Packet Execution: A New Concept in Multiprogramming

Angel Diez

A Technical Report
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

August 1984

ABSTRACT

Packet Execution: A New Concept in Multiprogramming

Angel Diez

This paper presents a new approach to achieve an improvement of the cost/performance ratio of a multiprogramming workload running on a single-CPU, via a multi-micro-processor system. The proposed approach is based on the partitioning of programs into blocks called packets. These packets travel in a multiprocessor network structure seeking the first idle micro-processor, where the code performs the required function (execution, I/O, storage, etc). Should the peak workload increase, the multiprocessor machine can easily be upgraded by adding processors in a modular fashion.

PREFACE

The completion of this report is due to three main and only factors. First, to the generation of a concept, brought to paper-life by stubborness, belief and a reasonably broad experience with packet-switching processors used in data communications. Second, to the constant support of Dr. Bipin Desai. Without his positive influence, the concept of Packet Execution would probably have remained buried within the boundaries of my skull. Third, to the perseverance and support of my wife Linda, whose smile was the best medicine against the common frustations encountered during the project.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 BACKGROUND.

The idea of distributing the power of a large CPU into a multitude of smaller processors that perform individual tasks, while offering a better performance as a whole, is not new. P. H. Enslow (ENS-1) identifies 63 multiprocessors of different kind commercially available. The multitude of smaller, experimental systems is much greater. Several well documented papers give an insight into the different aspects of multiprocessing. (FULL-1), (JEN-1), (DAV-1).

When distributing a computer system, several questions arise. How should the tasks executed in the uniprocessor be decomposed to run on a set of smaller processors?. Should the processors be loosly coupled or tightly coupled ?. What is the most efficient connection between processors: Bus, ring, crossbar ?. How can the synchronization of processes and overhead be controlled?. These and a multitude of other different questions have been addressed in the literature. Myers (MYE-1) among others stresses the need for matching the hardware structure with the software that is to be executed in it. S. I. Kartashev (KAR-1 &-2), for instance, addresses this problem by designing a dynamically variable multiprocessor system.

L. Svobodova (SVO-1) supplies logical reasons for a well organized set of requirements in a distributed system.

Simulation studies and theory of message passing systems can be found in (MON-1), (HAL-1), (BRY-1) and (JEN-2).

The type of network used to connect the processors has a strong influence on the throughput capability of the multiprocessor system. General studies of network organizations are found in (ENS-1), (BAS-1 &-2), (BER-1), (DAV-1).

Following Švobodova (SVO-1) and Halstead (HAL-1) we have concentrated in this project on fully distributed systems where there is no centralized control. Group of processors having the same authority, decide, by message interchange, on issues that affect a different group of processors. One way of connecting processors in the same group is by a loop or ring. The advantage of this type of connection is that loop protocols are easy to implement and expand. Loop networks are analyzed in (JAF-1), (LIU-1), (PIER-1), (NEW-1).

In a fully distributed multiprocessor the operating system loses its identity as the control of resources is done by communications protocols rather than by central decision making. Most operating systems up to date have been built around a kernel and higher level processes that see the hardware in a synchronous and transparent fashion.

Synchronization of processes is vital to avoid deadlock situations. Operating systems that specifically stress this issue are the THE (MCK-1) and the RC-4000 (HAN-1). In this paper we emphasize synchronization by message passing.

Hierarchical structures are analyzed in (GOO-1) and (PAR-1). Implementation of distributed operating systems as a set of processors has been analyzed in (JEN-1), (BUH-1), (MYE-1), (PRO-1), (BRO-1) and (WUL-1). Most of the systems analyzed in these papers show a certain centralization of functions. We emphasize maximum decentralization in a fully distributed system.

Systems that have substantially influenced our thinking are the STAR-OS and MEDUSA operating systems, both running on the Cm* multiprocessor, (JON-1), (JON-2), (SWA-1), as well as the SYMBOL machine and operating system. (RIC-1). All these papers discuss the limitations encountered when trying to distribute functions in a multiprocessor. We feel that these constraints can be minimized with the concept of Packet Execution.

2.0 PACKET EXECUTION: BASIC CONCEPTS

In a typical data processing shop the main CPU is shared by programmers and users to test, edit and compile programs as well as to run production applications. At one point in time, many different programs are being scheduled and run by the centralized operating system and CPU. Bottlenecks occur during peak periods and the typical answer to heavy CPU loading is to acquire either another tightly-coupled CPU or a faster CPU. In both cases, the cost of upgrading is fairly high.

We could reduce the CPU load by providing each programmer with a stand-alone microcomputer so that the editing, compiling and testing of individual programs could be done on these units. Unfortunately, this approach is rarely cost-effective because testing of programs usually requires access to large, centralized code modules and data files. The amount of main memory and disk space associated with the individual microcomputers would, most likely, be prohibitive.

We could also reduce the CPU load by executing programs or portions of programs on a multitude of small processors. To achieve this task in an effective manner, the processors should operate asynchronously. This means that no dependencies should exist between the different execution units. More important, the programs have to be decomposed in blocks that are self-addressable. (We will call these blocks **packets** from now on.) In other words, the programs or parts thereof should be accessible to any processor in the machine regardless of their position in main memory or disk. In this way, centralized files and data bases are accessible to all the users, while the task of executing programs is uniformly shared by the different execution units.

These execution processors only require a fraction of the speed and operating system overhead of the centralized CPU. Naturally, the overhead associated with routing the individual packets to the different processors, increases. We feel, however, that this communications overhead can be minimized with an adequate design.

To illustrate our approach, let us consider programs and data files structured in the following way:(see Fig. 1)

The object code of program A is decomposed in blocks called Instruction Packets (IP). Each IP references variables, subroutines and data files whose descriptors are contained in Environment Packets (EP). Data files are also decomposed in Data Packets (DP). To run program A we start by loading the first IP and EP of the program as well as the data packets referenced in the EP. These are IP1A, EP1A, DP1 and DP2. This set of packets can run on a stand-alone execution processor and form a group called **Packet Group.**

To speed-up execution, we also want to load in memory all the packets that are most likely to be dispatched next. These are referenced in the EP1A packet: IP2A,EP2A,IP3A,EP3A,DP3 and DP4. This set of packets, together with IP1A, EP1A, DP1 and DP2, form the **Packet Group Set** of EP1A.

The abstract machine that can run programs decomposed in this fashion is illustrated in Fig. 2. We call it PACOS (Packetized Computing System). Processors are interconnected by a ring network. Vertical rings will be called **Slices**. Horizontal rings will be called **Loops.** When we refer to the function that each Loop performs, we will be talking about Levels. There are four Levels in PACOS, each Level consisting of several peer processors.

1. The User Manager Level (UM Level), monitors (and controls on exceptions) the execution flow of each program. It consists of several User Manager Switches (UMS) and Processors (UMP).

2. I/O is controlled by the Device Manager Level. (DM Level) It consists of Device Manager Switches and Processors.(DMS-DMP)

3. The Memory Manager Level (MM Level), consists of several peer memory units, each one controlled by a Memory Manager Processor (MMP) and a Memory Manager Switch (MMS). Each Memory Manager can hold several self-addressable packets that may belong to different programs.

4. The execution unit consists of several peer Execution Manager Processors, each one executing a Packet Group at a time. A local memory holds the Packet Group under execution. Each EMP interfaces with PACOS via an EMS (switch).

The general processing flow of packets within PACOS is explained below, taking as an example the program A of Fig.1. To simplify the description, a single Slice will be considered. (see Fig. 3). EP and DP packets are routed to the desired destination by means of Control Packets (CP). The switches (UMS, DMS, MMS and EMS) scan the header of the arriving Control Packets and route them accordingly. Control Packet commands are explained in Section 5.3.

Initially, a user requests to run program A from a terminal attached to the DMP.

1.- The DMP translates this request into a CREATE(A) control packet that is sent to the UM Level.

2.- The UMP accepts the packet and sends a GET(EP1A, from UM to MM) control packet requesting EP1A from memory.

3.- As the EP1A packet does not exist in memory, the MMP retrieves it from disk by sending the stream: GET(PKT.GRP.SET of EP1A, from MM to DM)

4.- The DMP accepts the packet, retrieves the Packet Group Set of EP1A and generates the packet: STORE(PKT.GRP.SET of EP1A, to MM).

5.- The MMP stores all packets of the PKT.GRP.SET and sends the stream: SEND(EP1A,GLB.DP, to UM). GLB.DP is a DP packet holding all the global variables of program A. (See Section 6, p.42). This Global DP packet is only required once at creation of the program object.

6.-When the UMP receives EP1A plus the Data Packet that contains all the Global variables, it generates a request to commence the execution of the first Packet Group, by sending an imbedded SEND control packet: SEND(SEND(PKT.GRP.1A, to EM) to MM).

7.- The MMP accepts the packet and generates SEND(PKT.GRP.1A, to EM):

8.- The EMP accepts the Packet Group and executes it. Once execution is finished, the EMP generates the block: UPDATE(EP1A, to UM)

9.- The UMP accepts the block and updates the EP packet (which provides information about the current status of the program) as well as any global variables that might have been affected, before sending the stream: UPDATE(EP,DP.PKTS, to MM) + SEND(SEND(PKT.GRP.2A, to EM) to MM) + GET(PKT.GRP.SET of EP2A, to MM).

10.-The MMP accepts the block, updates all EP, DP packets affected and sends the stream: SEND(PKT.GRP.2A, to EM). The second Packet Group is sent to the EM Level for execution. If PKT.GRP.SET of EP2A does not exist in memory, the MMP retrieves it from disk by sending: GET(PKT.GRP.SET of EP2A, to DM).

11.-The EMP accepts PKT.GRP.2A and executes it.

12.-The DMP retrieves PKT.GRP.SET of EP2A, requested by the MMP in step 10, and sends: STORE(PKT.GRP.SET of EP2A, to MM).

13.-When the PKT.GRP.2A has finished executing, the EMP sends: UPDATE(EP2A, to UM).

At this time the cycle repeats, going back to step 9, until the end of execution of program A is reached.

The packet flow has been described for a single Slice and a single program. Naturally, multiple slices will speed-up the execution of several jobs or tasks running in a multiprogramming environment. The transfer of packets is done via parallel buses, at very high speed. We figure that the overall performance of the machine will be quite high, as long as the delay caused by transfering packets remains a fraction of the time spent in processing chores such as: a) Execution of a Packet Group in the EMP. b) Storing/retrieving packets in the DMP. c) Storing/retrieving packets in the MMP. A simulation of the packet flow will be quite useful in assessing the consistency of our thinking.

The proposed scheme offers several advantages:

1. Reliability: If any unit fails, the rest of the system continues working normally.

2. Modularity: As all unit in one group (or level) are similar in construction, expansion and replacement of equipment is straight forward.

3. Cost: It is less costly to produce 1000 units of a VLSI circuit board than 10 larger systems of 100 circuit boards each.

The disadvantages of this structure are:

1. The partitioning of a large program into several packets that run on different processors in an efficient manner, is still a challenging area of research.

2. A great deal of care has to be placed in the design of the communications links and routing algorithms, in order to avoid a degradation in performance due to transmission delays.

Fig. 1.- THE PACKET GROUP

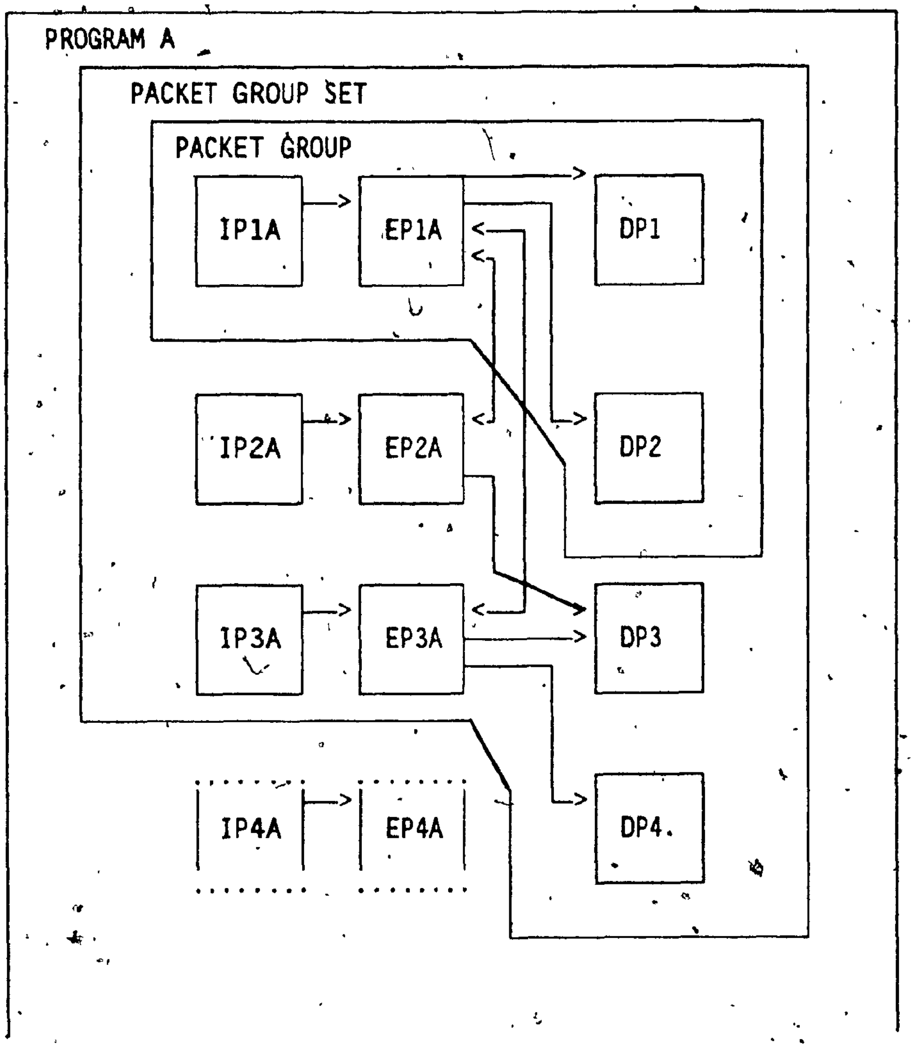


Fig. 2.- PACOS ARCHITECTURE

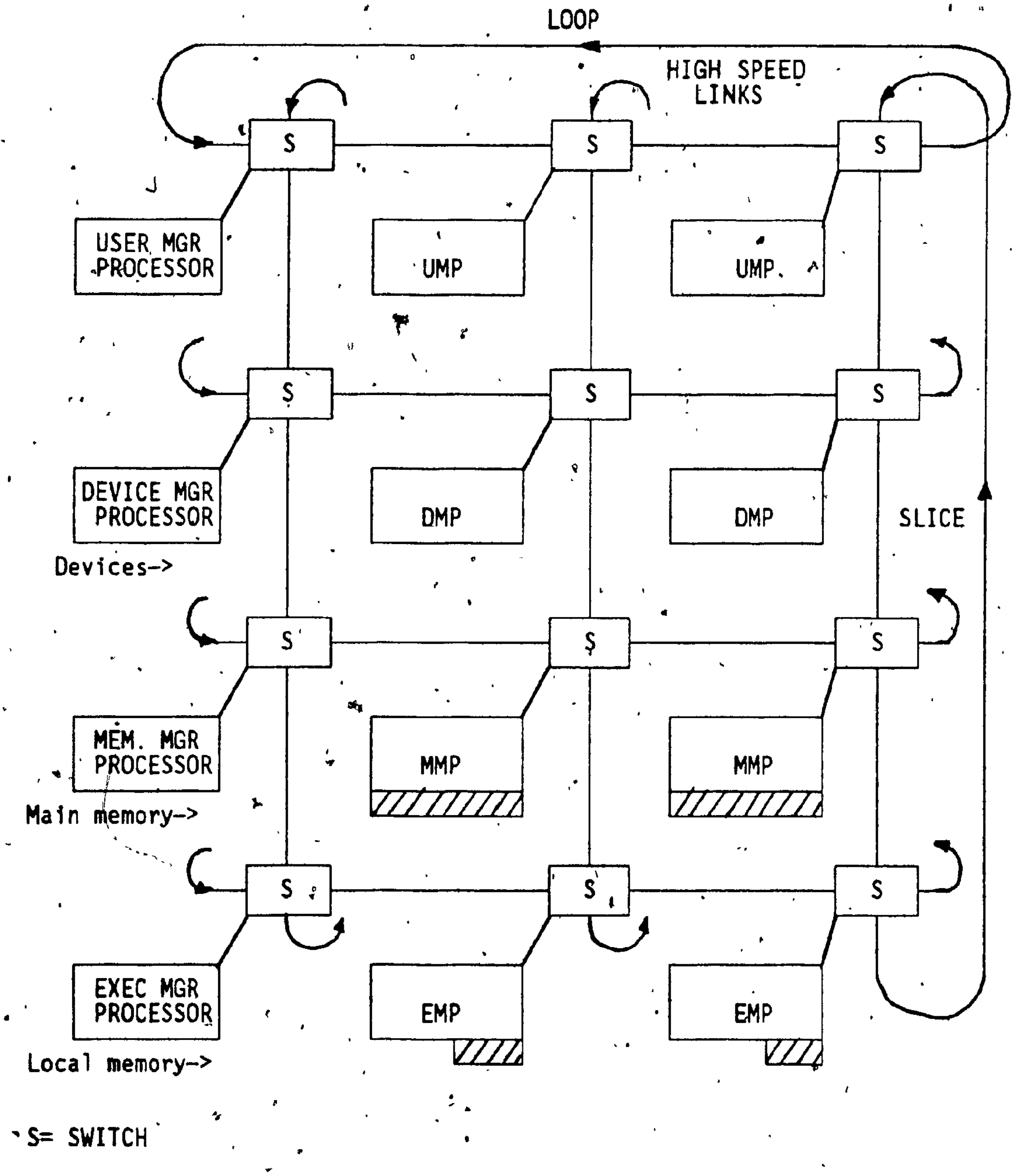
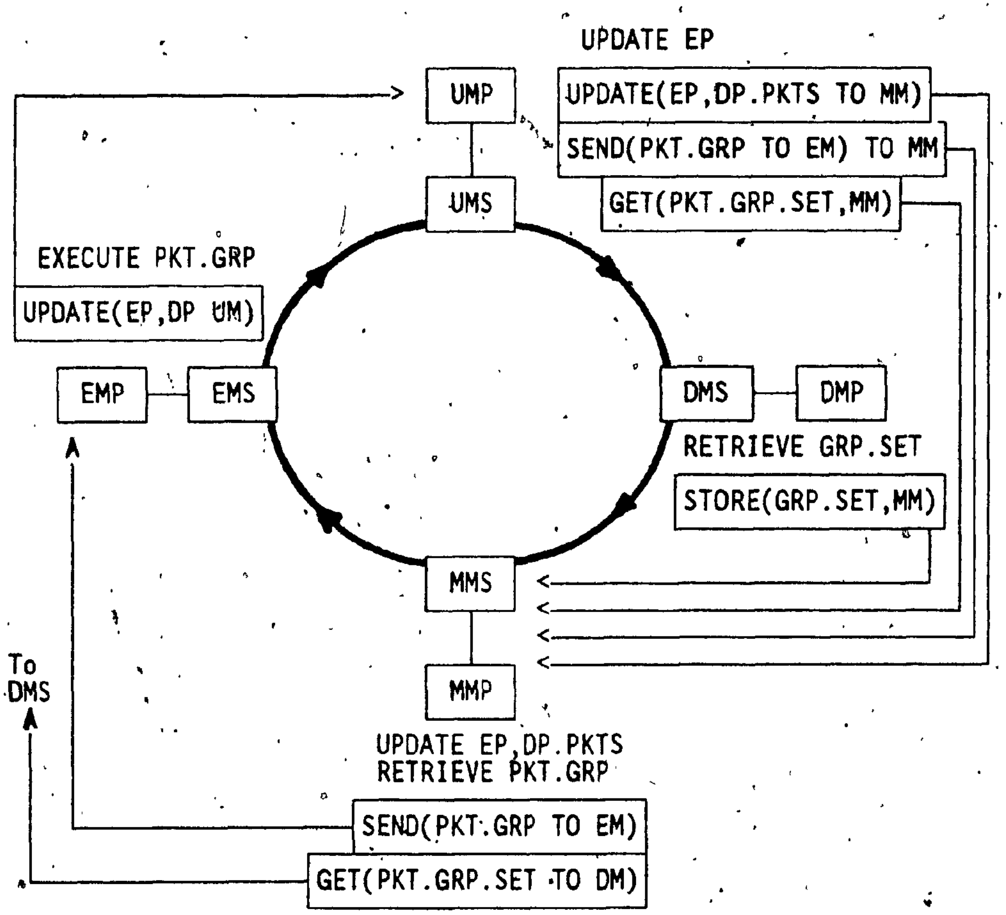


Fig. 3.- PACKET EXECUTION (ONE SLICE)



3.0 THE PACKET GROUP

A **Packet Group** is defined as the group of packets that execute in a single Execution Manager Processor. It consists of:

- One Instruction Packet (IP) that holds the object code.
- One Environment Packet (EP) that holds the variables, labels and data referenced in the code.
- One or more packets Data Packets (DP) that hold data.

Packets flow through the distributed machine by means of switches and commands inserted in Control Packets (CP).

Packets have a unique identifier that consists of the packet type, the object number and the packet number. For example, to uniquely identify the EP packet (code 01), number 15 of object number (00...01111101100000110) , we simply have to refer to the following packet ID:

0	2	32	48 … 63
01	00... ..01111101100000110	00.. .. 01111	
EP	OBJECT NO. 30 bits	PKT NO. 16 bits	

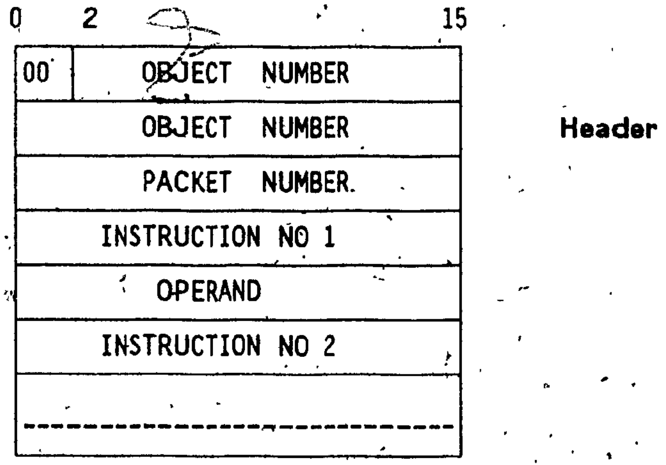
To illustrate the operation of PACOS, a specific machine organization will be analyzed.

