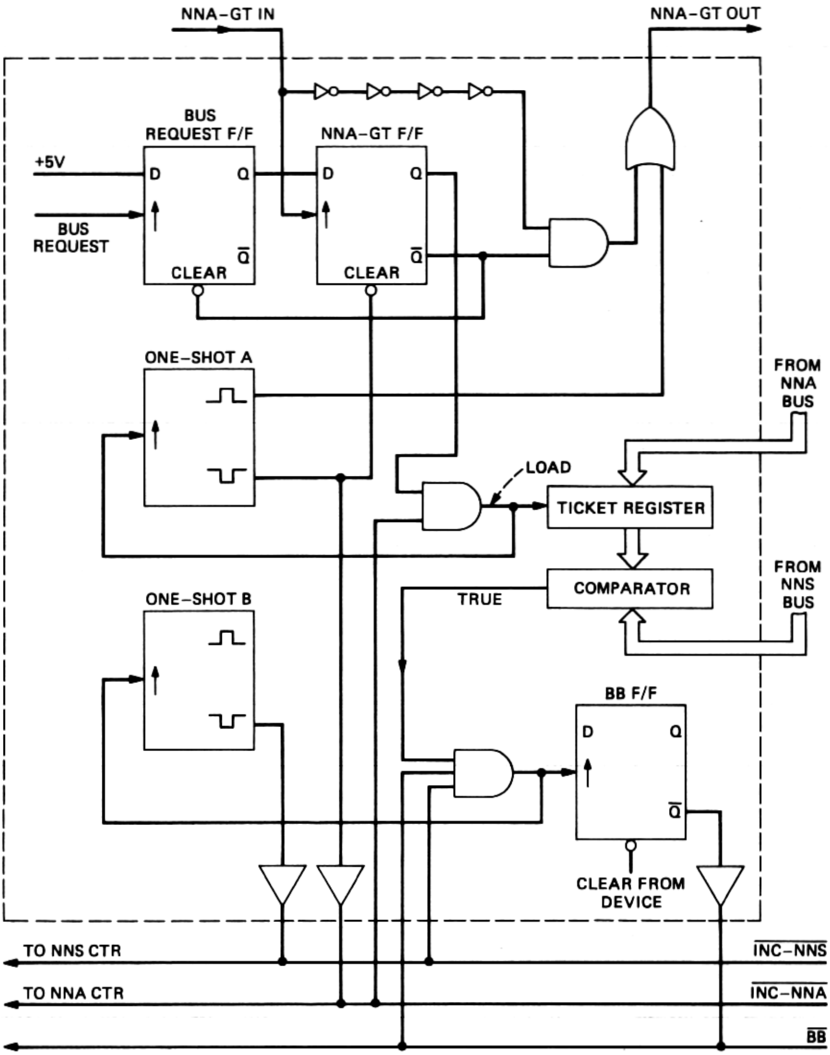

Fig. 1b—The detailed circuit diagram associated with the schematic Fig. 1a. The circuit enclosed within the broken lines is contained in each device. F/F denotes flip-flop. The bus-request F/F and NNA-GT F/F capture the grant signal. The one-shot A generates the outgoing grant signal, the negative of which is also used to signal on the INC-NNA line. The one-shot B generates the INC-NNA signal. Notice that the buses use negative logic.

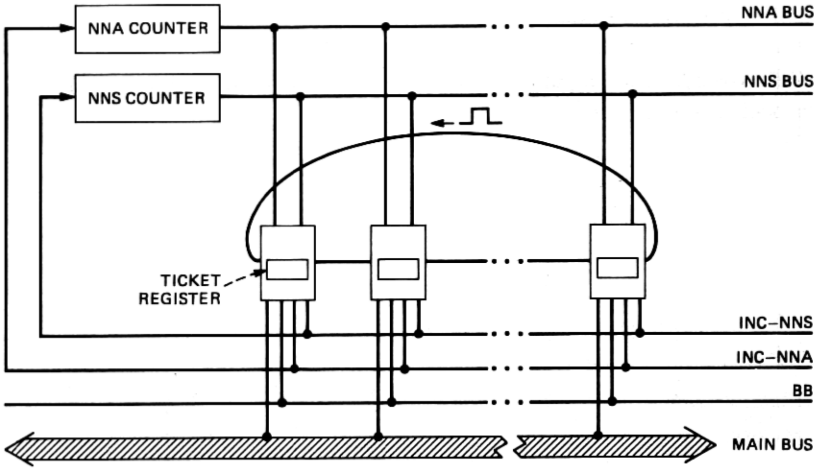

Fig. 2—An implementation schematic for FCFS/MDP where NNA and NNS are distributed. Each device has NNA and NNS registers. INC-NNA and INC-NNS signals are used to keep all the NNA and NNS registers up to date. The NNA and NNS buses have been eliminated.

(*i*) It has fewer lines, since it does not need the NNA and NNS buses.
(*ii*) Devices do not have to wait for voltage levels on the NNA and NNS buses to settle down, as NNA and NNS are available from their local counters. The longer the bus, the more significant this advantage because the settling time of voltages on the bus is proportional to the length of the bus.

