MODELLING AND PERFORMANCE EVALUATION OF MULTI–PROCESSORS ORGANISATION WITH SHARED MEMORIES

M. Naderi

Department of Electrical Engineering Tehran University of Science and Technology, Tehran, Iran

Received April 1987

Abstract This paper is primarily concerned with theoretical evaluation of the performance of multi-processors system. A markovian waiting line model has been developed for various different multi-processors configurations, with shared memory. The system is analysed at the request level rather than job level.

چکیده بحث اصلی مقاله حاضر در مورد بررسی و تجزیه و تحلیل تئوری رفتار و عکس‌العمل سیستم‌های کامپیوتری است که شامل چندین پردازشگر میباشند. مدل ریاضی برای چندین نوع از این تشکیلات با حافظه مشترک براساس صفوف انتظار در حالت Markovian تهیه و استخراج گردیده است. در تهیه مدلهای ریاضی اساس تجزیه و تحلیل بر پایه درخواست‌های ورود بحافظه بجای حجم عملیات اجرایی در سیستم کامپیوتری در نظر گرفته شده است.

Modelling and Performance Evaluation of Multi-Processors Organisation.

Thirty years ago the designer designed his systems out of circuit level components such as resistors, diodes and transistors. Later, switching circuit level components which are represented by gates and flip-flops, became available as SSI components. With the introduction of medium scale integration (MSI) register transfer level components appeared as well as arithmetic and logic units, registers, etc. The advent of large scale integration (LSI) has made memories and, even processors, primitive components from which systems are designed.

VLSI as microprocessors have had a dramatic impact on applications that require minor adjustments. They have been used for instruments, industrial controllers, intelligent terminals, and communication systems as special function processors in large computers and multi-micro-networks.

The question naturally arises as to whether the microprocessor, which has proved so successful in these diverse applications, can be used as a building block for large general purpose computer networks. In other words, can a suitably interconnected set of microprocessors be used for tasks that currently require large uniprocessor capable of executing millions of instruction per second?

At present there is no definitive answer to this question, but there are several reasons to believe that multiple-microprocessor systems might indeed be viable. Following this belief one can think about how one can connect several of these processors and memories together and what is the effectiveness of the organisations. Finally, how can one measure or evaluate the performance parameters of such an organisation.

Throughout this paper an attempt is made to provide resonable answers to some of the above questions through a mathematical waiting line model approach. A method of investigation is proposed to evaluate the performance of a large scale multi-processors as a sophisticated tool for decreasing the financial risk involved in developing a network of large organisation.

The objectives of this investigation are:

1: The ability to model existing and proposed multiprocessors.

2: The ability to compare different multiprocessor organisations, through system performance.

3: To provide a facility (tool) with which one can easily create and try out new multiprocessor systems, tailored to a particular application

4: To enable one to measure the performance parameters of available multi-processor organisations.

Performance Evaluation of Multi-Processors Organisation.

The following statistical problem arises in the study of many computer organisations. A memory unit is required to attend to a number of processors and I/O units which request memory from time to time. Each time a unit (processor or I/O) makes a request, the memory unit organisation has to do a certain amount of decision-making and testing before it can be restarted to give a service to that unit. If a unit makes a request while the memory is on communication mode with another unit, the second unit must wait until the memory finishes with the first one. A waiting line or queue occurs in any multicomputer organisation. When, at a given time, the number of requests desiring service (from memory or any resources) exceeds the capacity of the service facility.

The problem is to estimate the effect of interference on throughput and efficiency, the time taken to give service to a request and the number of waiting units for a shared memory in multi-processor organisations. We are interested in applying a waiting line model in such organisations which is shown in figure 1. In such an organisation each microprocessor or I/O unit apply a request for a shared memory unit.

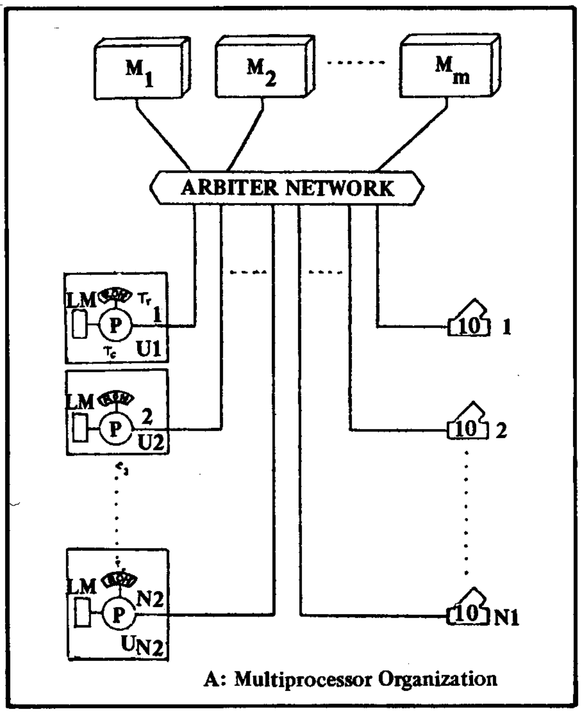


Figure 1. Multi-processors organization.