We consider a machine that has the following addressing capabilities: Object address : 30 bits.(This gives one billion unique objects). Packet address: 16 bits. (This gives 64 kPackets/ object). IP memory size: 16 bits. (This gives a maximum of 64 kwords/packet). EP memory word size: 64 bits. Data memory word size: 64 bits. (six tag bits plus 58 value bits).

3.1 INSTRUCTION PACKET.

This type of packet is identified by the code 00. The packet has a header consisting of the packet type (00), the object number (30 bits) and the packet number (16 bits). The object code follows the packet header. Operation code occupies one word (16 bits) and operands occupy also one word (16 bits). As the machine is stack oriented, a maximum of one operand per instruction is necessary. The structure of the IP packet is shown in Fig 4.

Fig. 4.- Instruction Packet



3.2 ENVIRONMENT PACKET

This packet is identified by the code 01 and contains all references associated with the respective IP. Fig. 5 shows the structure of the EP. Every IP has a corresponding EP that holds all the variables referenced in the IP. One EP, however, can be associated with several DP's (Data Packets), depending on the data being referenced. The EP packet has a packet header that contains the type code (01), the object number (owner of the packet), the packet number and the starting address. The starting address is to be given to the EMP (execution manager) so that the related IP can start executing on the right local address.

Following the header, all references are inserted in the packet. Each reference is defined with two-64 bits words. The first word contains the name of the identifier (in a standard code such as ASCII). This information is important to notify the user dynamically of any errors during execution. The second word contains the Identifier Control Word.(ICW). The ICW contains all necessary information to locate the value of the identifier being addressed. Table 1 shows the structure of the ICW.

Fig 5.- Environment Packet

0	2	32	48 63
01	OBJECT NUMBER	PKT NUMBER	START ADDRESS
IDENTIFIER NAME			
IDENTIFIER CONTROL WORD (ICW)			
IDENTIFIER NAME			
IDENTIFIER CONTROL WORD (ICW)			

ICW

0 7 32 48 63

0	c	t	value
1	x	y	z

TABLE 1.- CONTENTS OF ICW

FIELD	BITS	VALUE	DESCRIPTION
h (home)	0	0	Value of identifier is contained in the ICW (bits 7-63)
		1	ICW contains descriptor to the identifier (bits I-63)
c (class)	1-3	000	Global variable
		001	Local variable
		010	Array pointer
		011	Label
		100	Procedure
		101	Parameter
		110	Program segment
t (type)	4-6	000	Bit
		001	Boolean
		010	Character
		011	Fixed
		100	Float
		101	Subscript
x	1-31		Identifies the object number of the reference.
y	32-47		Identifies the Packet number of the referenced word.
z	48-63		Identifies the address of the referenced word inside the packet.

If the reference is local (bit 0= 0) the value is contained in bits 7-63.

3.3 CONTROL PACKET

This type of packet is identified by the code 10 and contains commands, responses and notifications addressed to other processors. The header contains the type code (10), the control command, the object number of the originating process and the packet number originating the command. The destination level number, destination object number (30 bits) and the destination packet number (16 bits) complete the control packet. The originating object number of the control packet may differ from the originating object number of the packet in transit.

CP packets are attached to the header of IP, EP, and DP packets, to route them properly. Fig 6 shows a sample Control Packet.

Fig 6.- Control Packet

0	2 15
1 0	COMMAND
d	OBJECT NUMBER (dest)
	OBJECT NUMBER (dest)
	PACKET NUMBER (dest)
o	OBJECT NUMBER (origin)
	OBJECT NUMBER (origin)
	PACKET NO. (origin)
	PACKET IN TRANSIT

The destination field is described in Table 2.

TABLE 2.- Destination Level Number (Origin)

FIELD	BITS	VALUE	DESTINATION (ORIGIN)
d (o)	0-1	00	User Manager
		01	Device Manager
		10	Memory Manager
		11	Execution Manager

The different types of commands are explained below:

- GET.- Retrieves a certain packet from the destination layer, on behalf of the originating object.

- UPDATE.-Updates the original EP or DP packet according to the information attached to the CP packet.

- STORE.-Stores the EP, IP or DP attached to the control packet in any available MMP or device if the destination is a DMP.

- CREATE.-Creates a new object in the UM layer.

- CLEAR.-Clears the originating packet number from the MMP holder of the packet.

- SEND.-Transfers one or several packets to the destination layer. This packet may contain imbedded control packets.

3.4 DATA PACKET (DP)

The packet header contains the type code (11), the object number (bits 2-31) and the packet number (bits 32-47). Data items (64 bits), follow the header. Each data item is qualified with a 6-bit Tag field as per Table 3. These tags are used when the data item is indirectly addressed by a descriptor in the ICW. The structure of the Data Packet is shown in Fig 7.

Fig 7.- Data Packet

0	2		32	48	63
11	OBJECT NUMBER		PKT NUMBER	not used	
c	t	DATA ITEM			
c	t	DATA ITEM			
c	t	DATA ITEM			

Table 3.- Data Tags

TAG FIELD	BITS	VALUE	DESCRIPTION
c (class)	0-2	000	Global variable
		001	Local variable
		010	Array pointer
		011	Label
		100	Procedure
		101	Parameter
		110	Program segment
t (type)	3-5	000	Bit
		001	Boolean
		010	Character
		011	Fixed
		100	Float
		101	Subscript

4.0 PACOS ARCHITECTURE.

An abstract architecture of the PACOS machine that will execute a program submitted in packetized form is illustrated in Fig. 2.

The packet switching network is composed of switching processors connected by two half-duplex rings. Processors at the same level have similar characteristics and form a LOOP. Processors located on the same vertical ring form a SLICE. (In theory, we can build a PACOS computer with a single Slice, although it would be very inefficient).

Packets originally reside in mass-storage devices (At the DM Level). Upon request, packets are transferred to the Memory Manager Level where they remain until cleared by CLEAR commands. Execution of a Packet Group is performed asynchronously. The packets of the group are transferred to the Execution Manager Level (EML). The EP packet will actively seek an idle EMP by checking each EMS (switch) in the EML Loop. Once an EMP is found idle, the Packet Group will grab it and will execute in it. After execution this EMP becomes idle again. If one processor fails, automatic reconfiguration occurs and the machine continues processing normally.

The overall PACOS network resembles a toroid where information flows always in the same direction. The transmission of packets from node to node is done by 16 and 64-bit parallel buses, that accomodate the width of the different packet types. Buffers are used to hold the packets before they are released and after they are received by a switch. An additional signal in the inter-nodal bus serves as control line to indicate FREE or FULL BUFFER conditions.

4.1 THE USER MANAGER LEVEL.

The User Manager Level consists of several User Managers that are peer to each other both physically and logically. Each User Manager Processor (UMP) interfaces with the rest of the machine via a User Manager Switch (UMS). This switch is a routing unit that maintains routing algorithms for incoming and outgoing packets according to the status information supplied to or from the UMP illustrated in Fig 8. The basic interface is defined as follows:

1. OK.- This signal is supplied by the User Manager Switch (UMS) after having received approval from the other UMP's regarding scheduling priority and table updates of the object just created. If the object just created is a program, the UMP will initiate its execution.

2. UPDATE.- This signal has an initial OFF status and is turned ON as soon as a request for updating the EP or tables arrives. The signal is turned OFF as soon as the receving UM has updated the appropiate EP packet and tables.

3. UPBUS.- This interface transfers packets between the UMP and the switch.

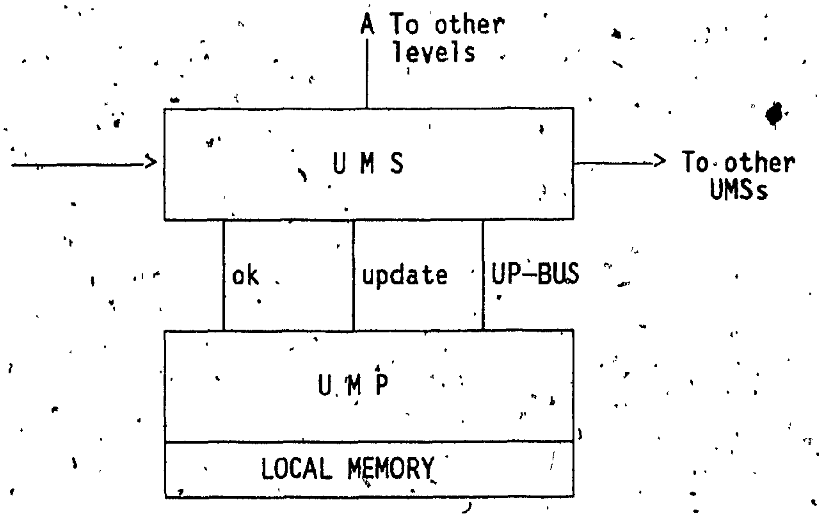
A picture of the UMP-UMS interface is shown in Fig 8. The dynamic behaviour of the UMP is described in pseudo-code form in the next page.

The UMS makes decisions based on the status of three tables: a)UMLTABLE maintains a record for all the existing objects in PACOS. b) UMPTABLE(n) maintains a record for all the existing objects that have been created by the UMP number "n". c) UMSTABLE(n) maintains the routing tables for all incoming packets.

When an UPDATE command is received both UML and UMP tables are updated. When a packet arrives at the UMS its destination is checked and it is routed according to the UMSTABLE.

The operation of the UM is as follows: Initially, the UM is free to accept any objects for creation and dispatching (if object is a program to execute). As soon as a CREATE request is received, the UMS will decide whether the new object can be accepted by this UMP or not. If not accepted, the UMS will simply pass the CREATE request to the next UMS. If accepted, the UMS will send an UPDATE command to the other UMP's in order to update their UMP-UMLTABLES. The UPDATE command is returned to the UMP with the acceptance. At that time The UM switch sets the OK flag ON and the UMP requests execution of the first Packet Group by sending an imbedded SEND command to the Memory Manager, MM. The UMP also requests the next Packet Group Set from the DML with a GET command. This request is done in parallel with the execution of the current Packet Group. At program creation time, the DML will send to the UMP a special DP packet containing the Global variables of the program, GLB.DP. The UMP keeps on receiving EPs as Packet Groups get executed, and updating the EPs, DPs and global variables. When the last EP is received, the UMP sends an UPDATE command to the other UMs indicating that the just finished object will be removed from the UMP-UMLTABLES, if the other UMPs do not need this object.CLEAR commands are also sent to the MML to clear the memory space used by the terminated object.

Fig 8.– USER MANAGER



4.2 USER MANAGER DYNAMIC FLOW.-

```
USER MANAGER
Begin
Do forever
If CREATE= True then
      If object is accepted by UMS then
         Begin
         OK:= False
         SEND (UPDATE UML-UMPTABLES, next UMS)
         If OK= true then
               Begin
               GET(PKT.GRP.SET,MML)  ** Get next pkt grp set from MM **
               SEND(SEND(PKT GRP, EML)  MML)
               End
         End.
If UPDATE= True then
       Begin
       Update(table or EP,DP)     **   Table is updated by the UMP **
       If origin of Update is this UMP
            then OK:= True else        ** Acknowledging returned Update *
            SEND (UPDATE, next UMS)     **  Update passed to other UMs*
       UPDATE:= False
       If End of PGRM= True then        ** Process finished **
             SEND(CLEAR, object, MML)
       End
End
```

4.3 THE DEVICE MANAGER LEVEL.

The DM Level consists of a group of peer Device Managers each consisting of a processor and a switch. Each DMP can handle several devices. The DMP (Device Manager Processor) interfaces with the DMS via flags and buses. It also interfaces with the devices via a data bus. The basic structure of the DM is shown in Fig. 9. The basic interface signals are described below.

1. IO(P).- This flag is set ON by the switch as soon as a request to read or write a packet is received.

2. MORE(n).- This signal is to indicate that the Device "n" cannot accept I/O requests other than for the file currently being accessed. It puts the device into a WAIT state and the device will not be FREE until the MORE flag is set OFF.

3. DPBUS.- This bus transfers packets between the DMS and the DMP.

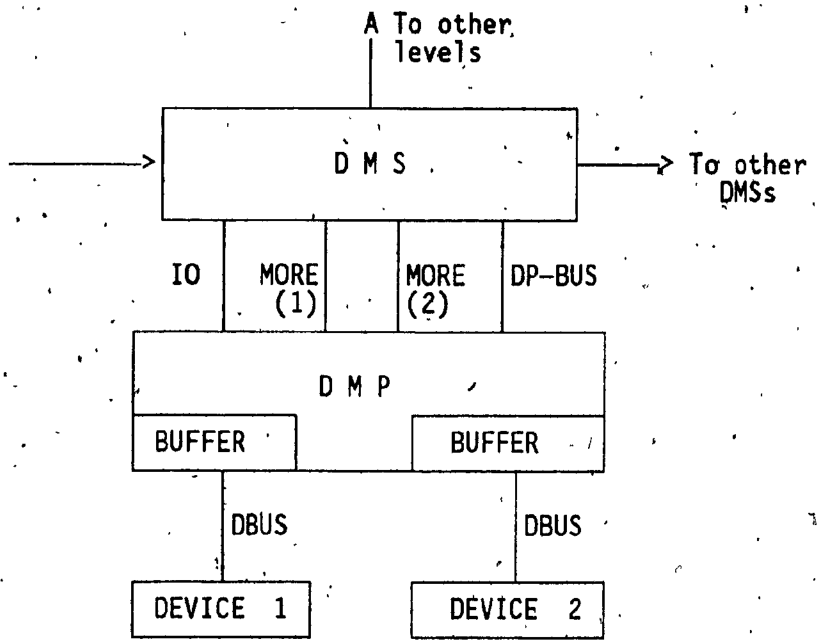
Devices are given a permanent, protected, object number at system generation time. Files are assigned an object number at creation time. Therefore writing (reading) to (from) devices, such as terminals, lines, printers, etc, or files is only a matter of mapping object number to physical location of the desired resource. This mapping is done by the DMLTABLE and the DMPTABLE, maintained by the DMS switches. The DMLTABLE maps object number, packet number, to DMP number. The DMPTABLE maps object number, packet number, to physical device and location within the device.

For instance, to print a block of data to a specific printer, the command

SEND(Dest= Printer object number) is sent with a Control Packet to the DML. The first DMS that receives the CP packet will scan its DMLTABLE and decide whether the printer device belongs to its DMP. If so, it will scan the DMPTABLE to identify the device; otherwise the DMS will pass along the CP packet to the next DMS.

Three basic states, FREE, WAIT and BUSY are assigned to each device. The device is considered FREE when it is ready to accept IO requests from ANY object. While in the BUSY state, the DM is transferring data to (from) the device. If the packet being read or written is part of a sequential transfer that cannot be interrupted, (such as a print file), the MORE flag will be set and the device will switch from BUSY to WAIT state and viceversa until the IO request has been completed. At that time the device goes into the FREE state. If an IO request is received for a device that is in a BUSY or WAIT state, this request will either : a) Wait in the device queue until the device is available or b) Be passed to the next DMS. The dynamic flow of the DMP is explained below in pseudocode form.

Fig 9.- DEVICE MANAGER



4.4 DEVICE MANAGER DYNAMIC FLOW.

```
DEVICE MANAGER
Begin
Do forever
BUSY:= False
WAIT:= False
FREE:= True
Repeat
     If IO(P) = true then
     Begin
     WAIT:= False
     FREE:= False
     BUSY:= True
     DONE:= False
     DOIO          ** The IO is being done  **
     DONE:= True
     If MORE= true then  ** Sequential transfer **
          Begin
          BUSY:= False
          WAIT:= true
          End
     End
until MORE= False

End.
```

4.5 THE MEMORY MANAGER LEVEL.

The Memory Manager Level consists of several peer Memory Managers (MM) each one having a Memory switch and a processor (MMP). Each MMP has the responsibility of storing and retrieving packets of information stored in main memory. Each packet has self-addressing capabilities. Therefore, a file can be stored in different memories belonging to different MMs. Each MMP keeps status tables of the packets and files kept by all the MMs. The basic interface between the MMP and the MMS switch is shown in Fig. 10. While a common memory with interleaved MMP modules might have some advantages (particularly in the sharing of common data by different programs), a fully distributed memory approach was chosen. This was to take advantage of two important features of the PACOS architecture: easy expandibility and minimum level of inter-dependencies among processors.

The interface signals and buses are described below:

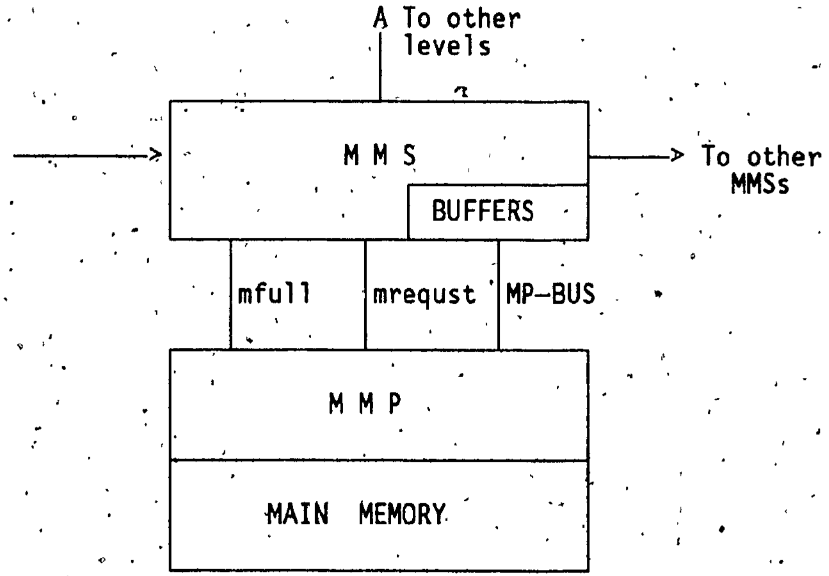
1. MREQUEST.- This flag is set ON as soon as a request to the MM arrives to the switch. When the request has been honored, the flag is set OFF.

2. MFULL.- This flag is set ON when the storage capacity of the MMP has been reached. The flag will be set OFF as soon as a CLEAR(P) control packet is accepted by the MMP. The CLEAR packet deletes one packet of data from main memory.

3. MPBUS.- Bus used to transfer packets between the MMP and the MMS.

Initially, the MM has all main memory available and is ready to store packets. The initial state is FREE. As soon as a request arrives to the switch, the MMS will check its tables to see if the arriving packet should be honored by the MMP. If the request is to update tables (from other MMs), then the MMS will update the appropiate table and will pass the request to the other MMs. If the request is a GET, the MMP will get the requested packet either from main memory or from the Device Manager. If the latter, a CP packet, requesting the packet, will be sent to the DM. If the request is a STORE, the MMP will first check if there is enough storage available. If not, the request will be passed on to other MMs. If the request is CLEAR, the MMS will update the MMPTABLE and will notify this to the other MMs. The cleared packet will no longer exist in this MM. The dynamic behaviour of the MM is described in the following page.

Fig 10.– MEMORY MANAGER



4.6 MEMORY MANAGER DYNAMIC FLOW

```
MEMORY MANAGER
Begin
FREE:= True
BUSY:= False
MAVAIL:= MSIZE
MFULL:= false
Begin
Do forever
If Dest= MML then
Case  COMMAND of
     UPDATE:   Begin
               Update MMPTABLE,MMLTABLE
               SEND(UPDATE, next MMS)
               End
     STORE:    Begin
               If (MFULL or MREQUEST) then
               SEND(STORE, next MMS) else
                  Begin
                  MREQUEST= True
                  Load packet      ** The packet has been stored
                  MAVAIL= MAVAIL-1
                  MREQUEST= False
                  SEND(UPDATE,next MMS)
                  End
              End
     GET:     Begin
              If PKT not in MMLTABLE then
              SEND(GET, DML)  else       ** pass GET to Device Level **
              If PKT in MMPTABLE then
                   Begin
                   MREQUEST= True
                   Read packet
                   MREQUEST= False
                   SEND(PKT, Origin level)
                   SEND(UPDATE,next MMS)
                   End  else
              SEND(GET, next MMS)     ** pass GET to next MMS **
              End
     CLEAR:   Begin
              If PKT not in MMLTABLE then ERROR else
              If PKT in MMPTABLE then
                  Begin
                  MREQUEST= True
                  Clear Packet
                  MAVAIL= MAVAIL+1
                  MREQUEST= False
                  SEND(UPDATE, next MMS)
                  End
              else SEND(CLEAR, next MMS)
              End
      End
End
```

4.7 THE EXECUTION MANAGER LEVEL

The Execution Manager Level consists of several Execution Managers (EMs) each one consisting of a switch and a processor.The processor also has a local memory to hold the packet group. The function of the EMP is to execute the IP packet once the corresponding EP and DPs have been received. The interface between the switch and the EMP is shown in Fig. 11 and consists of the following flags and buses:

1. EPRQST.-This signal is set ON as soon as an EP packet arrives at the EM-switch and encounters the EMP free.

2. GFLAG.-This flag is set ON as soon as all the members of the Packet Group are found in the EMP's local memory.

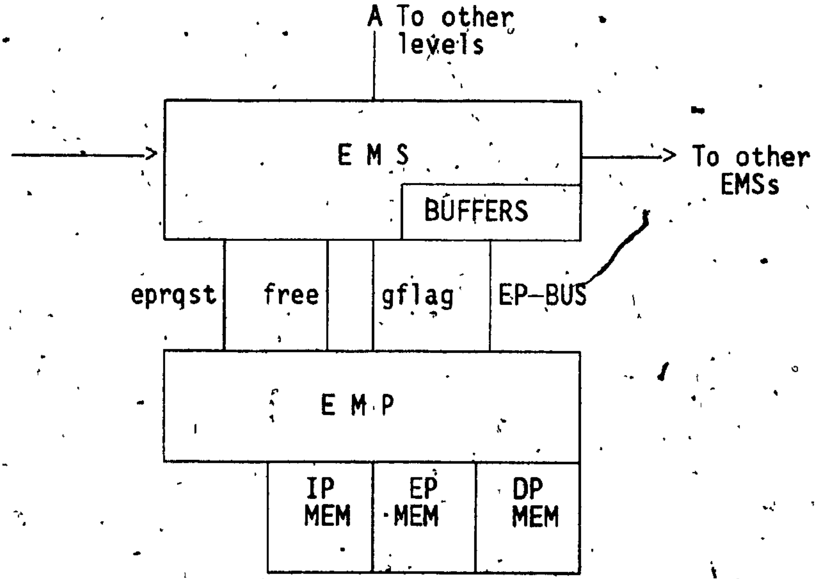
3. FREE.-This flag is set ON as soon as the EMP is ready to accept a new Packet Group.