The distributed implementation has two disadvantages as compared to the centralized implementation:

(*i*) In order to introduce new devices in the system, their NNA and NNS counters must be current with those in other devices. Although not always satisfactory, this can be done by stopping the system momentarily to reset the counters.
(*ii*) The scheme will malfunction if any one of the counters malfunctions. Depending upon the reliability of the hardware, this disadvantage may not be serious.

Thus, since neither implementation is unequivocally superior to the other, the final choice should be made depending upon the requirements of the application at hand.

IV. SUMMARY

We presented a FCFS bus arbitration scheme that is based upon assigning ticket numbers to the devices as they request the bus. The arbitration for the main bus essentially depends upon the arbitration for the next available ticket number. Several schemes for the latter arbitration were considered, and their impact on the overall scheme was examined using the standard deviation of wait times of all requests and the standard deviation of the average weight times of devices. Using simulation results, we showed that the overall scheme is the closest approximation to the ideal FCFS scheme, when the lower level arbitration is performed by the dynamic, parallel priority-resolution scheme; the resulting overall scheme is called FCFS/DPPR. Two alternative implementations, one centralized and the other distributed, of the overall scheme were described.

V. ACKNOWLEDGMENT

We gratefully acknowledge several fruitful discussions that we had with W. L. Bain, Jr., during the course of this work.

APPENDIX

In the following, we show (*i*) that the ideal FCFS has the minimum value of s, the standard duration of waiting times of all the requests, and (*ii*) that all disciplines of serving the requests have the same, $\bar{w}$, the average waiting time of requests.

Consider the idealized arrangement where the incoming requests are put in a queue in their order of arrival, and the server always picks the first—the oldest—element, in the queue. This is the ideal FCFS scheme. If at any time, the elements in the queue are permuted, we obtain deviations from the ideal case.

Let w_i be the waiting time of the ith request when the queue is not disturbed. Then, for the ideal FCFS scheme,

$$\bar{w} = \frac{1}{N} \sum w_i,$$

and

$$s^2 = \frac{1}{N} \sum (w_i - \bar{w})^2,$$

where N is the total number of requests.

Since any permutation can be expressed as a composition of a number of permutations that exchange two elements, we show that the value of $\bar{w}$ remains the same and that the value of s^2 is increased, if two elements in the queue are interchanged. For simplicity, we assume that the ith and the $(i - 1)$st elements are interchanged. A similar argument holds for the general case also. The new waiting times for these two elements are

$$w_i' = w_i - t,$$

and

$$w_{i-1}' = w_{i-1} + t,$$

where t is the service time for each request. Hence, the difference between the new and the old values of $\bar{w}$ is

$$\Delta \bar{w} = \frac{1}{N}(w_{i-1}' + w_i') - \frac{1}{N}(w_{i-1} + w_i)$$

$$= 0.$$

Also, the difference between the new and the old values of s^2 is

$$\Delta s^2 = \frac{1}{N}[(w_{i-1}' - \bar{w} - \Delta \bar{w})^2 + (w_i' - \bar{w} - \Delta \bar{w})^2]$$

$$- \frac{1}{N}[(w_{i-1} - \bar{w})^2 + (w_i - \bar{w})^2]$$

$$= \frac{2t}{N}(t + w_{i-1} - w_i).$$

Note that the maximum value of w_i occurs when the ith request arrives in the queue immediately after the $(i-1)$th request. If the ith request comes later, then the server services some requests in the meantime, thus, reducing the waiting time of the ith request. Hence,

$$w_i \leq w_{i-1} + t.$$

This gives us

$$\Delta s^2 \geq 0,$$

where the equality occurs only when the ith and the $(i-1)$th requests come at the same time. Hence, the ideal FCFS scheme has the minimum value of s^2.

REFERENCES

1. R. C. Chen, "Bus Communication Schemes," NTIS, PB 235-897 (January 1974).
2. K. J. Thurber et al., "A Systematic Approach to the Design of Digital Bussing Structures." AFIPS Conf. Proc. FJCC, 1972.
3. J. V. Levy, "Buses, the Skeletons of Computer Structures," *Computer Engineering*, Bedford, Mass.: Digital Press, 1978, Chapter 11.
4. Digital Equipment Corporation, *VAX-11/780 Architecture Handbook*, 1977.
5. J. W. Conway, "Approach to Unified Bus Architecture Sidestepping, Drawbacks," Comput. Des. *16*, No. 1 (January 1977), pp. 71–6.
6. H. Keller and E. H. Forrester, "Rapid Priority Resolution," U.S. Patent 3,983,540, September 28, 1976.

7. A. G. Fraser, private communication, 1975.
8. K. A. Elmquist, *et al.* "Standard Specifications for S-100 Bus Interface Devices," *Comput., 12,* No. 7 (July 1979), pp. 28–52.
9. W. L. Bain, Jr., and S. R. Ahuja, "Performance Analysis of Digital Buses for Multiprocessing," Proc. Eighth Int. Symp. Computer Architecture, Minneapolis, Minn., May 1981.