All such situations may be described by mathematical models which we term Finite Request Source Waiting Line Modles and define as waiting line models in which the request for use of memory emanates from N units (processors and I/O, s). In figure 2 it shows how several units made a request for a memory unit and how the memory unit gave service to them, one by one, in the order of first come, first served (FCFS).

If a processor or I/O unit makes a request it will be immediatley attended to by the memory if the memory unit is free, but if the unit makes a request while the memory is busy with another unit it has to wait until the memory becomes free. This results in a loss of efficiency which is said to arise due to interference. Obviously, if the number units assigned to the one memory system becomes large, the loss of efficiency due to the interference increases. On the other hand, if fewer processor and I/O units are assigned to the one shared memory (or any resources) the price per performance increases. Hence, the problem is to determine the number of units to be assigned to a memory in order to minimize output costs by balancing the cost of the memory system against the loss of efficiency due to multi-processor organisations.

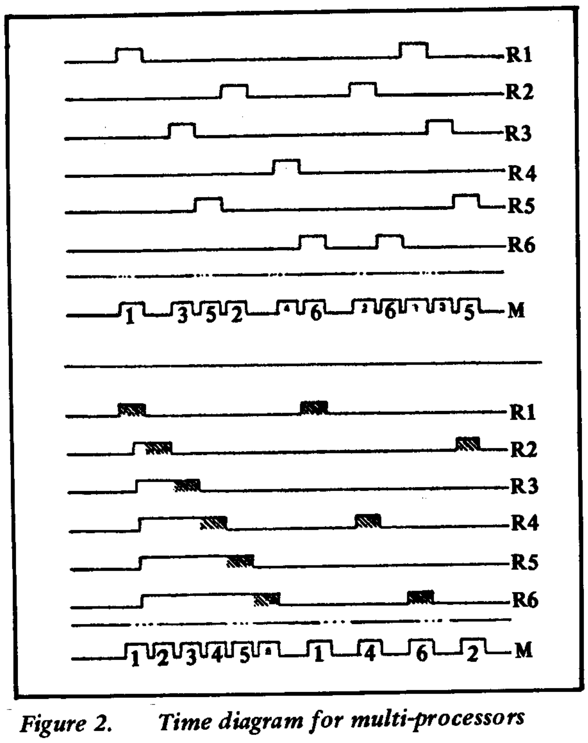


Figure 2. Time diagram for multi-processors organization.

Markovian Waiting Line Model Organisation for Multi-Processors System

We now consider a subsystem (figure-3) of multi-processor organisations with N2 processors and N1 I/O, S. Each processor has a local or private memory which communicates with it. As far as it can satisfy all requirements, if no desirable instruction or data at local memory can be found, the processor makes a request to communicate with the main memory unit. The effect of I/O units activity will not be modelled explicitly, the I/O is an I/O processor. Thus, from the main memory point of view, the I/O channel is logically equivalent to the other processors.

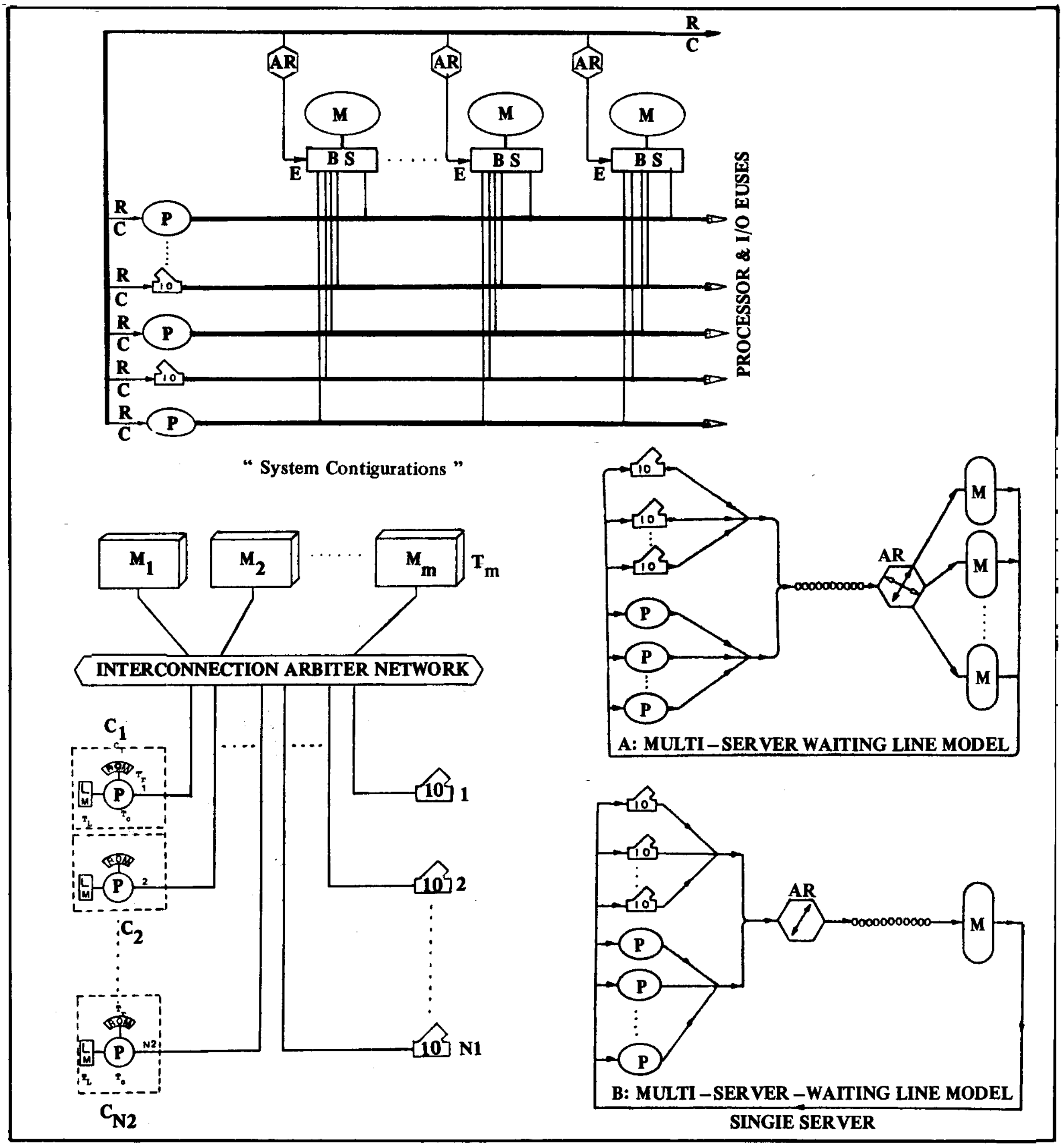
Since all units function independently, it follows that the probability of occurrence of a request is proportional to the total number of units i.e., N= N1 + N2, functioning at a given instant. The mean frequency of arrival requents in the system is a function of the number of waiting units in the queue under the service, i.e., n. In order to simplify the calculations, it is assumed that R is the mean arrival request frequency per unit of time so that when all N units are running independently of the main memory or in case of I/O units being idle.