4. EPBUS.-This bus transfers packets between the EMP and the EMS.

Initially, the EMP is free to receive requests for execution from any User Manager. As soon as a request arrives, if the EMP is free, the switch will accept the EP by turning EPRQST flag ON. The EMP will go into a WAIT state until the IP and DPs forming the Packet Group of the particular EP are received by the switch. At that time GFLAG is turned ON and the IP can start executing. The EMP goes into a BUSY state. At the end of the execution process, the EMP goes into the FREE state again by turning the FREE flag ON. If the EMP is busy when a request arrives, the EMS will pass the request to the next EMS until one idle EMP is found.

The dynamic behaviour of the EM is described in the following page in pseudocode format.

Fig 11.- EXECUTION MANAGER



4.8 EXECUTION MANAGER DYNAMIC FLOW

```
EXECUTION MANAGER
Begin
Do forever
FREE:= True
Case PKTTYPE of
     EP: Begin
         If FREE= True then
              Begin
              FREE= False
              EPRQST= True
              Load EP    *** EP grabs the idle EMP  ***
              WAIT= True
              End  else
         SEND(PKT, next EMS)
         End
   IP,DP:Begin
         If FREE= False then
         If PKTGROUP(P)= PKTGROUP(IP,DP) then
             Begin
             Load IP,DP*** Packet Group being collected **
             If GFLAG= 1 then      *** Packet Group ready  ***
                   Begin
                   Run Packet Group
                   SEND(Update,EP,DP.PKTs, UML)
                   WAIT= False
                   End
             else SEND(PKT, next EMS)     ** PKT.GRP(P)<> PKT.GRP(IP,DP)
             End
      End
End
```

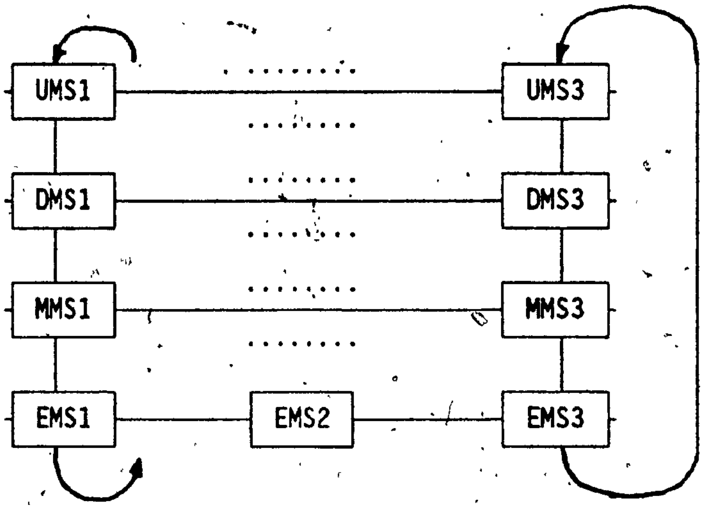
4.9 HARDWARE CHANGES IN THE PACOS ARCHITECTURE.

Although adding one Slice to PACOS is seen as the simple process of inserting a Slice PC board into a circular mother board and updating all the switching tables, it may happen that the reason for expansion is motivated by a very high utilization of only one level in PACOS. (e.g. the EML, but not the other levels). Therefore we should provide for modular units at the Switch-Processor scale, rather than at the Slice scale.

Fig. 12 illustrates a situation where only one EMS-EMP module has been inserted. (only switches are shown for simplicity). By inhibiting the non-used ports of the new switch, correct routing of incoming packets is guaranteed; i.e. packets are not routed to non-existing MMS2, DMS2 or UMS2.

To ensure uniformity, a zero value for table item "Slice 2" should be entered in the switch tables of the first three levels. By introducing hardware in this manner, optimum usage of resources is obtained. In a similar manner, Device Managers, Memory Managers or User Managers can be added when the need arises. In order to reflect the fact that EMS3 and EMS1 are now routing packets to other levels on behalf of EMS2, the routing algorithms would have to be changed.

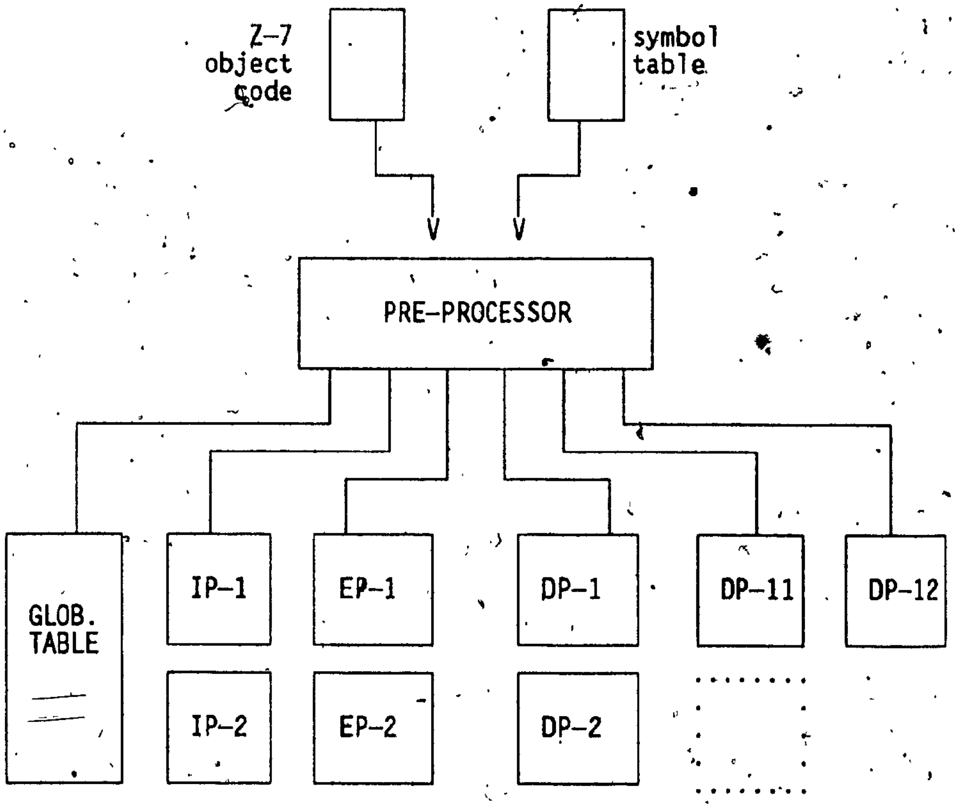
Fig. 12 - ADDING ONE EMS-EMP



5.0 THE PARTITIONING ALGORITHM.

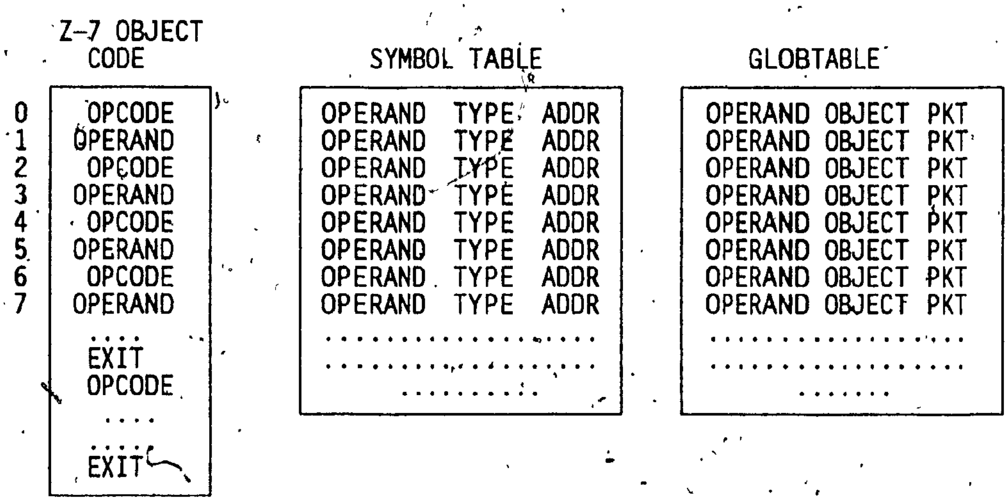
To illustrate the basic concepts of packetizing a standard object code program, a basic machine language will be chosen. This is the stack-oriented language for the academic machine Z-7. (Ref. HOL-1). The main reason to choose this language has been simplicity. Programs written in Z-7 consist of a symbol table and several blocks of code, each block starting with a label entry and ending with an EXIT instruction. A program written in Z-7 language has to pass through a pre-processor that will generate the packets in a format that is understood by PACOS. See Fig 13.

Fig. 13.– PACOS PRE-PROCESSOR



The input to the pre-processor is a list of object code plus a symbol table. The output consists of packets and a table, GLOBTABLE, that references all global variables. Local variables are referenced in the EP packets. See Fig 14.

Fig.14.- PRE-PROCESSOR INPUT.- GLOBTABLE



```
  Z-7 OBJECT
    CODE            SYMBOL TABLE               GLOBTABLE

0   OPCODE      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
1  OPERAND      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
2   OPCODE      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
3  OPERAND      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
4   OPCODE      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
5  OPERAND      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
6   OPCODE      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
7  OPERAND      OPERAND  TYPE  ADDR     OPERAND OBJECT PKT
    ....        ..................      ..................
    EXIT        ..................      ..................
    OPCODE         ..........                 .......
    ....
    ....
    EXIT
```

The algorithm of the pre-processor, necessary to produce PACOS executable code, is described below in pseudocode format.

The main program reads the object code and starts generating the IP, EP packets as well as initializing the GLOBTABLE. If the next word read is PUSH or ENTER or ALLOCATE, the procedure CHECK will read the following operand, and will write it onto the EP packet. If the variable is global, the table GLOBTABLE is also updated. If the variable is a substructure the procedure ICW will generate a DP packet that will contain working space for the allocation of the substructure. A new packet will be generated as soon as EXIT or the maximum packet length has been reached.

PACKETIZING ALGORITHM. (Object SAMPLE is being packetized)

```
ICW: Procedure .
Begin
If Word(I)=Global then            (* Known from the symbol table *)
If Word(I) not in GLOBTABLE then
Generate Word(I) entry in GLOBTABLE
If Word(I)= substructure then
        Begin
        Create new DP packet header
        Allocate locations in DP  (* Space is allocated for Word(I) *)
        End
End

CHECK: Procedure
Begin
I:=I+1
Read Word(I)                          (* Read operand   *)
Write Word(I) onto IP
If Word(I) not in EP then
         Begin
         Write Word(I) onto EP
         ICW
         End
End

Begin                (* Main program  *)
OBJECTNAME:= SAMPLE
MAX:=Maximum Packet Length
Create GLOBTABLE Packet Header
While not EOF(SAMPLE) do
      Begin
      Create new IP Packet Header
      Create new EP Packet Header
      I:=0
      Read Word(I)
      While (I< MAX) or (Word(I)¬=EXIT) do
          Begin
          Write Word(I) onto IP
          If (Word(I) = PUSH) or ENTER) or ALLOCATE) then
          CHECK
          I:=I+1
          Read Word(I)           (* Read opcode *)
          End
      End

End
```

A major research effort should be placed in identifying optimum algorithms for a specific PACOS structure. Packet length, hardware design, program idiosyncrasies, etc. play an important role in this optimization. Because of this complexity, we are attempting to initially work with simple tools and algorithms, and further develop more accurate solutions, while progressing in our observations.

6.0 AN APPLICATION EXAMPLE.

The following program converts Fahrenheit degrees to Celsius. It is shown here in Pascal-like format for easy comprehension, although it runs in PACOS in the stack-oriented Z-7 language (see Ref. HOL-1), as indicated in Appendix A. The input file INFILE contains the number of data items plus the data values in Fahrenheit degrees. The output file OUTFILE contains the values in Celsius degrees.

```
Program CELSIUS (I=INFILE,O=OUTFILE)
Var   I,M: integer
      TEMPF,TEMPC: Array(0..20) of integer

Procedure INPUT   (* IP-1 Packet  *)
Begin
for I:=0 to M do
Readln(INFILE,TEMPF(I))
End

Procedure CONVERT   (*  IP-2 Packet   *)
var   I: integer
Begin
for I:=0 to M do
        Begin
        TEMPC(I):= (TEMPF(I)-32)*5/9
        Writeln(OUTFILE,TEMPC(I))
        End
End

Begin                (*  Main program, IP-3  *)
Readln(INFILE,M)
I:=0
INPUT
I:=0
CONVERT
End
```

The PACOS pre-processor will partition the program CELSIUS into 3 IPs, 3 EPs and one GLOBTABLE (DP-0). The files INFILE and OUTFILE will be partitioned into one DP packet each.

Initially, a user requests to run CELSIUS. The starting address of CELSIUS,(IP3,19), is assumed to be known. The steps outlined in Section 2 of this report will be used to explain the flow of packets in PACOS.

1.-A CREATE(CELSIUS) control packet is sent to the UM Level.

2.-A UMP accepts the job and sends GET(EP3(CELSIUS), to MM).

3.-AS EP3 does not exist in memory, the MMP will attempt to retrieve from disk not only EP3, but rather the complete Packet Group Set of EP3. The PKT.GRP.SET of EP3 consists of IP3,EP3,IP2,EP2,IP1,EP1,DP1(INFILE), DP1(OUTFILE) and GLOBTABLE. (In this case all the packets are in the same Packet Group Set.)

4.-PKT.GRP.SET is sent to MM.

5.-PKT.GRP.SET is stored in memory.

6.-UMP receives EP3 plus GLOBTABLE.

7.-MMP sends PKT.GRP.3 to EM.

8.-EMP executes the Packet Group starting at location 19. At location 20 it jumps to MAIN (IP3, location 0). At location 10 it jumps to INPUT (IP1, location 0). At this time execution stops since IP1 is not locally available. The execution address at the start of a process or at the return from a subroutine, is stored in the START ADDRESS field of the EP header. In our case, the return address (11) is stored in the header of EP3. EMP sends UPDATE(EP3) to UM.

9.-UMP updates the EP packet as well as the GLOBTABLE values affected (In this case, I=0, M=7). A request is sent to update EP3 in memory. A request is also sent to execute PKT.GRP.1.

10.-MMP updates the EP3 packet and sends PKT.GRP.1 to EM Level.

11.-PKT.GRP.1 executes in one EMP until the EXIT instruction is reached.

12.-Not applicable in this example (Next PKT.GRP.SET does not exist)

13.-After execution, the EMP sends the EP1 packet to the UM for updating.

9.-UMP updates the EP packet as well as the GLOBTABLE values affected (TEMPF(I),I). It sends the EP packet to be updated at the MM. It also sends a request to execute PKT.GRP.3 (starting address 11).

The same process continues until STOP is encountered. At that time the UMP will be notified that the program CELSIUS has finished execution and it will be destroyed from the system.

While the process of packetizing a program is one more step in the development process, this should not be seen as an impediment for programmers. The objective is to ensure that production programs that are already in packetized format, such as compilers, utilities, applications, etc, run efficiently and at minimum cost. Automatic utilities that can compile and packetize a high level language, such as Pascal, are feasible and should be transparent to programmers.

PKT.GRP.1 (Celsius) → GLOBTABLE

IP-1 (Celsius)

0 15

	00	Celsius (1)
	Celsius (1)	
	Pkt.No. (1)	
0	PUSH	(5)
1	TEMPF	(0)
2	PUSH	(5)
3	I	(1)
4	FETCH	(6)
5	ADD	(8)
6	GET	(13)
7	STORE	(7)
8	PUSH	(5)
9	I	(1)
10	PUSH	(5)
11	I	(1)
12	FETCH	(6)
13	PUSH	(5)
14	1	(3)
15	ADD	(8)
16	STORE	(7)
17	PUSH	(5)
18	M	(2)
19	FETCH	(6)
20	PUSH	(5)
21	I	(1)
22	FETCH	(6)
23	SUBTRACT	(9)
24	REPEAT	(5)
25	EXIT	(4)

EP-1 (Celsius)

0 32 48 63

	01	Celsius	Pkt. No.1	0
0	TEMPF			
	1	INFILE	DP–1	1
1	I			
	0000101	(value)		0
2	M			
	1	INFILE	DP–1	0
3	1			
	00100111	(value)		1

0 DP–1–INFILE 63

	11	INFILE	1	0
0	M			7
1	TEMPF			23
2				45
3				52
4				3
5				75
6				100
7				65

GLOBTABLE (Celsius)

0 63

11	Celsius	DPG–0	0
M	INFILE	DP–1	
M	CELSIUS	EP–1	
M	CELSIUS	EP–2	
I	CELSIUS	EP–1	
TEMPF	INFILE	DP–1	
TEMPF	CELSIUS	EP–1	
TEMPC	OUTFILE	DP–1	
TEMPC	CELSIUS	EP–2	

PKT.GRP.2 (Celsius)

IP-2 (Celsius)

	0		15
	00	Celsius	(1)
		Celsius	(1)
		Pkt.No.	(2)
0		PUSH	(5)
1		K	(5)
2		PUSH	(5)
3		TEMPF	(0)
4		PUSH	(5)
5		I	(1)
6		FETCH	(6)
7		ADD	(8)
8		FETCH	(6)
		....	
		....	
		....	
		....	
		....	
62		ADD	(8)
63		FETCH	(6)
64		PUT	(4)
65		PUSH	(5)
66		I	(1)
67		PUSH	(5)
68		I	(1)
69		FETCH	(6)
70		PUSH	(5)
71		1	(3)
72		ADD	(8)
73		STORE	(7)
74		PUSH	(5)
75		M	(2)
76		FETCH	(6)
77		PUSH	(5)
78		I	(9)
79		FETCH	(5)
80		SUBTRACT	(9)
81		REPEAT	(3)
82		EXIT	(4)

EP-2 (Celsius)

	0		32	48	63
	01	Celsius	Pkt. No.2		0
0		TEMPF			
	1	INFILE	DP-1		1
1		I			
	0010101	(value)			0
2		M			
	1	INFILE	DP-1		0
3		1			
	0010011	(value)			1
4		TEMPC			
	1	OUTFILE	DP-1		0

DP-1-INFILE

	0			63
	11	INFILE	1	0
0		M		7
1		TEMPF		23
2				45
3				52
4				3
5				75
6				100
7				65

DP-1-OUTFILE

	0			63
	11	OUTFILE	1	0
0		TEMPC		
1				
2				
3				
4				

PKT.GRP.3 (Celsius)

IP–3 (Celsius)

0 15

	00	Celsius (1)
	Celsius	(1)
	Pkt.No.	(3)
0	PUSH	(5)
1	M	(0)
2	GET	(13)
3	STORE	(7)
4	PUSH	(5)
5	I	(1)
6	PUSH	(5)
7	0	(5)
8	STORE	(7)
9	ENTER	(1)
10	INPUT	(2)
11	PUSH	(5)
12	I	(1)
13	PUSH	(5)
14	0	(5)
15	STORE	(7)
16	ENTER	(1)
17	CONVERT	(3)
18	EXIT	(4)
19	ENTER	(1)
20	MAIN	(4)
21	STOP	(12)

EP–3 (Celsius)

0 32 48 63

	01	Celsius	Pkt. No.3	19
0	M			
	1	INFILE	DP–1	0
1	I			
	0000101	(value)		0
2	INPUT			
	1	CELSIUS	IP–1	0
3	CONVERT			
	1	CELSIUS	IP–2	0
4	MAIN			
	1	CELSIUS	IP–3	0
5	0			
	0010011	(value)		0

DP–1–INFILE

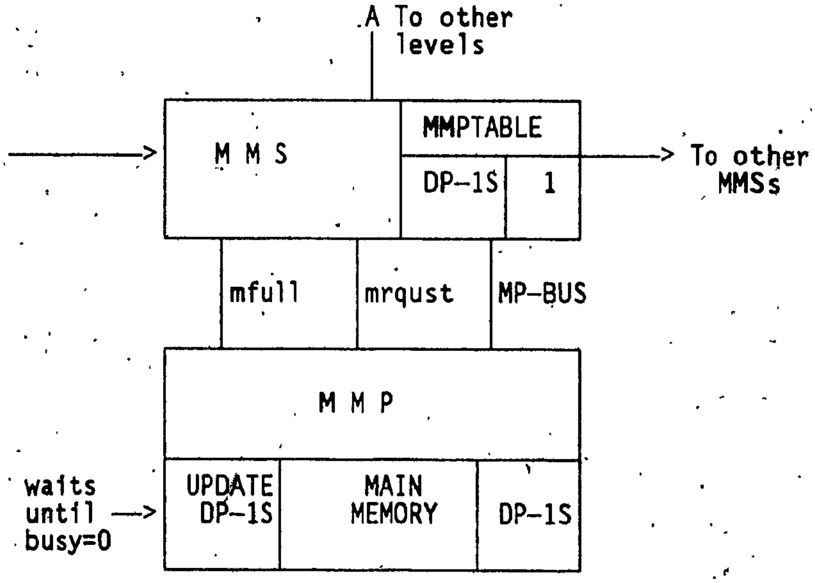
0 63

	11	INFILE	1	0
0		M		7
1		TEMPF		23
2				45
3				52
4				3
5				75
6				100
7				65

7.0 SYNCHRONIZATION CONSIDERATIONS.

Synchronization in PACOS is obtained by a-semaphore-like mechanism. To illustrate how a shared resource is accessed by several processes without danger of deadlock, we will use a shared DP packet as an example. The shared packet,DP-1S, resides in one MMP. (Refer to Fig. 15). The MMPTABLE, located in the MM switch, allocates one entry for this shared packet, that contains a "Busy bit". This bit will be set to 1 as soon as a request to update the shared packet arrives. If, while the bit is set to 1, another update request for the same packet arrives from a different program, this request will be queued in a FIFO queue. The arriving UPDATE packet will be temporarily allocated in a buffer until the "Busy bit" returns to 0. At that time, the waiting request will be honored by the MMP.

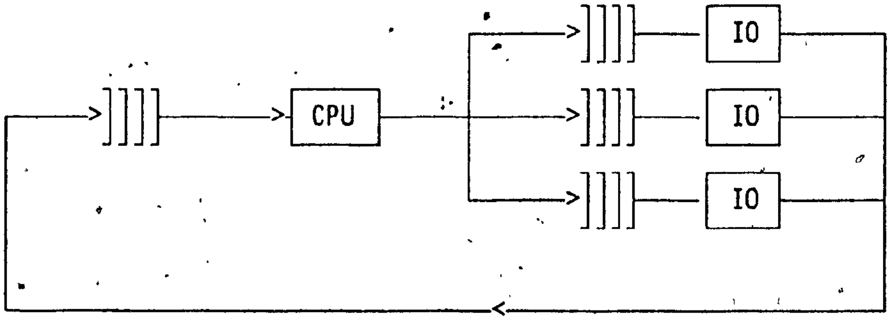
Fig 15.- SHARED PACKET



8.0 PERFORMANCE CONSIDERATIONS.

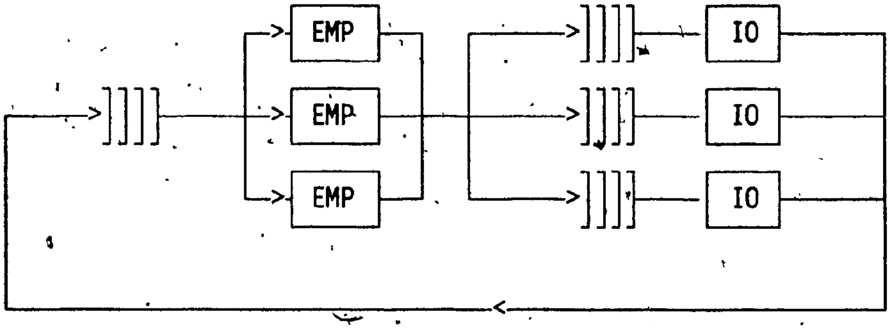
A single-CPU multiprogramming system can be simulated by a set of queues and servers as represented in Fig 16.

Fig. 16.- Single CPU system.



The multi-CPU multiprogramming system PACOS can be represented by a set of queues and servers as per Fig. 17

Fig. 17.- PACOS multi-server queueing model.



To calculate the number of EMP processors necessary to replace a single large processor CPU, the total throughput supplied by both machines will be equated. The same mixture of programs will be used..

A = Throughput of a single CPU. (instr/sec)

B = Throughput of a single EMP. (instr/sec)

TTIME = Average turnaround time per job (secs). It is equal to the average execution time plus the waiting time per job.

ETIME = Average execution time per job. (secs)

$\mathbf{E(t_A)}$ = Execution time of one page in a single-CPU system. $\mathbf{E(t_B)}$ = Execution time of one packet group in the EMP.

L = Number of instructions executed per packet group (or page).

n = Number of Slices in PACOS.

σ_A = Delay caused by memory fetch and data transfer of one page in the single-CPU system. σ_B = Delay of one packet group in PACOS.

φ_A = Delay caused by waiting time spent in the dispatch queue for a single-CPU system. φ_B = Waiting time spent until a free EMP is grabbed by a packet group.

ρ_A = Utilization of the CPU. ρ_B = Utilization of one EMP in PACOS.

In a multiprogramming environment, the average turnaround time of J jobs running concurrently on a single-CPU system will be compared to the average turnaround time of the same number of jobs running on PACOS. We will consider that packets and pages are of the same length.

Each job consists, on the average, of M executable pages or packet groups.

$$J*TTIME = J*M*(\frac{L}{A} + \varphi_A + \sigma_A) = J*M*(\frac{L}{B} + \varphi_B + \sigma_B) \quad \text{or:}$$

$$\frac{L}{A} + \varphi_A + \sigma_A = \frac{L}{B} + \varphi_B + \sigma_B \qquad (1)$$

Using standard queueing theory, (IBM-1, MAR-1), and assuming that the service time of execution processors follows the exponential distribution, we derive the average waiting times φ_A and φ_B.

$$\varphi_A = \frac{\rho_A * E(t_A)}{1 - \rho_A} \quad (2) \quad \text{where:} \begin{cases} E(t_A) = \frac{L}{A} \\ \rho_A = \text{CPU utilization} \end{cases}$$

$$\varphi_B = \frac{P_n(\mu) * E(t_B)}{n*(1 - \rho_B)} \quad (3) \quad \text{where:} \begin{cases} \mu = n * \rho_B \\ \rho_B = \text{EMP utilization} \\ P_n(\mu) = \frac{1 - r(\mu)}{1 - \rho_B * r(\mu)} \\ r(\mu) = 1 - \frac{\mu^n}{n! \sum_{k=0}^{n} \frac{\mu^k}{k!}} \\ E(t_B) = \frac{L}{B} \end{cases}$$

Equation 2 indicates that when the utilization of the single-CPU becomes close to 1, the degradation of waiting time can only be reduced by increasing the CPU processing speed, i.e. reducing the execution time $E(t_A)$. However, when the utilization of each EMP in PACOS becomes close to 1, we can reduce the waiting time by simply adding more slices (increasing "n"), which in most cases is simpler and less costly than upgrading the single-CPU processor.

Equation 1 becomes:

$$\frac{L}{A}\left(1+\frac{\rho_A}{1-\rho_A}\right)+\sigma_A = \frac{L}{B}\left(1+\frac{P_n(\mu)}{n*(1-\rho_B)}\right)+\sigma_B = K$$

where K is the specific cycle time (part of the turnaround time) per page or packet group to be used as a parameter reference. Substituting L as a function of K in the previous equation, we obtain:

$$\frac{A}{B} = n* \frac{(1-\rho_B)}{(1-\rho_A)} * \frac{(K-\sigma_B)}{(K-\sigma_A)} * \frac{1}{(n*(1-\rho_B)+P_n(\mu))} \qquad (4)$$

Equation 4 establishes the performance function of a single-CPU system (in multiples of the basic EMP speed) in relationship to the performance obtained by a PACOS machine of "n" slices while loading both systems with the same multiprogramming level.

Following A.V. Pohm (POH-1), the optimum level of service is obtained when the average execution time per job is equal to the average wait time per job. (memory access + routing delays + page fault timing). In other words, the optimum balance is obtained when neither resource, EMP or rest of the machine, has to wait for the other to finish.

We define the Balance Ratio as :

$$BR = \frac{ETIME}{TTIME} \qquad (5)$$

The optimum BR is 0.5. When BR > 0.5 then the EMP is the bottleneck. When BR < 0.5 then the wait-for-EMP, routing and memory fetching delays are the bottleneck.

From equations 4 and 5 we can derive the ratio of processing speeds of single-CPU versus single EMP as a function of the number of slices in PACOS, "n", in relationship to the optimum Balance Ratio of PACOS.

Although P_n is a function of "n", its absolute value oscillates between 0.5 and 1. Typical value is 0.6. (see IBM-1, pp. 34-40). Since the routing delays of a single-CPU system are in the order of microseconds while in PACOS are in the order of miliseconds, we can estimate that $\sigma_B >> \sigma_A$. Other typical values, used in the simulation of PACOS in Section 11 are: L= 4,000 instructions executed per packet group; B= 100,000 instr/sec.; σ_B= 2 msecs. This delay is estimated based on the fact that a 4 Kbytes packet takes on average, four inter-nodal delays of 0.4 msecs to reach the EM level, plus 0.2 msecs to be fetched from memory, plus 0.2 msecs to update tables in the UM and MM levels.

With BR= 0.5, we have: K = TTIME/M = 2 * ETIME/M = 2*L/B. where M is the average number of packet groups executed per job.

Using K as a parameter, we can plot equation 4, illustrated in Fig. 18. The maximum value of A/B is obtained when n ---> ∞. The result is $(A/B)\max = 1/(1-\rho_A)$, as long as $K >> \sigma_A, \sigma_B$. The maximum effective value of "n" is obtained when A/B = n since, beyond this point, the relative performance of PACOS does not improve with the addition of more slices.

Let us consider an example with the following typical values:
$\rho_A = 0.99$; $\rho_B = 0.9$; K= 0.08 secs/packet group; $P_n = 0.6$. Substituting these values, we obtain:

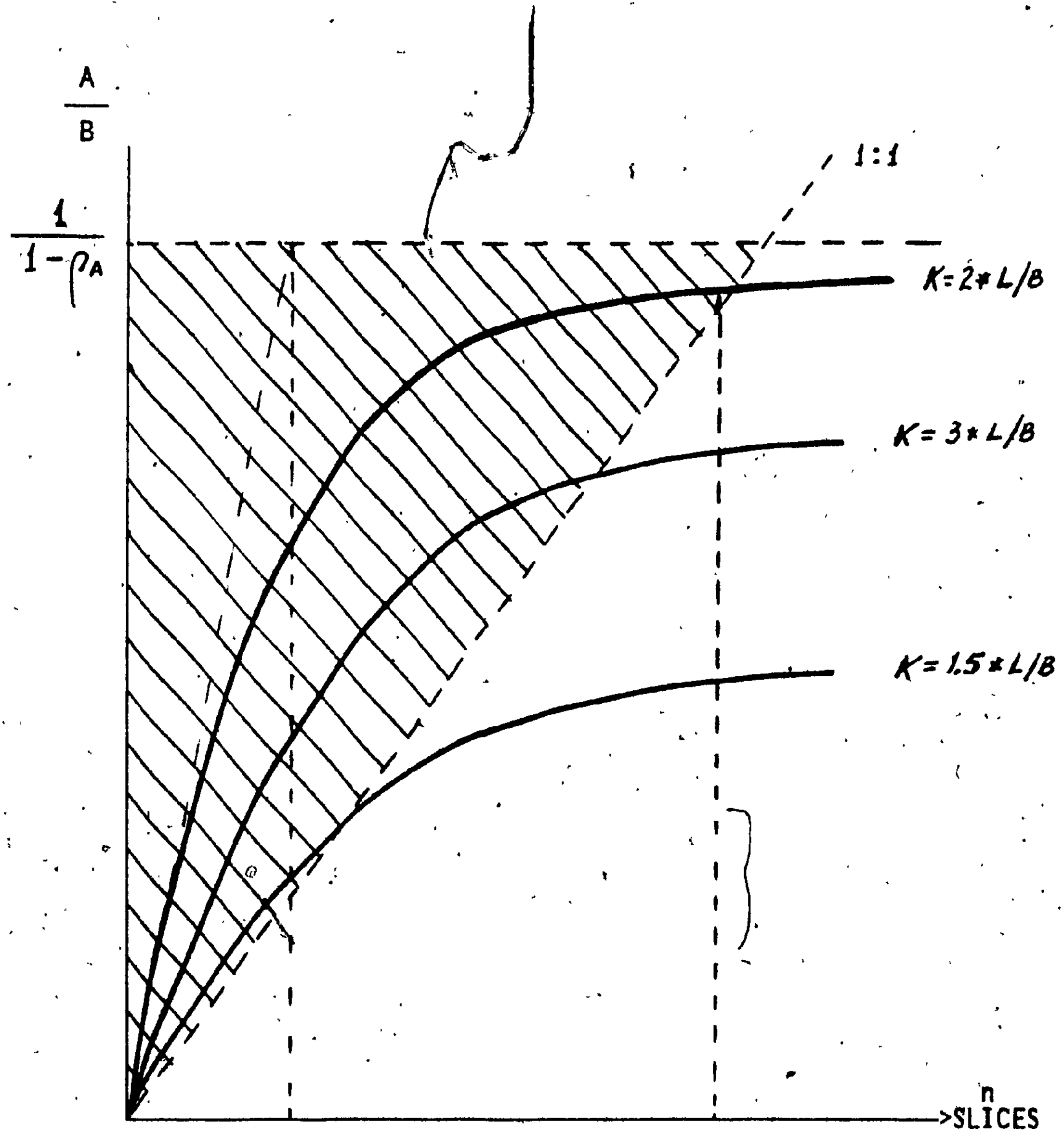
$$\left(\frac{A}{B}\right)_{max} = 100 \quad ; \quad n_{max} = 94 \text{ slices} \quad ;$$

Substituting the same values in equation 4 we obtain the CPU speed required to match the performance of a PACOS machine with "n" slices. For example:

For n = 6 slices, A/B = 50.

These results indicate that a PACOS machine used at the indicated EMP utilization level, will consistently outperform a single-CPU system having a speed ratio equal to the number of slices of PACOS. For 94 or more slices, PACOS' comparative performance will start degrading. The linear margin where PACOS outperforms the single-CPU system is, however, significant.

Fig. 18.- Processing ratio (CPU) vs. No. of Slices (PACOS)



$$n_o = \frac{P_n}{\frac{K-\sigma_B}{K-\sigma_A}\left(1-\rho_B\right)} \qquad n_{max} = \frac{1}{1-\rho_A} * \frac{K-\sigma_B}{K-\rho_A} - \frac{P_n}{1-\rho_B}$$

Several conclusions can be obtained from these curves:

1. The performance curves flatten towards the asymptote $A/B = 1/(1 - \rho_A)$ when $K >> \sigma_A, \sigma_B$;(usually this is the case). The higher the throughput (or CPU utilization), the higher is the range of operation of PACOS.

2. Since we are interested in achieving a performance which is equal or superior to the one provided by a single-CPU system, the effective area of operation of PACOS is the zone where $A/B > n$. Within this area, the PACOS machine outperforms a single-CPU system. For example, with K = 2* L/B (optimum Balance Ratio) and $\rho_A = 0.99$, $\rho_B = 0.9$, only 6 slices are required to replace a CPU having an A/B value of 50.

3. The optimum curve is obtained for K = 2* ETIME/M = 2* L/B, as expected. For higher or lower values of K, the performance of PACOS degrades.

Equation 4 and the corresponding curves indicate that the performance of a PACOS machine, intended to replace a large single-CPU system, is almost linearly proportional to the number of slices, and outperforms the latter, within a large range of processor utilization. These theoretical results are very encouraging and will be compared with the simulation results in Section 11.

9.0 PACOS SIMULATOR

The abstract machine PACOS has been simulated in GPSS, according to the original specifications and the performance assumptions indicated in Section 8. The GPSS program is attached in Appendix C.

The first GENERATE block generates programs, one per GPSS transaction. This program transaction is later split into multiple transactions that, in reality, are packets. These packets move back and forth between PACOS levels until the end of the program execution is reached. At that time, turnaround time, execution time and other statistics of the program are saved for printing and further analysis. Packet transactions have been assigned the following fullword parameters:

PARM. NO.	DESCRIPTION
1.-	Processor (or node) number where the packet resides at a specific time.
2.-	Program No.
3.-	Packet No.
4.-	Packet length (in bytes)
5.-	Destination level. Levels of PACOS are: 0: User Manager 1: Device Manager 2: Memory Manager 3: Execution Manager
6.-	Type of Control Command. Commands in PACOS are:

1: GET

2: UPDATE

3: STORE

4: CLEAR

5: CREATE

6: SEND

7.- Packet Type: 1(EP), 2(IP), 3(DP), 4(CP)

8.- Originating level (see Parm. 5)

9.- Horizontal (Loop) communications link no. It is assigned the number of the processor (PF1) minus 1000 by variable LINK1.

10.- Vertical (Slice) communications link no. It is assigned the value of variable LINK1 plus 400 by variable LINK2.

11.- Holds the number of packets that belong to the same packet group.

12.- Program length. (Initially in number of packet groups executed)

13.- Holds the turnaround time of the job. Used for tabulation purposes.

14.- Holds the clock time of every Packet Group entering a free EMP.

15.- Accumulated number of Packet Groups called for execution.

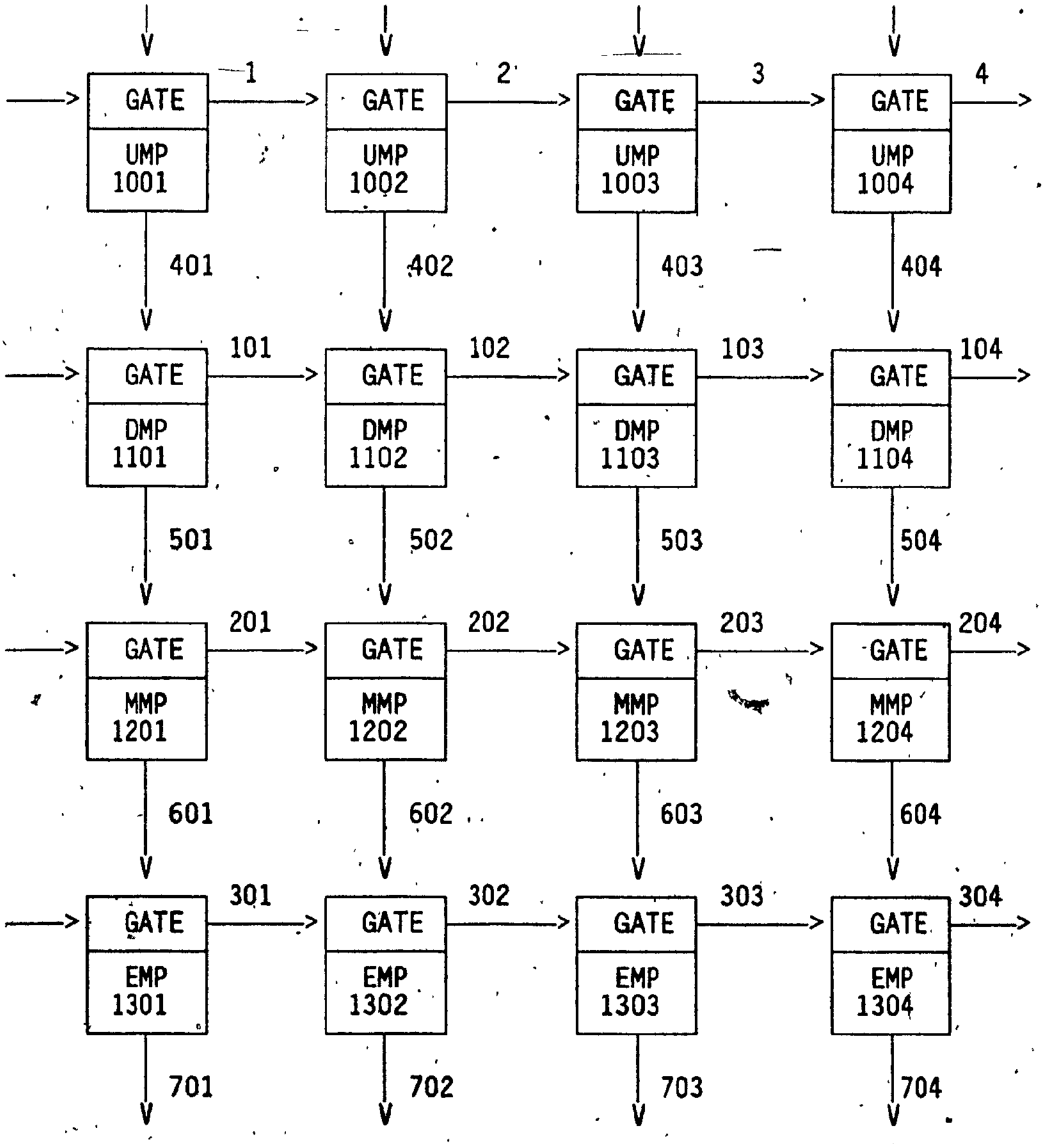
16.- Holds the UMP number creator of the object.

17.- Holds the actual time that the job spent while executing in the Execution Manager Level.

9.1 SIMULATION OF THE SWITCHING NETWORK

Processors and links are GPSS facilities numbered as per Fig 19.

Fig. 19.– MEANING OF FACILITIES



The following limitations are applicable to this structure:

a)The network is half-duplex, ring topology. Packets are transmitted in one direction only. Therefore Slice link no. 801 becomes automatically no. 401 and the last Loop link on level 0 becomes link no. 1.

b) For simulation purposes, a maximum number of 100 slices is available.

c) Packets are transmitted to the next switching processor as soon as the link (facility) is available. A transmission delay (V$DEL) is assigned before reaching the next switching processor. The value of this delay is 100 microseconds for a packet size of 1 Kbytes. The speed of the links is 80 Mbps or 10,000 Kbytes/s. The transmission delay is proportional to the length of the packet.

9.2 SIMULATION OF PROGRAM BEHAVIOUR DURING EXECUTION

The concept of Packet Execution is based on the known "locality of access" characteristic of program behaviour. (MAD-1), (CAR-1). In simple words, it states that the probability of the CPU addressing the instruction previous or next to the current one is very high. Therefore it is advantageous to execute code while keeping in local memory a certain number of previous and subsequent instructions.

It is also known that an adequate design of the memory hierarchy, (cache, main memory, secondary memory) is vital to achieve an adequate optimization of resources. If the memory hierarchy is properly designed, the timing involved in accessing packets from secondary storage (Device Manager), should not degrade the performance of the system since these accesses can be done in parallel with execution of packet groups. (The Packet Group Set is fetched in advance). (see POH-1, pp. 92-116, for a discussion on optimization of different memory hierarchies). Therefore, in simulating PACOS, we will assume that access to secondary storage is completely transparent to the system.