In the most common stochastic queuing models it is assumed that inter-arrival request times and service or communcation times obey the exponential distribution or, equivalently, that the arrival rate of request (R) and service rate (S) follow a poisson distribution.

In subsystem organisations under these conditions and assumptions, the probability that a request can occur between time (t) and time (t + dt) is equal to: (N–n) Rdt + O(dt) The probability of service completion in(dt) is:

$$Sdt + O(dt)$$

and the probability of more than one request (R) more than one service completion (s) in(dt) is infinitesimal and will be O(dt)



R
C
AR
M
BS
E
P
IO
PROCESSOR & I/O EUSES
" System Contigurations "
M_1
M_2
M_m
T_m
INTERCONNECTION ARBITER NETWORK
C_1
C_2
C_{N2}
IO 1
IO 2
IO N1
A: MULTI-SERVER WAITING LINE MODEL
B: MULTI-SERVER-WAITING LINE MODEL
SINGIE SERVER

Figure 3. Multi Micro Processors Organization and Waiting line Models

We have, then, a process with requests and services occurring randomly over time, with the probability mechanism just described. Request arrivals can be considered as "BIRTHS" to the system. Since, if the queue system is in state n (we consider system state as the number of waiting units in the system) and a new request occurs, the state is changed to (n+1).

On the other hand, a services completion occurring while the system is in state (n) sends the system down one to a state of (n−1) and can be looked upon as a "DEATH". This type of process is often referred to as a

"BIRTH–DEATH" process, or markov processes with stationary transition and countable state space.

The possible states transition of Markovian chainor birth and death processes is shown in figure 4.

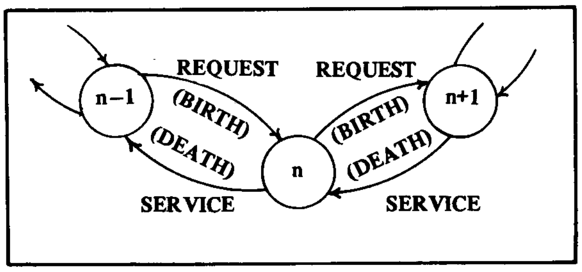


Figure 4. State Transition of markovian feeqss

According to the Chapman-Kolmogrov euation, the probability that there are (n) units in the system at some selected moment is the sum of probabilities associated with the state changes leading to state n.

$$P_n(t+dt) = \sum_m P_m(t) \cdot P_{mn}(dt) \qquad (1)$$

where

$P_m(t)$ = prob. system in state m at time t

$P_{mn}(dt)$ = prob. $\{$transition from state m to state n in dt/system in state m at time t$\}$.

In the markov process we shall permit direct transitions from any state (m) to any state (n). The transition probabilities are permitted to vary in time. But in our case, we have already seen the direct transition from a state n is possible only to the neighbouring states (n+1) or (n−1); (the poisson process property).
Thus in such a case, we have:

$$P_n(t+dt) = \sum_{m=n-1}^{n+1} P_m(t) \cdot P_{mn}(dt) \qquad (2)$$

since $P_{mn}(dt) = 0$ for $m < n-1$ and $m > n+1$.

an with arrival request (R) and service rate (S) both independent of each other, therefore we can write:

$P_n(t + dt) = P_n(t)$. prob. no request in dt prob. no service completions in dt $+P_n(t)$. prob. one request in dt . prob. one service in dt $+ P_{n+1}(t)$. prob. one service completed in dt . prob. no request in dt $+ P_{n-1}(t)$. prob. one request in dt . prob. no service completion in dt $+ o(dt)$

for $n \geqslant 1$ (3)

The transition probabilities from state (m) to (m+1) will be:

prob. no request in dt $= [1-(N-m)\,Rdt - o(dt)]$

prob. no service completions in dt $=[1-Sdt - o(dt)]$

prob. one request in dt $=(N-m)\,Rdt + o(dt)$

prob. one service in dt $=Sdt + o(dt)$

for $m=n-1, n, n+1$ (4)

We substitute the above transition probabilities to the differential equation (3) and combining all O(dt) terms and realising terms with $(dt)^2$ are also O(dt), we get,:

$$P_n(t + dt) = P_n(t)\,[1-(N-n)\,Rdt - Sdt] + P_{n+1}(t)\,[Sdt) + P_{n-1}(t)\,[N-(n-1))\,Rdt] \quad \text{for } o > n > N. \qquad (5)$$

While, for the limiting states, n=o and n=N

$$P_o(t+dt) = P_o(t)\,[1-NRdt] + P_1(t)\,[Sdt] \qquad n=o$$

$$P_N(t+dt) = P_N(t)\,[1-Sdt] + P_{n-1}(t)\,[Rdt] \qquad n=N$$

(6)

We can rewrite the three above equations (5, 6) slightly differently to yield differential-difference equations;

$$\frac{P_n(t+dt) - P_n(dt)}{dt} = \frac{d}{dt} P_n(t)$$

$$\left.\begin{array}{ll}\frac{d}{dt} P_o(t) = -N R P_o(t) + S P_1(t) & (n=o) \\ \frac{d}{} P_n(t) = -[(N-n) R + S]\, P_n(t) + & \\ \qquad (n-n+1) R P_{n-1}(t) + S P_{n+1}(t) & \\ & (o > n > N) \\ \frac{d}{dt} P_n(t) = -S P_n(t) + R P_{n-1}(t) & \\ & (n=N) \end{array}\right\} \tag{7}$$

To get the steady-state solution for P_n, the probability of n units waiting in the system at an arbitrary point of time after steady state is reached, we take the limit as $t \to \infty$ of equations (7), when $P_n(t)$ is independent of time, $d\,P_n(t)/dt$ is zero, and we have:

$$-N R P_o + S P_1 = 0 \qquad (n=0)$$

$$-[(N-n) R + S]\, P_n + (N-n+1) R.\, P_{n-1} +$$

$$S P_{n+1} = 0 \qquad (o > n > N)$$

$$-SP_n + R P_{n-1} = 0 \qquad (n=N) \tag{8}$$

From these equations (8) we can find the recursion formula

$$\begin{array}{l} P_1 = N\,(R/S)\,P_0 \\ P_2 = (N-1)\,(R/S)\,P_1 \\ P_3 = (N-2)\,(R/S)\,P_2 \\ P_{n+1} = (N-n)\,(R/S)\,P_n \end{array} \tag{9}$$

Now we have to solve a set of difference equations, to find the probability distribution of the random variable (n)

$$P_n = \frac{N!}{(N-n)!} (R/S)^n . P_0 \tag{10}$$

Since the boundary condition

$$\sum_{n=o}^{N} P_n = 1 \tag{11}$$

We can evaluate P_0 as:

$$P_o = \frac{1}{\sum_{n=o}^{N} (R/S)^n . \frac{N!}{(N-n)!}} \tag{12}$$

The steady state probability distribution for the subsystem and whole multi-processors organization will be computed and will be graphically represented for various N and (R/S). (see Fig. 5 in the results of simulation). The expected number of processors or I/O units in the system is:

$$L = \sum_{n=o}^{N} nP_n = N - \frac{S}{R} (1 - P_o)$$

and the mean number of waiting units in the queue will be

$$L_q = \sum_{n=1}^{N} (n-1)\,P_n = \sum_{n=1}^{N} nP_n - (1-P_o)$$

$$L_q = N - \frac{R+S}{R} (1 - P_o)$$

Now, we can proceed to derive the average waiting time in the queue W_q and system (W) in the base of the FIFS (first in, first served) queue discipline.

$$W_q = \frac{L_q}{R\,(N-L)}$$

$$W = W_q + \frac{1}{S} = W_q + T_s$$

The other important parameter in such multi-processors organization is a stretching factor, which is the ratio of the time needed to execute a program with conflict to access to memory (TC),to the necessary time to execute that piece of program without the presence of contention (Ts) i.e.,

$$S.F. = \frac{T_c}{T_s}$$

$$T_c = W_q + T_s$$

$$S.F. = \frac{W_q + T_s}{T_s} = 1 + \frac{W_q}{T_s} \tag{16}$$

To derive the unit availability or processor and memory efficiency in the multi-processors organisation which all units are shared memory units, let N represent the number of units and represent the number of those units waiting in the system. We can also define the processor and memory efficiency as the proportion of the processor's or memory's time for which they are engaged in useful tasks. So we have:

$$P.E. = \frac{N-L}{N} \tag{17}$$

$$\text{memory efficiency} = 1 - P_0$$

Results of Waiting Line Model Simulation: The following discussion is essential for an understanding of how the waiting line model works.