Since our main interest is to compare the performance of a single-CPU, virtual memory system with the one provided by PACOS, we will define "a priori" a packet length of 4,000 bytes, which is a standard page size in existing systems. The sample programs are short (average of 17 packet groups per program) to avoid lengthy GPSS simulation time. Since the same type of programs are run on both single-CPU and PACOS systems, we expect the results to be consistent regardless of program length.

In order to reflect the multiple characteristics of programs during execution, different functions have been defined. The following situations, that usually occur during program execution, will be considered.

- A jump to a loop or subroutine (N cycles). If the loop is outside the current packet, the calling IP will be brought in to execute N times.

- A jump to an instruction located in a different packet or I/O. (Execution of the current packet would stop)

- A jump to a subroutine located in a different packet. In this case execution stops, but the current IP will be called again to execute after the subroutine has finished processing.

- Calls to subroutines or loops, (both inside or outside the current packet), may occur several times within one packet.

To simulate this behaviour we define the following GPSS functions:

- GEN.- Gives the number of packets that will compose a new packet group minus one. (EP is not counted). Usually, a Packet Group will consist of one IP, one EP and one or more DPs.

- PTIME.- Gives the basic execution time of an instruction packet of 1 Kbytes in size. (62 instructions). At a rate of 0.1 MIPS, the execution time will be 1.5 msecs.

- PLEN.- Program length in number of Packet Groups executed. Basic packet size is 4 Kbytes.

- CBND.- Gives the expected number of iterations encountered in a loop. This figure is, of course, very much program dependent. We will use an average of 6 (six) iterations per loop.

- LPPR.- Gives the probability of encountering a jump to an I/O, loop or subroutine located in a different packet.

- LPNO.- Gives the number of jumps to a different packet that the current packet will experience during execution.

We also define several variables. Two of them relate to program behaviour: SPAWN and CPUTM.

SPAWN is defined as the variance of the number of times, over a normalized packet size of 4,000 bytes, that a given IP will be dispatched for execution.

$$\mathrm{SPAWN} = \mathrm{LPPR} * \mathrm{CBND} * \mathrm{LPNO} * (4 - \mathrm{NORM})$$

This variable reflects the fact that the number of packets executed per program decreases when the packet length increases.

We also define the variable CPUTM as the execution time used by one packet group.

$$\mathrm{CPUTM} = \mathrm{PTIME} * \mathrm{CBND} * \mathrm{NORM} / \mathrm{CPTM}$$

where NORM is the normalized packet size in thousands and CPTM is the normalized EMP processing speed in multiples of the basic 0.1 MIPS.(value of B in Section 10)

The simulation program operates in the following way: A number of programs are generated, one every seven msecs. Since all these programs run concurrently in the machine, they represent the Multiprogramming Level, MPL, or load of the system. The GPSS clock runs in microseconds. Each program, at creation time, is assigned a User Manager Processor that will be responsible for monitoring the process. This allocation is done in a round-robin manner.

Each program, (a GPSS transaction), is sent to the Memory Manager Level in the form of the first Packet Group. It splits in several packets according to the value of the function GEN. Each packet of the Packet Group is sent to the Execution Manager Level (Param. No.5= 3). The EP packet of the group searches for the first free EMP available (GATE NU PF1,OSW). This check is provided by the EMS, and if unsuccessful, the EP packet will get routed to the next EMS. After grabbing one EMP, the EP will wait for the rest of the Packet Group to arrive (ASSEMBLE block). Once the Group is complete, execution starts (ADVANCE CPUTM).

When execution of the packet group stops, due to one of the conditions previously mentioned, the original EP packet is routed to the UM level (Parm 5= 0), where the cycle is repeated. The end of the program will occur when the number of packet groups executed is equal to the initially assigned program length plus the accumulated value of the variable SPAWN (PF 15). Table TTIME stores all the values of the turnaround time for each program. Table ETIME stores all the values of the accumulated time that all the Packet Groups of each program spend executing in one EMP. Table NPAC stores all the values of the number of Packet Groups executed per program.

9.3 SIMULATION RESULTS

The first simulation run gives an indication of the variation of average turnaround time as a function of the number of Slices in PACOS. (see Fig. 20). Packet size is fixed at 4 Kbytes. CPTM (EMP processing speed) is fixed at 1 (0.1 MIPS). As indicated in Section 10, Performance Considerations, the optimum Balance Ratio is obtained when the execution time, ETIME, is 50% of the turnaround time. For an MPL (multiprogramming level) of 10 programs, we notice that this situation occurs when the number of slices is four.

With one Slice, the EMP is constantly busy and packets spend most of the time waiting for the EMP to become free.

With eight and more Slices, most of the time is spent in execution of packets in the EMP. (ETIME = 0.8 * TTIME).

The second simulation run (results in Fig 21) gives an indication of the performance that a single-CPU system would provide, using the same load of programs as on the previous run. We simulate this situation by changing the following parameters:

- The inter-nodal delays are reduced to 20 microseconds/packet. (from 400 microseconds on the PACOS machine). This time roughly approximates the register-transfer delays involved in a single-CPU system.

- The memory access time per packet is reduced to 20 microseconds per packet (from 200 microseconds in PACOS).This is based on a memory cycle time of 0.1 microseconds per instruction, which is a typical value for a medium size mainframe. (POH-1, pp. 77-80)

- The speed of the single-slice CPU is increased by factors of 4, 8, 12, 16, 20, 24 and 32.

For CPTM = 4, the turnaround time is worse than having a PACOS machine with four slices. This situation is caused by packets spending too much time waiting for the CPU to be free. The botleneck clears up when CPTM is increased to 8. The Balance Ratio, however, still remains too low. (ETIME = 0.2 * TTIME)

Finally, Fig. 22 indicates the variation of performance of a single-CPU system versus a PACOS machine with "n" number of Slices. The turnaround time has been kept constant at the optimum level of the simulated PACOS system. (BR = 0.5 or TTIME = 2.5 secs.), as well as at lower and higher levels of TTIME. This chart is a combination of the results obtained on Figs. 20 and 21. The close-to-linear relationship demonstrates that the concept of Packet Execution can effectively provide a solution to the problem of upgrading large, general purpose, single-CPU systems.

The resulting values are consistent (allowing for simulation variances), with the theoretical results obtained in Section 10 (equation 4, Fig. 18).

Fig. 20.- Turnaround time vs. No. of Slices

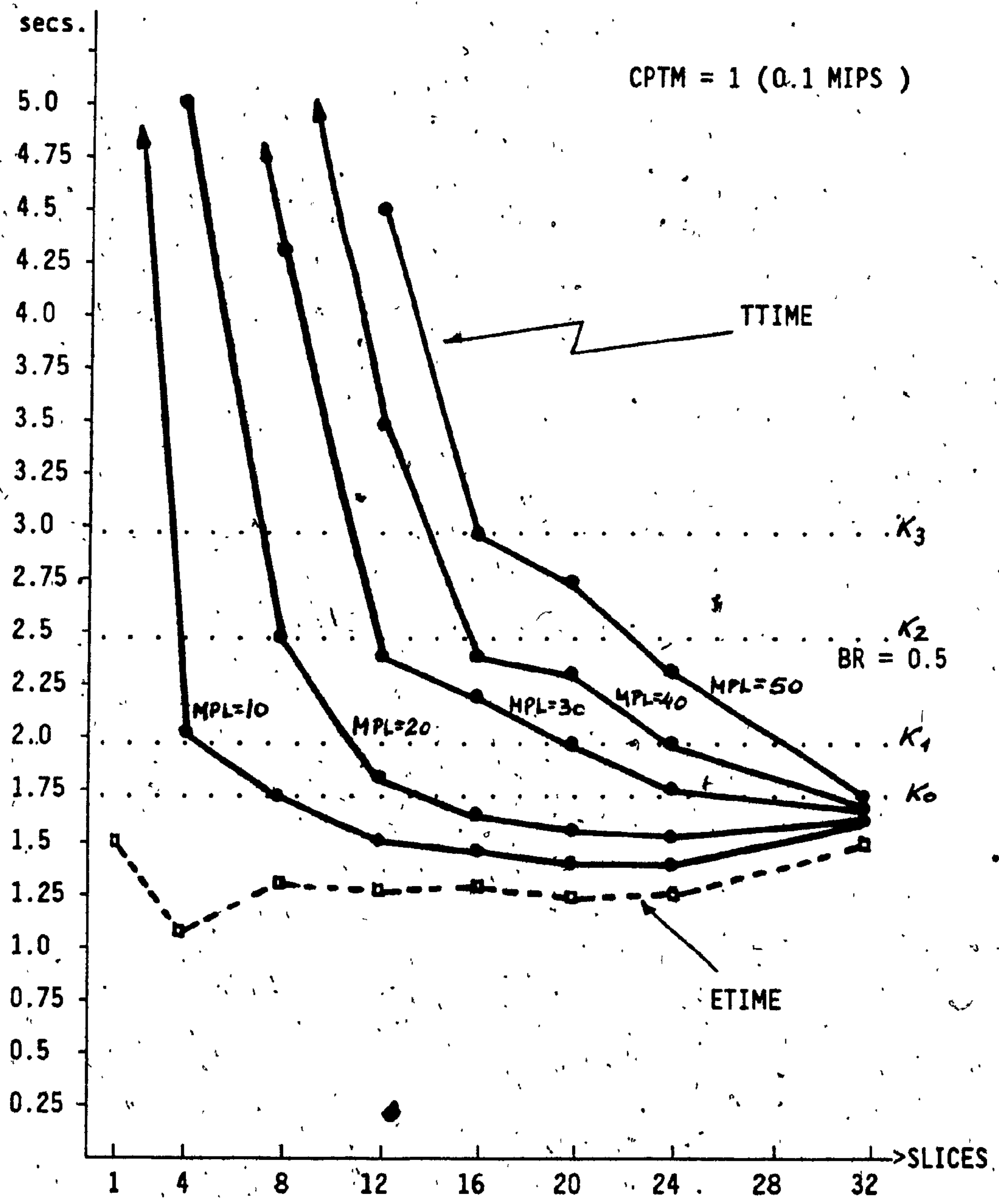


Fig. 21.- Turnaround time vs. EMP processing speed.(One Slice)

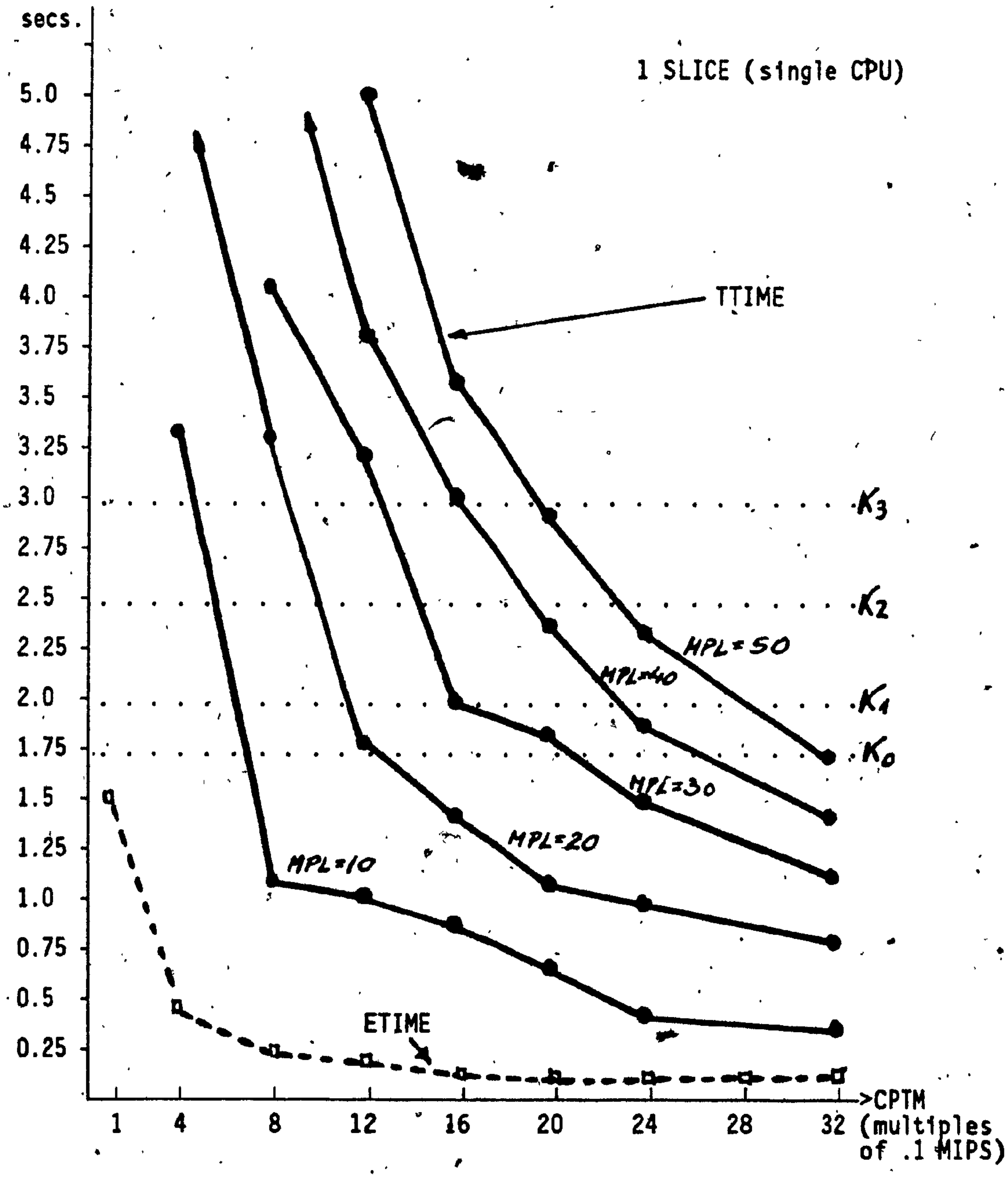
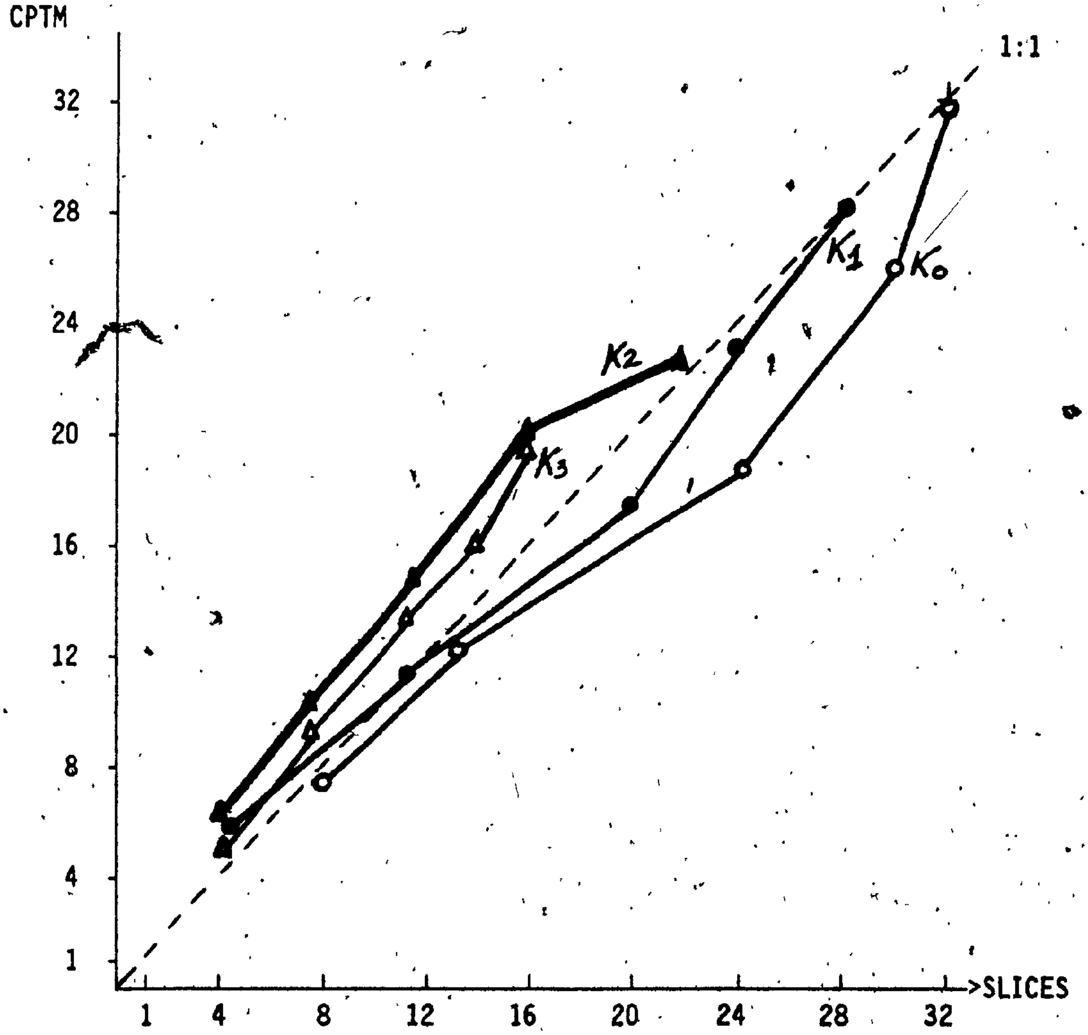


Fig. 22.- SINGLE CPU vs. PACOS.

TTIME = K (see Figs. 20,21)



10.0 CONCLUSION

The concept of Packet Execution attempts to replace a large, single-CPU multiprogramming system with a fully distributed architecture of smaller processors. While multi-processing is today a common approach used in distributing specific tasks, a fully distributed allocation of all tasks to all processors requires partitioning of programs in very specific ways. The Packet Group concept can be a cost-effective solution to the problem of generalizing the distribution of tasks among processors.

Simulation results indicate that the performance of PACOS does not degrade with the addition of subsequent slices. This is due to the multi-path structure of the packet architecture that avoids transmission bottlenecks by routing packets to the next free processor. As a result, the PACOS architecture can offer a very cost-effective solution to the expensive upgrading of heavily-loaded, single-CPU computer system.

The construction of a prototype of PACOS would be a most challenging project, that might open the doors to the replacement of massive and expensive single-CPU computer systems with modular, low-cost microprocessors.

11.0 APPENDICES

11.1 APPENDIX A.- Z-7 INSTRUCTION SET. CELSIUS PROGRAM

The Z7 user language is stack-oriented. There are siwteen different Z7 instructions, varying in length from one to three words. The instructions are:

OP CODE	MNEMONIC AND FORMAT
0	BRANCH <addr1> <addr2>
1	ENTER <addr>
2	RETURN
3	REPEAT
4	EXIT
5	PUSH <value>
6	FETCH
7	STORE
8	ADD
9	SUBSTRACT
10	MULTIPLY
11	DIVIDE
12	STOP
13	GET
14	PUT
15	ALLOCATE <value>

In the paragraphs that follow, the semantics of the various instructions are specified. Instructions with similar functions are grouped together. We use a simple notation to describe the instructions. The variables I and J are registeres local to the CPU. A left arrow (<-) denotes a stack operation. IP is the instruction pointer. STACK_PTR always addresses the word beyond the top of the stack; it is implicitly manipulated by most of the instructions.

```
ADD: J <- stack; I <- stack; stack <- I+J;
SUBSTRACT: J <- stack; I <- stack; stack <- I-J;
MULTIPLY: J <- stack; I <- stack; stack <- I*J;
DIVIDE:  J <- stack; IF J=0 THEN trap 12;
                          ELSE DO: I <- stack; stack <- I/J; END;
```

STOP, GET, and PUT each produce traps. In the case of a GET instruction, the CPU places the contents of the IO address on top of the stack. In the case of a PUT instruction, the entry on the top of the stack is placed at the specific address.

```
STOP: trap 3;
GET: stack <- memory(data segment, IO_ADDR);
PUT: memory(data segment, IO_ADDR) <- stack;
```

ALLOCATE increases the stack pointer by the amount specified, in order to reserve space for variables. The instruction pointer must be incremented to the word following <value>.

```
ALLOCATE <value>:
     STACK_PTR = STACK_PTR + <value>
     IP = IP + 1;  /* Bypass <value> */
```

PUSH, FETCH, and STORE each involve data transfers. PUSH places its operand on the top of the stack. FETCH removes the top entry from the stack, treats it as an address, and places the contents of that address on the top of the stack. STORE removes the entry on the the top of the stack and places it at the address specified by the second entry on the stack.

```
PUSH <value>:
    stack <- <value>;
    IP = IP + 1;  /* Bypass value */
FETCH: I <- stack; stack <- memory(data segment, I);
STORE: J <- stack; I <- stack; memory(data segment,I) = J;
```

ENTER, BRANCH, EXIT, RETURN, and REPEAT affect the flow of control of Z7 programs and are self-explanatory.

11.1.1 JOB CELSIUS

```
   M          EQU        0           VARIABLE M
   K          EQU        1           VARIABLE K
   I          EQU        2           VARIABLE I
   TEMPF      EQU        3           ARRAY TEMPF 20
   TEMPC      EQU        23          ARRAY TEMPC 20
   INPUT                             BLOCK INPUT
              PUSH       TEMPF       GET TEMPF(I)
              PUSH       I
              FETCH
              ADD
              GET
              STORE
              PUSH       I           I=I+1
              PUSH       I
              FETCH
              PUSH       1
              ADD
              STORE
              PUSH       M          REPEAT M-I
              FETCH
              PUSH       I
              FETCH
              SUBTRACT
              REPEAT
              EXIT                  END
CONVERT                             BLOCK CONVERT
              PUSH       K          K=TEMPF(I)
              PUSH       TEMPF
              PUSH       I
```

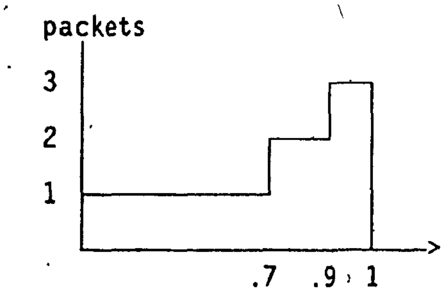
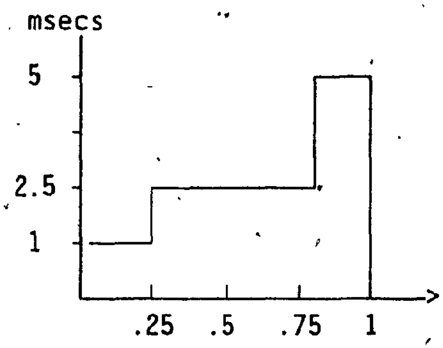
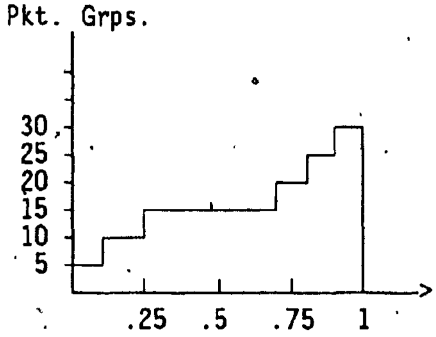
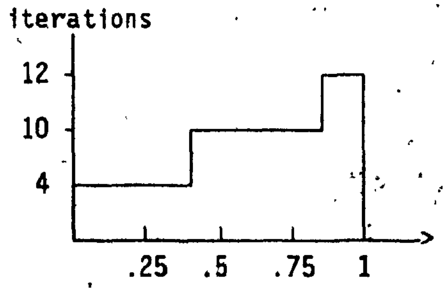
```
        FETCH
        ADD
        FETCH
        STORE
        PUSH        TEMPC           TEMPC(I)=K-32
        PUSH        I
        FETCH
        ADD
        PUSH        K
        FETCH
        PUSH        32
        SUBTRACT
        STORE
        PUSH        TEMPC           TEMPC(I)=TEMPC(I)*5
        PUSH        I
        FETCH
        ADD
        PUSH        TEMPC
        PUSH        I
        FETCH
        ADD
        FETCH
        PUSH        5
        MULTIPLY
        STORE
        PUSH        TEMPC           TEMPC(I)=TEMPC(I)/9
        PUSH        I
        FETCH
        ADD
        PUSH        TEMPC
        PUSH        I
        FETCH
        ADD
        FETCH
        PUSH        9
        DIVIDE
        STORE
        PUSH        TEMPC           PUT TEMPC(I)
        PUSH        I
        FETCH
        ADD
        FETCH
        PUT
        PUSH        I               I=I+1
        PUSH        I
        FETCH
        PUSH        1
        ADD
        STORE
        PUSH        M               REPEAT M-I
        FETCH
        PUSH        I
        FETCH
        SUBTRACT
        REPEAT
        EXIT                        END
MAIN    .                          BLOCK MAIN
        PUSH        M               GET M
        GET
```

```
STORE
PUSH        I             I=0
PUSH        0
STORE
ENTER       INPUT          ENTER INPUT
PUSH        I              I=0
PUSH        0
STORE
ENTER       CONVERT         ENTER CONVERT
EXIT                       END
ALLOCATE    43             BEGIN MAIN
ENTER       MAIN
STOP
```

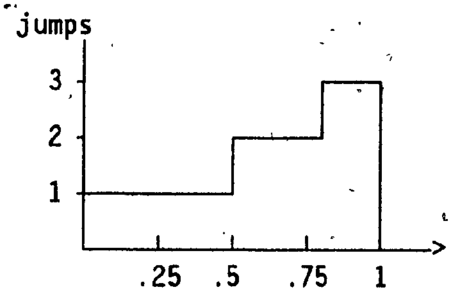
Data:

```
      7      23     45     52     3     75     100     65
```

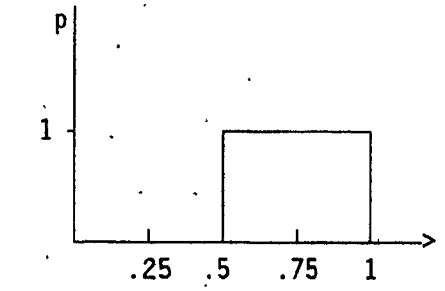
11.2 APPENDIX B.- GPSS SIMULATOR FUNCTIONS.

FUNCTION	DESCRIPTION	STRUCTURE
GEN	Gives the number of packets that will compose a new packet group (EP is not counted)	
PTIME	Execution time of one packet group (1 Kbytes)	
PLEN	Program length in number of packet groups executed.	
CBND	Number of iterations in a local loop. (Loop contained in the same packet)	

LPNO Number of jumps to subroutines or loops located in a different packet.



LPPR Probability of encountering a jump to subroutine or loop located in a different packet.



11.3 APPENDIX C.- GPSS PACOS SIMULATOR PROGRAM.- SAMPLE INPUT/OUTPUT.

PSLICE
NPL=20

```
CONCORDIA UNIVERSITY  GPSS V/6000                CRM GPSS V/6000 VER  2.0                          84/07/23  08 30 54
  BLOCK                                                                                                CARD
  NUMBER  *LOC  OPERATION  A,B,C,D,E,F,G,H,I,J       COMMENTS                                          NUMBER
                  SIMULATE                                                                                1
          **************************************************************                                  2
          **                                                                                              3
          **   PROGRAM PACOS SIMULATES THE BEHAVIOUR OF THE ABSTRACT                                      4
          **   MACHINE PACOS.                                                                             5
          **   THE FIRST GENERATE BLOCK GENERATES PROGRAMS, ONE PER                                       6
          **   GPSS TRANSACTION.                                                                          7
          **   EACH PROGRAM IS LATER ON SPLIT INTO A SET OF PACKET                                        8
          **   GROUPS. EACH PACKET GROUP CONTAINING TWO OR MORE                                           9
          **   PACKETS                                                                                   10
          **   THE SIMULATION PROGRAM MEASURES THE TURNAROUND TIME                                       11
          **   (OR RATHER EXECUTION TIME PLUS CPU WAIT TIME) OF EACH                                     12
          **   PROGRAM  THIS TIME IS TABULATED IN TTIME TABLE                                            13
          **   PROCESSORS AND LINKS ARE SIMULATED BY MEANS OF                                            14
          **   FACILITIES AND QUEUES  MAIN MEMORY IS SIMULATED BY THE                                    15
          **   USE OF STORAGES                                                                           16
          **   USER MANAGER PROCESSORS ARE FACILITIES 1001-1100                                          17
          **   DEVICE MANAGER PROCESSORS ARE FACILITIES 1101-1200                                        18
          **   MEMORY MANAGER PROCESSORS ARE FACILITIES 1201-1300                                        19
          **   EXECUTION MANAGER PROCESSORS ARE FACILITIES 1301-1400                                     20
          **   LOOPS (HORIZONTAL RINGS), ARE FACILITIES 1-400.                                           21
          **   SLICES (VERTICAL RINGS), ARE FACILITIES 401-900.                                          22
          **                                                                                             23
          **   EACH PACKET MOVES THROUGHOUT PACOS CARRYING ALL THE                                       24
          **   INFORMATION IN THE FOLLOWING PARAMET       ERS                                            25
          **                                                                                             26
          **      PARM. NO.               DESCRIPTION                                                    27
          **      ==========             ===============                                                 28
          **          1.-        PROCESSOR NO  WHERE THE PACKET RESIDES                                  29
          **          2.-        PROGRAM NUMBER                                                          30
          **          3.-        PACKET NUMBER                                                           31
          **          4.-        PACKET LENGTH (IN BYTES)                                                32
          **          5.-        DESTINATION 0= LEVEL                                                    33
          **          6.-        TYPE OF CONTROL COMMA          ND                                       34
          **                      1=GET,2=UPDATE,3=STORE,4=CLEAR,5=CREATE,6=SEND                         35
          **          7.-        PACKET TYPE 1=EP,2=I                                                    36
          **          8.-        ORIGINATING LEVEL (SAME AS PARM 5)                                      37
          **          9.-        HORIZONTAL LINK NUMBER.  IT IS CALCULATED BY                            38
          **                      VARIABLE LINK1.                                                        39
          **         10.-        VERTICAL LINK NUMBER  IT IS CALCULATED BY                               40
          **                     VARIABLE LINK2  IT IS EQUAL TO LINK1+400.                               41
          **         11.-        HOLDS THE NUMBER OF PACKETS THAT BELONG TO THE                          42
          **                     SAME PACKET GROUP                                                       43
          **         12.-        PROGRAM LENGTH. (NUMBER OF PACKET GROUPS TO                             44
          **                     BE EXECUTED)                                                            45
          **         13.-        HOLDS THE TURNAROUND TIME OF THE JOB.                                   46
          **         14.-        HOLDS THE RESPONSE TIME OF COMPUTER TRANSACTIONS.                       47
          **         15.-        HOLDS THE ACCUMULATED PROGRAM LENGTH DURING                             48
          **                     EXECUTION                                                               49
          **         16.-        HOLDS THE UMP NO. THAT IS PROGRAM CREATOR                               50
          **         17.-        HOLDS THE ACCUMULATED EXEC TIME PER PROGRAM                             51
          **                                                                                             52
          ************************************************************************                       53
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                    CNM GPSS V/6000 VER. 2 0                    84/07/23. 08.30.55.

           BLOCK                                                                                              CARD
           NUMBER  *LOC    OPERATION   A,B,C,D,E,F,G,H,I,J       COMMENTS                                    NUMBER

                           REALLOCATE FAC,1400,GRP,200,QUE,1400,STO,1400                                       54
                   PSIZE  EQU          1,X                                                                      1
                   SPEED  EQU          2,X                                                                      2
                   MSIZE  EQU          3,X                                                                      3
                   PACSZ  EQU          4,X                                                                      4
                   PNO    EQU          5,X                                                                      5
                   LEVX   EQU          6,X                                                                      6
                   HORX   EQU          7,X                                                                      7
                   VERX   EQU          8,X                                                                      8
                   PRGX   EQU          9,X                                                                      9
                   GENX   EQU          10,X                                                                    10
                   CPTM   EQU          11,X                                                                    11
             1     LINK1  VARIABLE     PF1-1000                                                                12
             2     LINK2  VARIABLE     V$LINK1+400                                                             13
             3     LEVL   VARIABLE     V$LINK1/100                                                             14
             4     SLICE  VARIABLE     V$LINK1//100                                                            15
             5     NORM   VARIABLE     X1/1000                                                                 16
             6     DEL    VARIABLE     V$NORM*100                                                              17
             7     CPUTM  FVARIABLE    FN$PTIME*FN$CBND*V$NORM*1000/X11                                        18
             8     SPAWN  VARIABLE     FN$LPPR*FN$CBND*FN$LPND*(4-V$NORM)                                      19
                                                                                                               20
                          INITIAL      X1,4000      ****  PACKET SIZE IN BYTES ****                            21
                          INITIAL      X2,4000        **** SPEED****                                           22
                          INITIAL      X3,50        ****  SIZE IN PACKETS OF EICH MEMORY MANAGER****           23
                          INITIAL      X4,4         ****  NUMBER OF SLICES ****                                24
                          INITIAL      X9,0         ****  INITIALY PROGRM NO. 0  ****                          25
                          INITIAL      X11,1     ****  PROCESSING SPEED OF EACH PROCESSOR =1 ***               26
                          INITIAL      X12,1001        ****   USER MANAGER NO OF PROGRAM ****                  27
                                                                                                               28
                                                                                                               29
             5     GEN    FUNCTION     RN1,D3    ***  NUMBER OF PACKETS THAT FORM A NEW PACKET GROUP***        30
             5     .7,1/.9,2/1,3/                                                                              31
                                                                                                               32
             1     PTIME  FUNCTION     RN1,D3    ***  EXECUTION TIME OF ONE PACKET GROUP ***                   33
             1     .2,1/.8,2.5/1,5/                                                                            34
                                                                                                               35
             6     PLEN   FUNCTION     RN1,D6      **   PROGRAM LENGTH IN NO OF PKT. GROUPS   **               36
             6     .1,5/.2,10/.7,15/.8,20/.9,25/1,30/                                                          37
                                                                                                               38
             2     CBND   FUNCTION     RN1,D3 *** NUMBER OF ITERATIONS IN A LOCAL LOOP***                      39
             2     .4,4/.8,10/1,12                                                                             40
                                                                                                               41
             4     LPND   FUNCTION     RN1,D3  *** NUMBER OF JUMPS TO A LOOP IN A DIFFERENT PACKET***          42
             4     .5,1/.8,2/1,3/                                                                              43
                                                                                                               44
             3     LPPR   FUNCTION     RN1,D2 *** PROB. OF FINDING A JUMP TO SUBR. IN DIFFERENT PACKET ***     45
             3     .5,0/1,1/                                                                                   46
                                                                                                               47
             7     NEXTP  FUNCTION     RN1,D2       **  PROBABILITY OF PACKET GROUP FAULT IN MM  **            48
             7     .7,0/1,1/                                                                                   49
                                                                                                               50
                                                                                                               51
                                                                                                               52
                 1         GENERATE    7000,,1,20,,26,F  *** GEN  PROS ONE EVERY 7 MSECS***                    53
                 2         MARK        13                     **  TURNAROUND TIME CLOCK STARTS  **             54
```

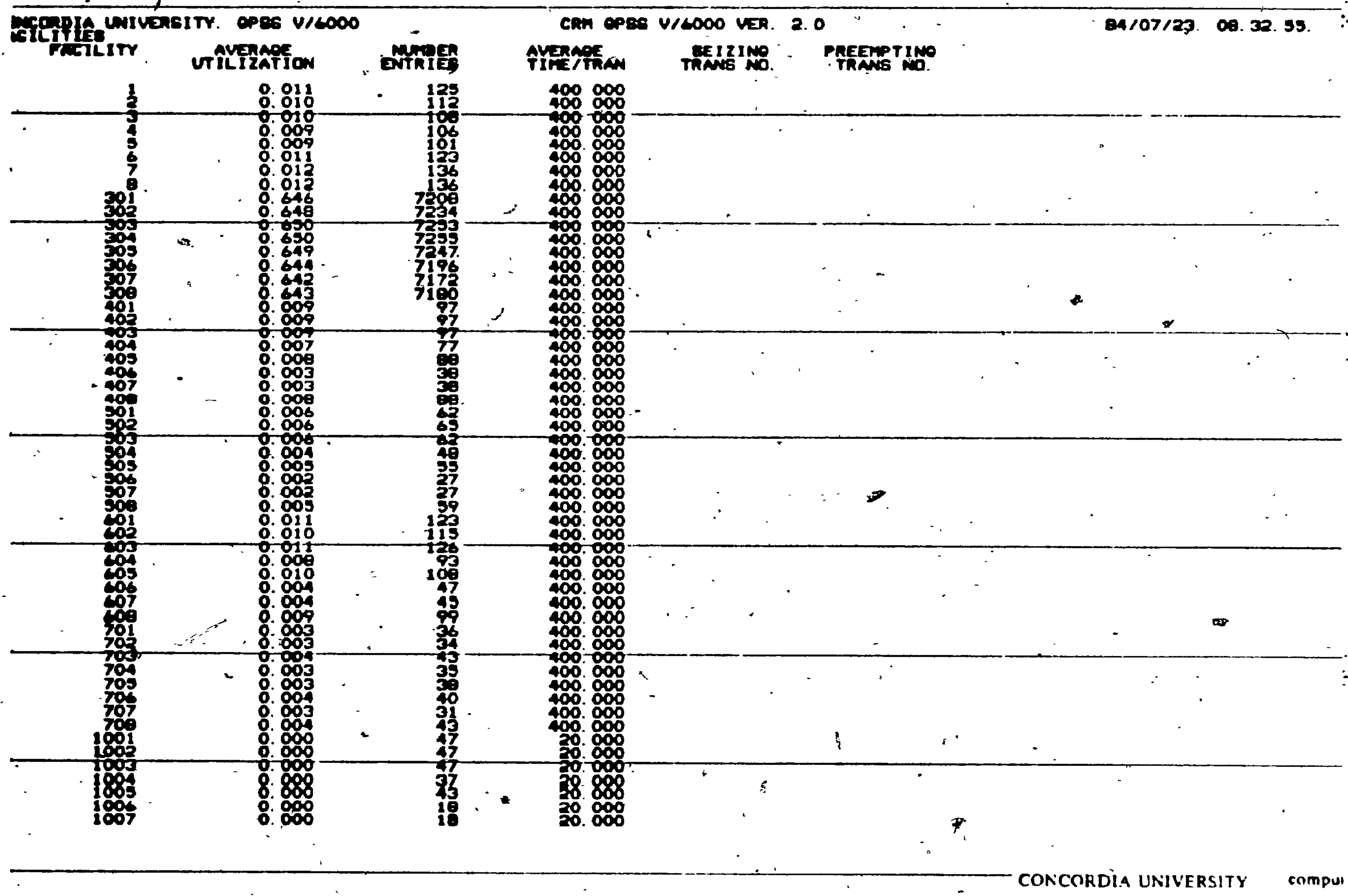
```
CONCORDIA UNIVERSITY. GPSS V/6000            CRM GPSS V/6000 VER. 2.0                      84/07/23. 08 30.55.
 BLOCK                                                                                             CARD
NUMBER  *LOC   OPERATION  A,B,C,D,E,F,G,H,I,J      COMMENTS                                        NUMBER
   3           ASSIGN     1,X12            ** ASSIGN  UMP PROCESSOR      **                          55
   4           ASSIGN     16,X12      **   ASSIGNS UMP PROGRAM CREATOR   **                          56
   5           TEST G     X4,V*SLICE,ULAST        ** CHECK IF LAST SLICE   **                        57
   6    UFRST  SAVEVALUE  12+,1          **  INCREASE UMP NO  BY ONE     **                          58
   7           TRANSFER   ,UNEXT                                                                     59
   8    ULAST  SAVEVALUE  12-,X4            **  BACK TO FIRST SLICE     ***                          60
   9           TRANSFER   ,UFRST                                                                     61
  10    UNEXT  SAVEVALUE  9+,1              **  NEW PROGRAM NUMBER      **                           62
  11           ASSIGN     2,X9                **  SET ASSEMBLY GROUP NUMBER  **                      63
  12           ASSIGN     5,2                **  DESTINATION LEVEL = MM  **                          64
  13           ASSIGN     6,1                **  CONTROL COMMAND = GET     **                        65
  14           ASSIGN     7,1            **  PACKET TYPE = EP    ***                                 66
  15           ASSIGN     9,V*LINK1            **     SET LOOP LINK NO    **                         67
  16           ASSIGN     10,V*LINK2           **     SET SLICE LINK NO    **                        68
  17           ASSIGN     12,K0              **   INITIALIZE  PROGRAM LENGTH   **                    69
  18           ASSIGN     15,FN*PLEN         **   ASSIGN BASIC PROGRAM LENGTH    **                  70
  19           ASSIGN     17,K0          **  INITIALIZE  EXEC TIME COUNTER  **                       71
                                                                                                     72
  20    CLEVL  TEST E     PF5,V*LEVL,OLEVL *** CHECK IF CURRENT LEVEL IS THE DESIRED ONE ***         73
  21    CRING  TEST NE    P5,0,USW   *** IS LEVEL 0  (USER MANAGER) ***                              74
  22           TEST NE    P5,1,DSW *** IS DESTINATION LEVEL 1 ***                                    75
  23           TEST NE    P5,2,MSW                                                                   76
  24           TEST NE    P5,3,ESW                                                                   77
                                                                                                     78
  25    OLEVL  QUEUE      PF10   *** DEST IS ANOTHER LEVEL. QUEUE FOR LINK ***                       79
  26           SEIZE      PF10             **   GRAB SLICE LINK TO GO TO OTHER LEVEL   **            80
  27           TRANSFER   ,ALEVL                                                                     81
  28    DSW    QUEUE      PF9              **  DEST IS ANOTHER SWITCH AT SAME LEVEL  **              82
  29           SEIZE      PF9            **  GRAB LOOP LINK TO GO TO OTHER SWITCH  **                83
  30           TRANSFER   ,ARING                                                                     84
  31    ARING  ADVANCE    V*DEL              ***    INTRA-NODAL DELAY    **                          85
  32           RELEASE    PF9            ***    LOOP  LINK IS RELEASED     ***                       86
  33           DEPART     PF9                                                                        87
  34           TEST G     X4,V*SLIC[illegible]NG       *** IS THIS THE LAST SLICE  **                88
  35    BRING  ASSIGN     1+,1                 **  INCREASE PROCESSOR NO.    **                      89
  36           SAVEVALUE  PNO,PF1.                                                                   90
  37           ASSIGN     9,V*LINK1              NEW LOOP LINK NO  **                                91
  38           ASSIGN     10,V*LIN[illegible]      **  NEW SLICE LINK NO.   **                      92
  39           TRANSFER   ,CRING                                                                     93
  40    WRING  ASSIGN     1-,X4                ***  BACK TO FIRST SLICE    **                        94
  41           SAVEVALUE  PNO,PF1                                                                    95
  42           TRANSFER   ,BRING                                                                     96
                                                                                                     97
  43    ALEVL  ADVANCE    V*DEL                   **    INTER-NODAL DELAY  **                        98
  44           SAVEVALUE  6,V*LEVL                                                                   99
  45           RELEASE    PF10              **  RELEASE SLICE LINK  **                              100
  46           DEPART     PF10                                                                      101
  47           TEST L     X6,3,WLEVL          **  IS LEVEL   EML   **                               102
  48           ASSIGN     1+,100            **  NEW  LEVEL           **                             103
  49           SAVEVALUE  PNO,PF1                                                                   104
                                                                                                    105
  50    BLEVL  ASSIGN     9,V*LINK1               **  ASSIGN LOOP LINK NO.   **                     106
  51           ASSIGN     10,V*LINK2              **  ASSIGN NEW SLICE LINK NO  **                  107
  52           TRANSFER   ,CLEVL                                                                    108
                                                                                                    109
```

```
ONCORDIA UNIVERSITY.  GPSS V/6000                   CRM GPSS V/6000 VER  2.0                         84/07/23.  08.30.55.
  BLOCK                                                                                                     CARD
  NUMBER  *LOC    OPERATION  A,B,C,D,E,F,G,H,I,J        COMMENTS                                          NUMBER
    53    ULEVL   ASSIGN     1-,300               **  BACK TO LEVEL 0 (UML)   **                          110
    54            SAVEVALUE  PNO,PF1                                                                      111
    55            TRANSFER   ,DLEVL                                                                       112
                                                                                                          113
                  ****************************************************                                   114
                  **        USER MANAGER LEVEL                                                            115
                  ****************************************************                                   116
    56    USW     TEST E     PF5,0,GLEVL   ***  CHECK IF DEST= USER MANAGER ***                           117
    57            TEST E     PF6,2,UEND   *** CHECK IF COMMAND= UPDATE ***                                118
    58            TEST E     PF1,PF16,DSW  *** CHECK IF UMP IS PROGRAM CREATOR ***                        119
    59    UMP     QUEUE      PF1             **  QUEUE FOR  UMP  **                                       120
    60            SEIZE      PF1                                                                          121
    61            ADVANCE    K20     ***  UMP UPDATES EP PACKET ****                                      122
    62            RELEASE    PF1                                                                          123
    63            DEPART     PF1                                                                          124
    64            SPLIT      1,UGRP                                                                       125
    65            ASSIGN     5,2          **** DESTINATION= MML ****                                      126
    66            ASSIGN     6,1          **** GET EP COMMAND ***                                         127
    67            TRANSFER   ,CLEVL                                                                       128
    68    UGRP    ASSIGN     5,1   ****  DESTINATION= DML ***                                             129
    69            ASSIGN     6,1          ****  GET NEXT PKT.GRP.SET  COMMAND ****                        130
    70            TRANSFER   ,CLEVL                                                                       131
    71    UEND    TERMINATE                                                                               132
                                                                                                          133
                                                                                                          134
                  ****************************************************                                   135
                  **     DEVICE MANAGER LEVEL                                                             136
                  ****************************************************                                   137
    72    DSW     TEST NE    PF6,1,DMP      ***  CHECK FOR GET COMMAND ****                               138
    73            TERMINATE                                                                               139
    74    DMP     TEST NE    FN$NEXTP,1,DMP1     **  IF PACKET GROUP FAULT, GET IT FROM DMP  **           140
    75            TERMINATE                                                                               141
    76    DMP1    QUEUE      PF1                                                                          142
    77            SEIZE      PF1                                                                          143
    78            ADVANCE    K20000       *  RETRIEVING PKT.GRP.SET FROM DISK ***                         144
    79            RELEASE    PF1                                                                          145
    80            DEPART     PF1                                                                          146
    81            ASSIGN     5,2        ***  DEST= MEMORY MANAGER ***                                     147
    82            ASSIGN     6,3        ***   COMMAND = STORE     ***                                     148
    83            TRANSFER   ,CLEVL                                                                       149
                                                                                                          150
                                                                                                          151
                                                                                                          152
                  ****************************************************                                   153
                  **     MEMORY MANAGER LEVEL                                                             154
                  ****************************************************                                   155
    84    MSW     TEST NE    PF6,1,GET            **  CHECK FOR GET COMMAND  **                           156
    85            TEST NE    PF6,3,STORE      ***     CHECK IF STORE COMMAND ***                          157
    86    GET     QUEUE      PF1                                                                          158
    87            SEIZE      PF1                                                                          159
    88            ADVANCE    K200         ** MEMORY ACCESS AND UPDATE DELAYS  **                          160
    89            RELEASE    PF1                                                                          161
    90            DEPART     PF1                                                                          162
    91            SAVEVALUE  10,FN$GEN        **  NUMBER OF PACKETS IN GROUP  ***                         163
    92            ASSIGN     11,X10                                                                       164
```