Our objective is to develop a model that is flexible to study several possible system configurations. Using this model, it should be possible to analyse different systems, of different sizes, using the same formulas already derived. The simulation program written to implement the model are general so that a system of any size can be studied over any range of the parameter involved. However, the results reported here are limited to systems with not more than 20 unit requestors and 20 memory units for different values of (R/S).

In a subsystem of multi-processors the main memory is shared by all the processors and I/O, S, is also used as the "mail box" to pass messages between the processors. In addition to the shared memory, each processor has a "private" or "Local memory"; each of which can be accessed by only one processor. Local memories are used to store codes, tables of frequently used subroutines, tables for allocation of some private resources, etc, since the local memories do not communicate with each other. In a multi-microprocessor organization each processor can independently execute its own instructions. The interval of time until it becomes necessary to access a system data base is described by a negative exponential distribution function with parameter R and could be expressed by:

$$A(t) = 1 - e^{-Rt} \quad \text{(continuous dist.}$$

$$.P_n(t) = \frac{(Rt)^n}{n} e^{-Rt} \quad \text{(discrete dist.)}$$

$$R \qquad n = 0, 1, \ldots\ldots\ldots\ldots \tag{18}$$

The expected execution interval for such a processor has a value $T_s = \frac{1}{R}$ unit times.

The effect of I/O activity will not be modelled explicity. However, corresponding to tI/O and the overlapping, a fraction r_1 of the memory system time can be apportioned to I/O, S. The following figures show some representation of samples of several parameters. These parameters are:

The number or waiting units in the queue, the mean idle time for memory processors availability, memory efficiency, and finally the stretching factor in the system for several values of R/S.

Figure 5 shows a graphical representation

```
**** *** ** SIMULATION OF SSYSTEMEM PERFORMANCE ** *** ****
*** ** * A PROOBABILLLISTIC MODEL FOR MULTIPROCESSORS ** ***
***** *** ** QUEUE ORGANISATION FOR MULTI-MEMORY ** ****
* FOR RANDOM INPUT REQUEST, EXPONETIAL SERVICE TIMES .
DO YOU HAVE ANY OVERHEAD PERCENTAGE TIMES FOR TESTING -
AND DECISION MAKING TIMES IN INTERCONNECTION SWITCH ?

TYPE IN TWO PERCENTAGE NUMBER FOR YOUR ORGANISATION
 TESTING TIME=    0       DECISION MAKING TIME =    0

THERE ARE   10  OPERATING REQUEST UNITS &    1  MEMORIES
***********************************************************
                                    (R/S) = .1E-01
NO. OF WAITING UNITS IN THE SYSTEM  =       .108471
NO. OF WAITING UNITS IN THE QUEUE   = .955519E-02
AVERAGE NUMBER OF IDLE MEMORY UNITS =       .901085
THE STRETCHING FACTOR IN THE SYSTEM =    1.0966
OPERATING UNIT EFFICIENCY           =       .989153
MEMORY UNIT EFFICIENCY(AVAILIBILITY)= .989151E-01
=============================================================>
                                    (R/S) = .5E-01
NO. OF WAITING UNITS IN THE SYSTEM  =       .759263
NO. OF WAITING UNITS IN THE QUEUE   =       .297227
AVERAGE NUMBER OF IDLE MEMORY UNITS =       .537963
THE STRETCHING FACTOR IN THE SYSTEM =    1.6433
OPERATING UNIT EFFICIENCY           =       .924074
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .462037
=============================================================\
                                    (R/S) =      .15
NO. OF WAITING UNITS IN THE SYSTEM  =    3.77239
NO. OF WAITING UNITS IN THE QUEUE   =    2.83825
AVERAGE NUMBER OF IDLE MEMORY UNITS = .658592E-01
THE STRETCHING FACTOR IN THE SYSTEM =    4.03836
OPERATING UNIT EFFICIENCY           =       .622761
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .934141
=============================================================>
                                    (R/S) =      .25
NO. OF WAITING UNITS IN THE SYSTEM  =    6.02123
NO. OF WAITING UNITS IN THE QUEUE   =    5.02654
AVERAGE NUMBER OF IDLE MEMORY UNITS = .530755E-02
THE STRETCHING FACTOR IN THE SYSTEM =    6.05336
OPERATING UNIT EFFICIENCY           =       .397877
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .994692
=============================================================>
                                    (R/S) =      .35
NO. OF WAITING UNITS IN THE SYSTEM  =    7.1445
NO. OF WAITING UNITS IN THE QUEUE   =    6.14507
AVERAGE NUMBER OF IDLE MEMORY UNITS = .573852E-03
THE STRETCHING FACTOR IN THE SYSTEM =    7.1486
OPERATING UNIT EFFICIENCY           =       .28555
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .999426
=============================================================>
                                    (R/S) =      .45
NO. OF WAITING UNITS IN THE SYSTEM  =    7.77797
NO. OF WAITING UNITS IN THE QUEUE   =    6.77806
AVERAGE NUMBER OF IDLE MEMORY UNITS = .877046E-04
THE STRETCHING FACTOR IN THE SYSTEM =    7.77865
OPERATING UNIT EFFICIENCY           =       .222203
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .999912
=============================================================>
                                    (R/S) =      .55
NO. OF WAITING UNITS IN THE SYSTEM  =    8.18185
NO. OF WAITING UNITS IN THE QUEUE   =    7.18187
AVERAGE NUMBER OF IDLE MEMORY UNITS = .176598E-04
THE STRETCHING FACTOR IN THE SYSTEM =    8.18199
OPERATING UNIT EFFICIENCY           =       .181815
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .999983
=============================================================>
                                    (R/S) =      .65
NO. OF WAITING UNITS IN THE SYSTEM  =    8.46155
NO. OF WAITING UNITS IN THE QUEUE   =    7.46155
AVERAGE NUMBER OF IDLE MEMORY UNITS = .4395E-05
THE STRETCHING FACTOR IN THE SYSTEM =    8.46159
OPERATING UNIT EFFICIENCY           =       .153845
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .999996
=============================================================>
                                    (R/S) =      .75
NO. OF WAITING UNITS IN THE SYSTEM  =    8.66667
NO. OF WAITING UNITS IN THE QUEUE   =    7.66667
AVERAGE NUMBER OF IDLE MEMORY UNITS = .128993E-05
THE STRETCHING FACTOR IN THE SYSTEM =    8.66668
OPERATING UNIT EFFICIENCY           =       .133333
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=       .999999
=============================================================>
                                    (R/S) =      .85
NO. OF WAITING UNITS IN THE SYSTEM  =    8.82353
NO. OF WAITING UNITS IN THE QUEUE   =    7.82353
AVERAGE NUMBER OF IDLE MEMORY UNITS = .431631E-06
THE STRETCHING FACTOR IN THE SYSTEM =    8.82354
OPERATING UNIT EFFICIENCY           =       .117647
MEMORY UNIT EFFICIENCY(AVAILIBILITY)=    1
=============================================================>
```

```
                      DISTRIBUTION OF STATE PROBABILITIES
    .901085 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
.901085E-01 IIIIIIIIII*
.810976E-02 II*
.648781E-03 II*
.454147E-04 II*
.272480E-05 II*
.136244E-06 II*
.544976E-08 II*
.163493E-09 II*
.326985E-11 II*
.326985E-13 II*
-------------------------------------------------------> (R/S)= .1E-01
    .537963 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .268982 IIIIIIIIIIIIIIIIIIIIIIIIIII*
    .121042 IIIIIIIIIIIII*
.484167E-01 IIIII*
.169458E-01 II*
.508375E-02 II*
.127094E-02 II*
.254187E-03 II*
.381281E-04 II*
.381281E-05 II*
.190641E-06 II*
-------------------------------------------------------> (R/S)= .5E-01
.658592E-01 IIIIIII*
.987888E-01 IIIIIIIIII*
    .133365 IIIIIIIIIIIIII*
    .160038 IIIIIIIIIIIIIIIII*
    .16804  IIIIIIIIIIIIIIIII*
    .151236 IIIIIIIIIIIIIIII*
    .113427 IIIIIIIIIIII*
.680561E-01 IIIIIII*
.306252E-01 IIII*
.918758E-02 II*
.137814E-02 II*
-------------------------------------------------------> (R/S)=      .15
.530755E-02 II*
.132689E-01 II*
.29855E-01  III*
.597099E-01 IIIIII*
    .104492 IIIIIIIIIII*
    .156739 IIIIIIIIIIIIIIII*
    .195923 IIIIIIIIIIIIIIIIIIIII*
    .195923 IIIIIIIIIIIIIIIIIIIII*
    .146942 IIIIIIIIIIIIIII*
.734712E-01 IIIIIIII*
.183678E-01 II*
-------------------------------------------------------> (R/S)=      .25
.573852E-03 II*
.200848E-02 II*
.652671E-02 II*
.177148E-01 II*
.434013E-01 IIIII*
.911426E-01 IIIIIIIIII*
    .1595   IIIIIIIIIIIIIIII*
    .223299 IIIIIIIIIIIIIIIIIIIIIII*
    .234464 IIIIIIIIIIIIIIIIIIIIIIII*
    .164125 IIIIIIIIIIIIIIIII*
.574438E-01 IIIIIII*
-------------------------------------------------------> (R/S)=      .35
.877046E-04 II*
.394671E-03 II*
.159842E-02 II*
.57[illegible]E-02 II*
.18126E-01  II*
.489403E-01 IIIII*
    .110116 IIIIIIIIIIIII*
    .198208 IIIIIIIIIIIIIIIIIIIII*
    .267581 IIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .240823 IIIIIIIIIIIIIIIIIIIIIIIIIII*
    .10837  IIIIIIIIIIIII*
-------------------------------------------------------> (R/S)=      .45
.176598E-04 II*
.97[illegible]88E-04 II*
.480788E-03 II*
.211546E-02 II*
.814454E-02 II*
.26877E-01  III*
.739117E-01 IIIIIIIII*
    .162606 IIIIIIIIIIIIIIIIII*
    .268299 IIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .295129 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .162321 IIIIIIIIIIIIIIIIII*
-------------------------------------------------------> (R/S)=      .55
.4395E-05   II*
.285675E-04 II*
.16712E-03  II*
.869022E-03 II*
.395405E-02 II*
.154208E-01 II*
.501176E-01 IIIIII*
    .130306 IIIIIIIIIIIIIII*
    .254096 IIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .330325 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .214711 IIIIIIIIIIIIIIIIIIIIIIII*
-------------------------------------------------------> (R/S)=      .65
.128993E-05 II*
.967445E-05 II*
.653025E-04 II*
.391815E-03 II*
.205703E-02 II*
.925664E-02 II*
.347124E-01 IIII*
    .104137 IIIIIIIIIIII*
    .234309 IIIIIIIIIIIIIIIIIIIIIIIIIII*
    .351463 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .263597 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
-------------------------------------------------------> (R/S)=      .75
.431631E-06 II*
.366886E-05 II*
.280668E-04 II*
.190854E-03 II*
.113558E-02 II*
.57[illegible]48E-02 II*
.246138E-01 III*
.836869E-01 IIIIIIIII*
    .213401 IIIIIIIIIIIIIIIIIIIIIIIII*
    .362783 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
    .308365 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII*
-------------------------------------------------------> (R/S)=      .85

           DISTRIBUTION OF WAITING UNITS IN THE SYSTEM
          I------------------------------------------------------
  .108471 I *                                                 I

  .235825 I *                                                 I

  .759263 I   *                                               I

  2.14582 I          *                                        I

  3.77239 I                   *                               I

  5.09192 I                          *                        I

  6.02123 I                               *                   I

  6.67222 I                                  *                I

  7.1445  I                                    *              I

  7.50054 I                                      *            I

  7.77797 I                                       *           I

  8.00008 I                                         *         I

  8.18185 I                                         *         I

  8.33335 I                                          *        I

  8.46155 I                                           *       I

  8.57143 I                                           *       I

  8.66667 I                                            *      I

  8.75    I                                            *     I

  8.82353 I                                             *    I

  8.88889 I                                             *    I
          I                                                  I
          I                                                  I
          I                                                  I
          V
         R/S
```