CONCORDIA UNIVERSITY   compu

```
CONCORDIA UNIVERSITY. GPSS V/6000                  CRM GPSS V/6000 VER. 2.0                                       84/07/23. 08 30 55.

      BLOCK                                                                                                                CARD
      NUMBER  *LOC   OPERATION  A,B,C,D,E,F,G,H,I,J       COMMENTS                                                      NUMBER
        93           ASSIGN     11+,1            **  INCREASE NUMBER OF PACKETS BY ONE (EP) **                             165
        94           SPLIT      X10,NEXT         **  GENERATE PACKET GROUP  **                                             166
        95           JOIN       PF2              **  BELONG TO SAME ASSEMBLY SET   ***                                     167
        96           ASSIGN     5,3              **  DEST=  EM LEVEL  ***                                                  168
        97           TRANSFER   ,CLEVL                                                                                     169
        98    NEXT   JOIN       PF2                                                                                        170
        99           ASSIGN     5,3                                                                                        171
       100           ASSIGN     7,2              **  THESE,PACKETS ARE IP,DP, PACKETS  **                                  172
       101           TRANSFER   ,CLEVL                                                                                     173
                                                                                                                           174
       102    STORE  ENTER      PF1                   **  ACCESS MAIN MEMORY    **                                         175
       103           ADVANCE    K20        ** STORAGE AND UPDATE OF TABLES **                                              176
       104           LEAVE      PF1                                                                                        177
       105           TERMINATE                                                                                             178
                                                                                                                           179
                                                                                                                           180
                                                                                                                           181
              *********************************************************                                                    182
              **        EXECUTION  MANAGER  LEVEL                                                                          183
              *********************************************************                                                    184
                                                                                                                           185
       106    EBW    TEST E     PF5,3,OLEVL           **  CHECK FOR EM LEVEL  ***                                          186
       107           TEST E     PF7,1,OTHER              **  CHECK FOR  EP PACKET  ***                                     187
       108           GATE NU    PF1,OBW                **  IF EMP IS IN USE, GO TO GET ANOTHER  **                         188
       109           QUEUE      PF1                  **  ELSE, GET THIS EMP    ***                                         189
       110           SEIZE      PF1                                                                                        190
       111           LOGIC S    PF2                 **  SET  CONTROL SWITCH    **                                          191
       112           TRANSFER   ,GRP                                                                                       192
       113    OTHER  GATE LS    PF2,DLY            ** IF SWITCH NOT SET, WAIT A LITTLE  **                                 193
       114           TRANSFER   ,GRP                                                                                       194
                                                                                                                           195
       115    DLY    ADVANCE    V$DEL              **  WAIT BEFORE LOOKING FOR THE EP  **                                  196
       116           TRANSFER   ,DBW                                                                                       197
                                                                                                                           198
       117    GRP    ASSEMBLE   PF11              **  GATHER ALL THE MEMBERS OF THE PKT.GRP  **                            199
       118           REMOVE     PF2,ALL          **  CANCEL GROUP NUMBER  **                                               200
       119           LOGIC R    PF2              **  RESET  SWITCH  **                                                     201
       120           MARK       14             **  COUNTS BEGINNING OF EXECUTION  ***                                      202
       121           ADVANCE    V$CPUTM          **  PKT.GRP EXECUTES IN EMP   **                                          203
       122           ASSIGN     15+,V$SPAWN        **  INCREASE PROGRAM LENGTH   **                                        204
       123           RELEASE    PF1                                                                                        205
       124           DEPART     PF1                                                                                        206
       125           ASSIGN     17+,MP14      ** ACCUMULATES EXEC TIME PER PROGRAM **                                      207
       126           ASSIGN     12+,1            **  COUNTING PACKET GROUPS     **                                         208
       127           TEST L     PF12,PF15,OUT        **    IF LIMIT REACHED, PROGRAM ENDS- **                              209
       128           TEST L     PF12,50,OUT          ** NO MORE THAN 50 PACKET GROUPS  **                                  210
       129           ASSIGN     5,0           **  DESTINATION=USER MANAGER LEVEL **                                        211
       130           ASSIGN     7,1                    **  PACKET TYPE = EP   **                                           212
       131           ASSIGN     6,2            **  COMMAND= UPDATE **                                                      213
       132           TRANSFER   ,OLEVL                                                                                     214
                                                                                                                           215
       133    OUT    TABULATE   TTIME                **  TURNAROUND TIME PER PROGRAM  **                                   216
       134           TABULATE   ETIME              **  EXEC TIME PER PROGRAM  **                                           217
       135           TABULATE   NPAC                 **   NUMBER OF PACKET GROUPS PER PROGRAM  **                          218
   1          TTIME  TABLE      MP13,0,100,20                                                                              219
```

```
CONCORDIA UNIVERSITY.  GPSS V/6000                 CRM GPSS V/6000 VER. 2.0                            84/07/23.  08 30.55
                BLOCK                                                                                             CARD
                NUMBER  *LOC   OPERATION  A,B,C,D,E,F,G,H,I,J      COMMENTS                                      NUMBER
           2            ETIME  TABLE      PF17,0,100,20                                                              220
           3            NPAC   TABLE      PF12,0,5,8                                                                 221
                 138           TERMINATE  1                                                                          222
                                                                                                                     223
                                                                                                                     224
                               RESET                                                                                 225
                               INITIAL    X4,1               **    ONE  SLICE   ***                                  226
                               REPORT     PSLICE                                                                     227
                        FAC    TITLE      ,FACILITIES                                                                228
                        TAB    TITLE      ,TABLES                                                                    229
                        TAB    INCLUDE    T1-T471,2,3,4                                                              230
                               ENDREPORT                                                                             231
                                                                                                                     232
                      *        START      20,...,PSLICE                                                              233
                                                                                                                     234
                               CLEAR      X1-X3,X11-X12                                                              235
                               INITIAL    X4,4                 **  FOUR SLICES    **                                 236
                      *        START      20,...,PSLICE                                                              237
                                                                                                                     238
                               CLEAR      X1-X3,X11-X12                                                              239
                               INITIAL    X4,8               **   EIGHT SLICES   ***                                 240
                               START      20,...,PSLICE                                                              241
```



```
NCORDIA UNIVERSITY. GPSS V/6000                    CRM GPSS V/6000 VER. 2.0                    84/07/23. 08.32.55.
CILITIES
  FACILITY       AVERAGE          NUMBER      AVERAGE        SEIZING      PREEMPTING
               UTILIZATION        ENTRIES    TIME/TRAN      TRANS NO.     TRANS NO.

       1          0.011             125       400.000
       2          0.010             112       400.000
       3          0.010             108       400.000
       4          0.009             106       400.000
       5          0.009             101       400.000
       6          0.011             123       400.000
       7          0.012             136       400.000
       8          0.012             136       400.000
     301          0.646            7208       400.000
     302          0.648            7234       400.000
     303          0.650            7253       400.000
     304          0.650            7255       400.000
     305          0.649            7247       400.000
     306          0.644            7196       400.000
     307          0.642            7172       400.000
     308          0.643            7180       400.000
     401          0.009              97       400.000
     402          0.009              97       400.000
     403          0.009              97       400.000
     404          0.007              77       400.000
     405          0.008              88       400.000
     406          0.003              38       400.000
     407          0.003              38       400.000
     408          0.008              88       400.000
     501          0.006              62       400.000
     502          0.006              65       400.000
     503          0.006              62       400.000
     504          0.004              48       400.000
     505          0.005              55       400.000
     506          0.002              27       400.000
     507          0.002              27       400.000
     508          0.005              59       400.000
     601          0.011             123       400.000
     602          0.010             115       400.000
     603          0.011             126       400.000
     604          0.008              93       400.000
     605          0.010             108       400.000
     606          0.004              47       400.000
     607          0.004              45       400.000
     608          0.009              99       400.000
     701          0.003              36       400.000
     702          0.003              34       400.000
     703          0.004              43       400.000
     704          0.003              35       400.000
     705          0.003              38       400.000
     706          0.004              40       400.000
     707          0.003              31       400.000
     708          0.004              43       400.000
    1001          0.000              47        20.000
    1002          0.000              47        20.000
    1003          0.000              47        20.000
    1004          0.000              37        20.000
    1005          0.000              43        20.000
    1006          0.000              18        20.000
    1007          0.000              18        20.000
```

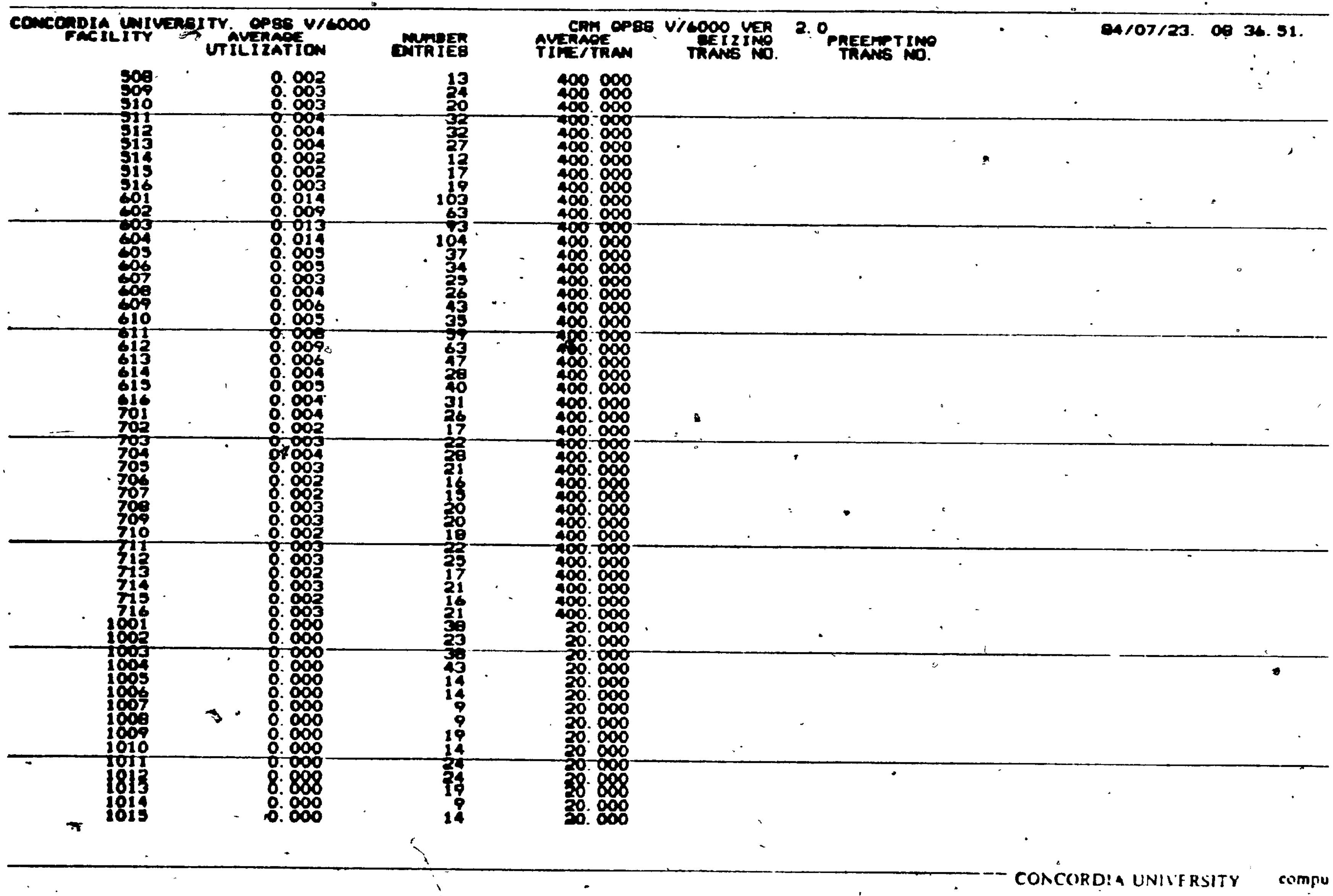
CONCORDIA UNIVERSITY compu

```
CONCORDIA UNIVERSITY. GPSS V/6000                    CRM GPSS V/6000 VER. 2.0                                84/07/23. 08.32.55.
    FACILITY        AVERAGE          NUMBER        AVERAGE       SEIZING     PREEMPTING
                 UTILIZATION        ENTRIES       TIME/TRAN      TRANS NO.    TRANS NO.

       1008          0.000               43          20.000
       1101          0.054               12       20000.000
       1102          0.067               15       20000.000
       1103          0.054               12       20000.000
       1104          0.036                8       20000.000
       1105          0.045               10       20000.000
       1106          0.031                7       20000.000
       1107          0.031                7       20000.000
       1108          0.063               14       20000.000
       1201          0.002               50         200.000
       1202          0.002               50         200.000
       1203          0.002               50         200.000
       1204          0.002               40         200.000
       1205          0.002               45         200.000
       1206          0.001               20         200.000
       1207          0.001               20         200.000
       1208          0.002               45         200.000
       1301          0.763               39       87411.795
       1302          0.825               37       99648.108
       1303          0.925               45       91786.667
       1304          0.734               36       91068.889
       1305          0.804               43       83525.581
       1306          0.786               44       79795.000
       1307          0.731               31      105329.032
       1308          0.977               45       96933.333
TABLES
    TABLE TTIME
    ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                  20            3011854.000                  695222.905

    TABLE ETIME
    ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                  20            1449600.000                  531757.602

    TABLE NPAC
    ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                  20                 16.000                       5.982

                      CLEAR       X1-X3,X11-X12                                                                242
                      INITIAL     X4,12                  **  TWELVE  SLICES  ***                               243
                      START       20,...,PSLICE                                                                244
                                                                                                               245
```