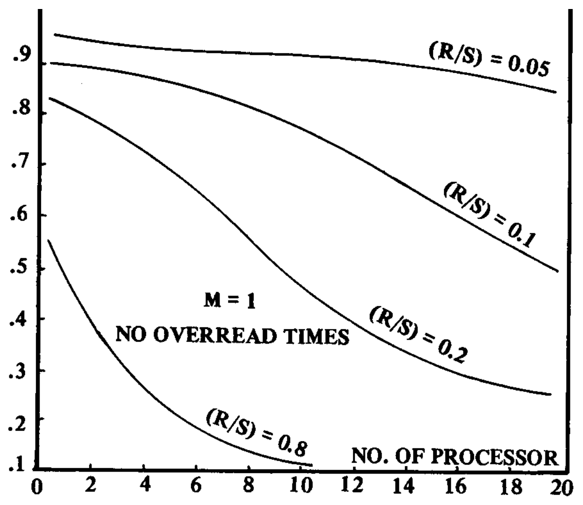


Figure 5. The Results of Waiting Line Model Simulation

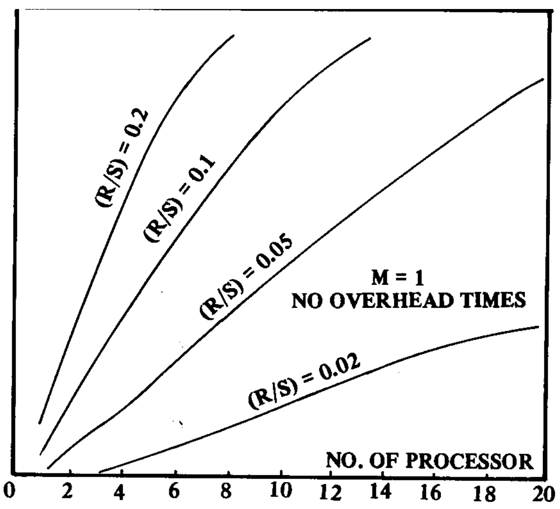


Figure 7. Memory Availability (Efficiency)

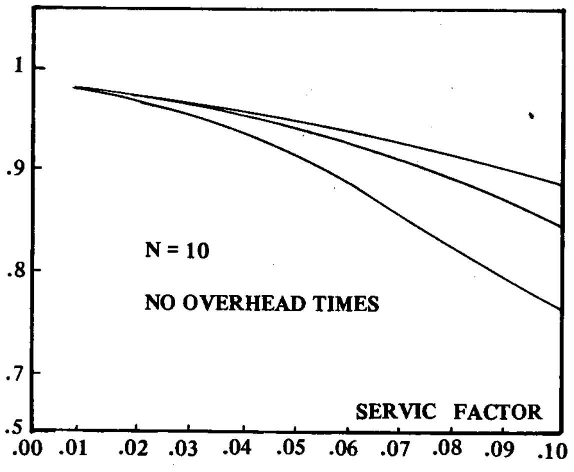


Figure 6. Processor Availability (Efficiency)

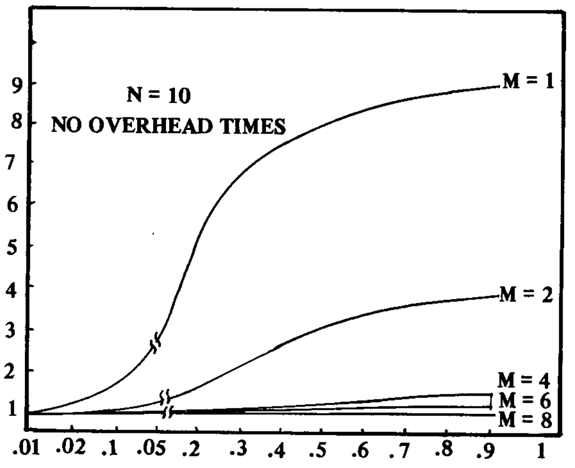


Figure 8. Stretching Factor in the System for Various Memory Units

of the distribution of P_n(probabilities of n units waiting in the system) for various values of (R/S) in order of n=0, 1, 2, ..., N.

Figure 6 and 7 show processor and memory availability versus (N) for several values of (R/S). Figure 8 shows the stretching factor plotted against N for several different values of R/S. However, in general performance of the system can be improved (increased) by adding more memory units to a system with fixed R/S.

CONCLUSIONS

This paper is concerned with the theoretical evaluation of the performance of multi-processors organisation, multi-microcomputers network.

A series of markovian waiting line models have been developed for multi-systems configurations with shared menories. The model is used to obtain quantitative results for system performance in terms of various parameters, such as:

Waiting and idle times, processor and

memory efficiency, stretching factor for program execution, and rate of instruction execution.

The objectives of this investigation are:
1– The ability to model existing and proposed multi-systems organisation.
2– To provide a facility to compare differnet multi-systems organisations, through system performance.
3– To enable one to measure the performance of available multi-processors-organisations.
4– To enable one to create and tailored a new multi-computers organisation to a particular application.

From the point of view of practical design in a multiprocessors and multi-microsnetwork organisations, the main memories are shared by all units such as processors and input-loutput units. A single copy of the operating system in this shared memory controls the entire system, since this shared memory should be accessible by all the units.

In addition to the shared memories each processor unit can have a private or local read/write memory and read only memory (ROM).

Three classes of system have been distinguished:
1– System without local and Rom memories.
2– System with local memory.
3– System with local memory and ROM. (Computer).

During system activity each memory unit is required to attend to a number of processors and I/O's which request access to that memory. The memory conflict occurs whenever two or more units attempt to access the same memory unit simultaneously. The overal offect of these conflicts is referred to as memory interfevence. The result is a loss of the system efficiency which is said to arise due to interference. Obviously, if the number of processors assigned to the one memory unit becomes large, the loss of efficiency due to interference increases. On the other hand, if fewer processors are assigned to the memory, the price per performance increases. Hence, the problem is to determine the number of processors to be assigned to a memory unit so as to minimise output costs by balancing the cost of the memory units in the system against the loss of efficiency due to units interference. It is clear that the lower values of servicing factors (RIS) create the greater idle time for memory units, fewer waiting units, smaller stretching factors and faster response time for a system. However, this may be an uneconomical design.

The economical constraints require that a good choice of relative memory and processor speed and efficiencies be made by choosing a certain value of R/S or $N \times \frac{R}{S}$ during the design of a multi-processors organisation.

REFERENCES

1. Karuh, J.: *"On the Chapman-Kolmogorov equation"*, Ann. Math. statist, Vol 32, 1961, pp. 1333–1337.
2. Lorin, H.: *"Parallelism in hardware and software; Real and apparent concurrency"* Prentice-Hall Inc, 1972.
3. Anacker, W. & Wang C. P: *"Performance evaluation of Computing systems with memory hierachies"*, IEEE Trans. On comp. EC–16 1967, 764–772.
4. Enslow, P. H. Jr.: *"Multiprocessors and parallel processing"*, John wiley & Sons, New York, 1974.
5. Computing Surveys: Special Issue; *"Parallel Processors and Processing"*, Vol. 9, No. 1, March 1977.
6. Skinner, C. E. & Asher, J. R.: *"Effects of storage contention on system performance"* IBM syst. J., Vol 8, No 4, 1969, 319–333.
7. Burnett, G. J.: *"Performance analysis of interleaved memory system"*, Ph. D Thesis, Princeton university, 1969.