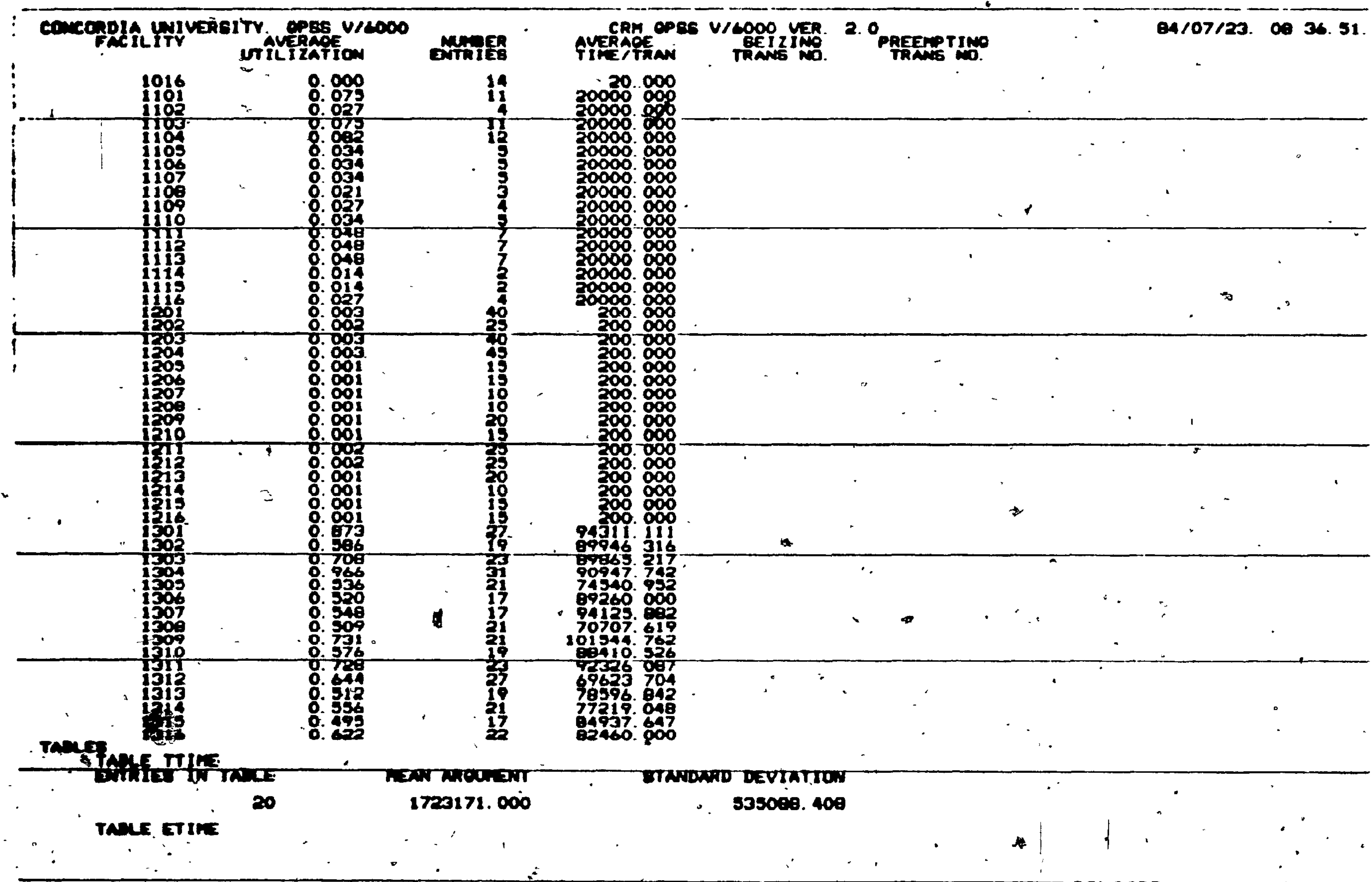
```
CONCORDIA UNIVERSITY.  GPSS V/6000              CRM GPSS V/6000 VER  2.0                          84/07/23.  08 35 50.
FACILITIES
    FACILITY          AVERAGE          NUMBER       AVERAGE       SEIZING     PREEMPTING
                    UTILIZATION        ENTRIES     TIME/TRAN      TRANS NO.    TRANS NO.

          1            0.011             93        400.000
          2            0.012             99        400.000
          3            0.012            103        400.000
          4            0.013            111        400 000
          5            0.014            123        400 000
          6            0.014            121        400 000
          7            0.013            113        400 000
          8            0.013            112        400.000
          9            0.012            105        400.000
         10            0.012            104        400 000
         11            0.012             99        400 000
         12            0.010             85        400 000
        301            0.274           2339        400.000
        302            0.272           2325        400 000
        303            0.272           2317        400.000
        304            0.269           2296        400.000
        305            0.267           2275        400 000
        306            0.267           2274        400 000
        307            0.270           2301        400 000
        308            0.270           2307        400.000
        309            0.271           2316        400.000
        310            0.273           2330        400.000
        311            0.274           2336        400.000
        312            0.275           2347        400.000
        401            0.003             29        400.000
        402            0.007             59        400.000
        403            0.006             49        400.000
        404            0.003             29        400.000
        405            0.003             28        400.000
        406            0.007             58        400.000
        407            0.007             58        400 000
        408            0.004             38        400.000
        409            0.009             78        400.000
        410            0.006             48        400.000
        411            0.007             58        400.000
        412            0.008             68        400.000
        501            0.002             18        400.000
        502            0.004             37        400.000
        503            0.004             31        400.000
        504            0.002             21        400.000
        505            0.002             20        400.000
        506            0.005             42        400.000
        507            0.005             40        400.000
        508            0.003             23        400.000
        509            0.006             49        400.000
        510            0 004             31        400.000
        511            0.005             40        400.000
        512            0.006             51        400.000
        601            0.004             38        400.000
        602            0.008             66        400.000
        603            0.007             59        400.000
        604            0.004             33        400.000
        605            0.004             36        400.000
        606            0.008             64        400.000
        607            0.008             72        400.000
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                           CRM GPSS V/6000 VER. 2.0                                    84/07/23. 08.35.51.
   FACILITY        AVERAGE           NUMBER         AVERAGE        SEIZING         PREEMPTING
                 UTILIZATION         ENTRIES       TIME/TRAN       TRANS NO.        TRANS NO.

       608          0.006               52          400.000
       609          0.011               90          400.000
       610          0.007               58          400.000
       611          0.008               67          400.000
       612          0.010               82          400.000
       701          0.003               22          400.000
       702          0.004               35          400.000
       703          0.003               28          400.000
       704          0.003               22          400.000
       705          0.003               25          400.000
       706          0.003               26          400.000
       707          0.002               20          400.000
       708          0.002               17          400.000
       709          0.004               31          400.000
       710          0.003               22          400.000
       711          0.003               23          400.000
       712          0.002               19          400.000
      1001          0.000               14           20.000
      1002          0.000               29           20.000
      1003          0.000               24           20.000
      1004          0.000               14           20.000
      1005          0.000               13           20.000
      1006          0.000               28           20.000
      1007          0.000               28           20.000
      1008          0.000               18           20.000
      1009          0.000               38           20.000
      1010          0.000               23           20.000
      1011          0.000               28           20.000
      1012          0.000               33           20.000
      1101          0.018                3        20000.000
      1102          0.041                7        20000.000
      1103          0.035                6        20000.000
      1104          0.035                6        20000.000
      1105          0.029                5        20000.000
      1106          0.070               12        20000.000
      1107          0.059               10        20000.000
      1108          0.018                3        20000.000
      1109          0.053                9        20000.000
      1110          0.035                6        20000.000
      1111          0.059               10        20000.000
      1112          0.094               16        20000.000
      1201          0.001               15          200.000
      1202          0.002               30          200.000
      1203          0.001               25          200.000
      1204          0.001               15          200.000
      1205          0.001               15          200.000
      1206          0.002               30          200.000
      1207          0.002               30          200.000
      1208          0.001               20          200.000
      1209          0.002               40          200.000
      1210          0.001               25          200.000
      1211          0.002               30          200.000
      1212          0.002               35          200.000
      1301          0.563               22        87360.000
      1302          0.960               36        91012.222
      1303          0.765               30        87010.000

                                                                                                     CONCORDIA UNIVERSITY    con
CONCORDIA UNIVERSITY, GPSS V/6000               CRM GPSS V/6000 VER. 2.0                          84/07/23. 08 35.51.
   FACILITY       AVERAGE         NUMBER       AVERAGE       SEIZING    PREEMPTING
               UTILIZATION        ENTRIES     TIME/TRAN     TRANS NO.    TRANS NO.
       1304          0.545            24      77558.333
       1305          0.534            26      70057.692
       1306          0.594            27      75066.667
       1307          0.555            20      94718.000
       1308          0.550            20      93887.000
       1309          0.732            34      73459.412
       1310          0.540            23      80072.174
       1311          0.626            27      79101.481
       1312          0.656            21     106587.619
TABLES
   TABLE TTIME
   ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                 20             1821785.000                  647127.039
   TABLE ETIME
   ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION
                 20             1293400.000                  595886.637
   TABLE NPAC
   ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION
                 20                  15.500                       6.048

                                                                                                           246
                                                                                                           247
                                                                                                           248
                      CLEAR        X1-X3,X11-X12                                                           249
                      INITIAL      X4,16              ** SIXTEEN SLICES **                                 250
                      START        20....PSLICE                                                            251
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                CRM GPSS V/6000 VER. 2.0                          84/07/23. 08 36.51.
FACILITIES
    FACILITY         AVERAGE           NUMBER        AVERAGE        SEIZING      PREEMPTING
                   UTILIZATION        ENTRIES       TIME/TRAN      TRANS NO.     TRANS NO

          1            0.018              129         400.000
          2            0.017              123         400.000
          3            0.015              107         400.000
          4            0.013               92         400.000
          5            0.014               99         400.000
          6            0.014              101         400.000
          7            0.015              107         400.000
          8            0.016              118         400.000
          9            0.016              119         400.000
         10            0.017              123         400.000
         11            0.017              121         400.000
         12            0.017              122         400.000
         13            0.016              120         400.000
         14            0.018              132         400.000
         15            0.018              134         400.000
         16            0.019              141         400.000
        301            0.095              690         400.000
        302            0.098              714         400.000
        303            0.103              755         400.000
        304            0.106              770         400.000
        305            0.104              760         400.000
        306            0.104              756         400.000
        307            0.101              739         400.000
        308            0.098              714         400.000
        309            0.097              711         400.000
        310            0.097              711         400.000
        311            0.097              705         400.000
        312            0.096              698         400.000
        313            0.095              694         400.000
        314            0.092              673         400.000
        315            0.092              672         400.000
        316            0.091              664         400.000
        401            0.011               78         400.000
        402            0.007               48         400.000
        403            0.011               78         400.000
        404            0.012               88         400.000
        405            0.004               29         400.000
        406            0.004               29         400.000
        407            0.003               19         400.000
        408            0.003               19         400.000
        409            0.005               39         400.000
        410            0.004               29         400.000
        411            0.007               49         400.000
        412            0.007               49         400.000
        413            0.005               39         400.000
        414            0.003               19         400.000
        415            0.004               29         400.000
        416            0.004               29         400.000
        501            0.007               51         400.000
        502            0.004               29         400.000
        503            0.007               51         400.000
        504            0.008               57         400.000
        505            0.003               20         400.000
        506            0.003               20         400.000
        507            0.002               15         400.000
```



```
CONCORDIA UNIVERSITY, GPSS V/6000                  CRM GPSS V/6000 VER  2.0                                   84/07/23.  08 34.51.
     FACILITY          AVERAGE         NUMBER       AVERAGE      SEIZING        PREEMPTING
                     UTILIZATION       ENTRIES     TIME/TRAN     TRANS NO.       TRANS NO.

          508           0.002             13        400.000
          509           0.003             24        400.000
          510           0.003             20        400.000
          511           0.004             32        400.000
          512           0.004             32        400.000
          513           0.004             27        400.000
          514           0.002             12        400.000
          515           0.002             17        400.000
          516           0.003             19        400.000
          601           0.014            103        400.000
          602           0.009             63        400.000
          603           0.013             93        400.000
          604           0.014            104        400.000
          605           0.005             37        400.000
          606           0.005             34        400.000
          607           0.003             25        400.000
          608           0.004             26        400.000
          609           0.006             43        400.000
          610           0.005             35        400.000
          611           0.008             59        400.000
          612           0.009             63        400.000
          613           0.006             47        400.000
          614           0.004             28        400.000
          615           0.005             40        400.000
          616           0.004             31        400.000
          701           0.004             26        400.000
          702           0.002             17        400.000
          703           0.003             22        400.000
          704           0.004             28        400.000
          705           0.003             21        400.000
          706           0.002             16        400.000
          707           0.002             15        400.000
          708           0.003             20        400.000
          709           0.003             20        400.000
          710           0.002             18        400.000
          711           0.003             22        400.000
          712           0.003             25        400.000
          713           0.002             17        400.000
          714           0.003             21        400.000
          715           0.002             16        400.000
          716           0.003             21        400.000
         1001           0.000             38         20.000
         1002           0.000             23         20.000
         1003           0.000             38         20.000
         1004           0.000             43         20.000
         1005           0.000             14         20.000
         1006           0.000             14         20.000
         1007           0.000              9         20.000
         1008           0.000              9         20.000
         1009           0.000             19         20.000
         1010           0.000             14         20.000
         1011           0.000             24         20.000
         1012           0.000             24         20.000
         1013           0.000             19         20.000
         1014           0.000              9         20.000
         1015           0.000             14         20.000

                                                                                     CONCORDIA UNIVERSITY    compu
```



```
CONCORDIA UNIVERSITY. GPSS V/6000                     CRM GPSS V/6000 VER. 2.0                    84/07/23. 08 36.51.
     FACILITY          AVERAGE         NUMBER         AVERAGE      SEIZING      PREEMPTING
                     UTILIZATION       ENTRIES       TIME/TRAN     TRANS NO.     TRANS NO.

         1016           0.000            14             20.000
         1101           0.075            11          20000.000
         1102           0.027             4          20000.000
         1103           0.075            11          20000.000
         1104           0.082            12          20000.000
         1105           0.034             5          20000.000
         1106           0.034             5          20000.000
         1107           0.034             5          20000.000
         1108           0.021             3          20000.000
         1109           0.027             4          20000.000
         1110           0.034             5          20000.000
         1111           0.048             7          20000.000
         1112           0.048             7          20000.000
         1113           0.048             7          20000.000
         1114           0.014             2          20000.000
         1115           0.014             2          20000.000
         1116           0.027             4          20000.000
         1201           0.003            40            200.000
         1202           0.002            25            200.000
         1203           0.003            40            200.000
         1204           0.003            45            200.000
         1205           0.001            15            200.000
         1206           0.001            15            200.000
         1207           0.001            10            200.000
         1208           0.001            10            200.000
         1209           0.001            20            200.000
         1210           0.001            15            200.000
         1211           0.002            25            200.000
         1212           0.002            25            200.000
         1213           0.001            20            200.000
         1214           0.001            10            200.000
         1215           0.001            15            200.000
         1216           0.001            15            200.000
         1301           0.873            27          94311.111
         1302           0.586            19          89946.316
         1303           0.708            23          89865.217
         1304           0.966            31          90947.742
         1305           0.536            21          74540.952
         1306           0.520            17          89260.000
         1307           0.548            17          94125.882
         1308           0.509            21          70707.619
         1309           0.731            21         101544.762
         1310           0.576            19          88410.526
         1311           0.728            23          92326.087
         1312           0.644            27          69623.704
         1313           0.512            19          78596.842
         1214           0.556            21          77219.048
  [illegible]           0.495            17          84937.647
  [illegible]           0.622            22          82460.000
TABLES
  TABLE TTIME
     ENTRIES IN TABLE             MEAN ARGUMENT              STANDARD DEVIATION

                   20               1723171.000                     535088.408

     TABLE ETIME
```

```
CONCORDIA UNIVERSITY  GPSS V/6000          CRM GPSS V/6000 VER. 2.0                       84/07/23.  08.36.51.
         ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION
                       20         1447200.000                  529725.410
         TABLE NPAC
         ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION
                       20              17.250                       5.955

                                                                                                   252
                      CLEAR       X1-X3,X11-X12                                                    253
                      INITIAL     X4,20            **  TWENTY        SLICES    **                  254
                      START       20,,,PSLICE                                                      255
                                                                                                   256
```

```
CONCORDIA UNIVERSITY.  GPSS V/6000                CRM GPSS V/6000 VER.  2.0                              84/07/23.  08.37.01.
FACILITIES
    FACILITY        AVERAGE         NUMBER       AVERAGE       SEIZING     PREEMPTING
                  UTILIZATION       ENTRIES     TIME/TRAN     TRANS NO.     TRANS NO.
         401          0.004            29        400.000
         402          0.004            29        400.000
         403          0.007            59        400.000
         404          0.006            49        400.000
         405          0.006            49        400.000
         406          0.004            29        400.000
         407          0.005            39        400.000
         408          0.007            59        400.000
         409          0.004            29        400.000
         410          0.006            49        400.000
         411          0.004            29        400.000
         412          0.006            49        400.000
         413          0.004            29        400.000
         414          0.002            19        400.000
         415          0.004            29        400.000
         416          0.004            29        400.000
         417          0.001             9        400.000
         418          0.005            39        400.000
         419          0.007            59        400.000
         420          0.004            29        400.000
         501          0.002            19        400.000
         502          0.002            18        400.000
         503          0.004            34        400.000
         504          0.004            32        400.000
         505          0.003            27        400.000
         506          0.003            25        400.000
         507          0.003            23        400.000
         508          0.005            42        400.000
         509          0.002            19        400.000
         510          0.004            33        400.000
         511          0.002            18        400.000
         512          0.004            31        400.000
         513          0.002            19        400.000
         514          0.001            11        400.000
         515          0.002            20        400.000
         516          0.002            17        400.000
         517          0.001             6        400.000
         518          0.003            25        400.000
         519          0.004            37        400.000
         520          0.003            22        400.000
         601          0.004            34        400.000
         602          0.004            36        400.000
         603          0.008            66        400.000
         604          0.007            56        400.000
         605          0.007            56        400.000
         606          0.005            41        400.000
         607          0.006            50        400.000
         608          0.009            77        400.000
         609          0.005            44        400.000
         610          0.007            58        400.000
         611          0.004            34        400.000
         612          0.008            63        400.000
         613          0.004            34        400.000
         614          0.003            24        400.000
         615          0.004            36        400.000
```

```
CONCORDIA UNIVERSITY  GPSS V/6000                     CRM GPSS V/6000 VER. 2.0                              84/07/23. 08.37.01.
   FACILITY          AVERAGE           NUMBER      AVERAGE       SEIZING       PREEMPTING
                   UTILIZATION        ENTRIES     TIME/TRAN     TRANS NO.       TRANS NO.
        616           0.004              37          400.000
        617           0.001              10          400.000
        618           0.006              51          400.000
        619           0.009              74          400.000
        620           0.004              37          400.000
        701           0.002              14          400.000
        702           0.002              14          400.000
        703           0.004              29          400.000
        704           0.003              24          400.000
        705           0.003              24          400.000
        706           0.002              14          400.000
        707           0.002              19          400.000
        708           0.004              29          400.000
        709           0.002              14          400.000
        710           0.003              24          400.000
        711           0.002              14          400.000
        712           0.003              24          400.000
        713           0.002              14          400.000
        714           0.001               9          400.000
        715           0.002              14          400.000
        716           0.002              14          400.000
        717           0.000               4          400.000
        718           0.002              19          400.000
        719           0.004              29          400.000
        720           0.002              14          400.000
       1001           0.000              14           20.000
       1002           0.000              14           20.000
       1003           0.000              29           20.000
       1004           0.000              24           20.000
       1005           0.000              24           20.000
       1006           0.000              14           20.000
       1007           0.000              19           20.000
       1008           0.000              29           20.000
       1009           0.000              14           20.000
       1010           0.000              24           20.000
       1011           0.000              14           20.000
       1012           0.000              24           20.000
       1013           0.000              14           20.000
       1014           0.000               9           20.000
       1015           0.000              14           20.000
       1016           0.000              14           20.000
       1017           0.000               4           20.000
       1018           0.000              19           20.000
       1019           0.000              29           20.000
       1020           0.000              14           20.000
       1101           0.024               4        20000.000
       1102           0.018               3        20000.000
       1103           0.024               4        20000.000
       1104           0.042               7        20000.000
       1105           0.012               2        20000.000
       1106           0.061              10        20000.000
       1107           0.018               3        20000.000
       1108           0.073              12        20000.000
       1109           0.024               4        20000.000
       1110           0.048               8        20000.000
       1111           0.018               3        20000.000
```

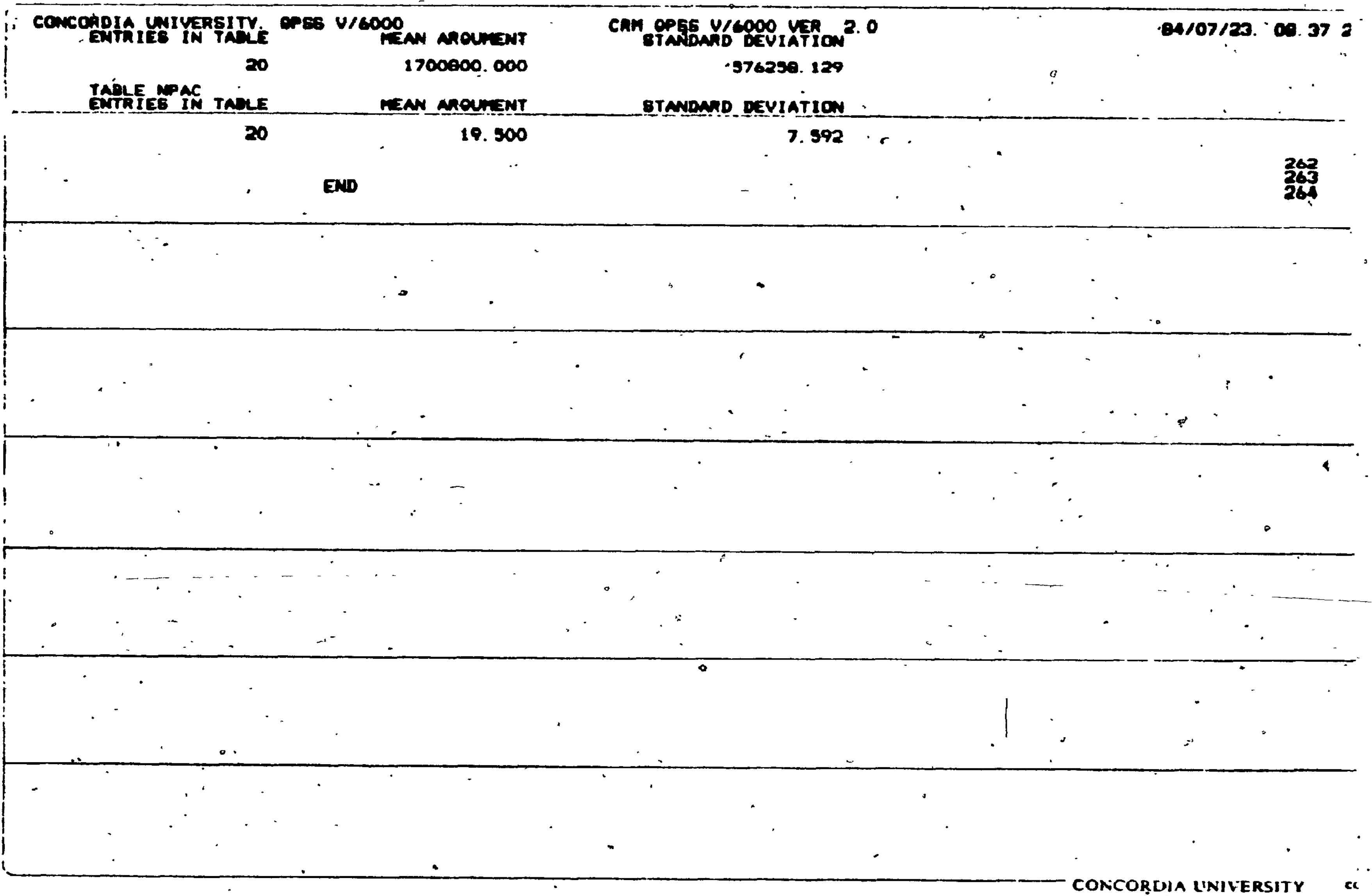
```
CONCORDIA UNIVERSITY. GPSS V/6000                          CRM GPSS V/6000 VER. 2.0                          84/07/23. 08.37.01.
   FACILITY          AVERAGE          NUMBER        AVERAGE        SEIZING      PREEMPTING
                   UTILIZATION        ENTRIES      TIME/TRAN       TRANS NO.     TRANS NO.

       1112           0.036              6        20000.000
       1113           0.024              4        20000.000
       1114           0.006              1        20000.000
       1115           0.030              5        20000.000
       1116           0.012              2        20000.000
       1117           0.006              1        20000.000
       1118           0.030              5        20000.000
       1119           0.042              7        20000.000
       1120           0.042              7        20000.000
       1201           0.001             15          200.000
       1202           0.001             15          200.000
       1203           0.002             30          200.000
       1204           0.002             25          200.000
       1205           0.002             25          200.000
       1206           0.001             15          200.000
       1207           0.001             20          200.000
       1208           0.002             30          200.000
       1209           0.001             15          200.000
       1210           0.002             25          200.000
       1211           0.001             15          200.000
       1212           0.002             25          200.000
       1213           0.001             15          200.000
       1214           0.001             10          200.000
       1215           0.001             15          200.000
       1216           0.001             15          200.000
       1217           0.000              5          200.000
       1218           0.001             20          200.000
       1219           0.002             30          200.000
       1220           0.001             15          200.000
       1301           0.384             15        84506.667
       1302           0.407             15        89626.667
       1303           0.945             30       104080.000
       1304           0.636             25        84016.000
       1305           0.682             25        90096.000
       1306           0.515             15       113493.333
       1307           0.608             20       100400.000
       1308           0.843             30        92760.000
       1309           0.424             15        93306.667
       1310           0.627             25        82768.000
       1311           0.312             15        68773.333
       1312           0.776             25       102528.000
       1313           0.417             15        91706.667
       1314           0.200             10        66160.000
       1315           0.474             15       104293.333
       1316           0.358             15        78720.000
       1317           0.207              5       136400.000
       1318           0.571             20        94220.000
       1319           0.593             30        65253.333
       1320           0.399             15        87786.667
TABLES
   TABLE TTIME
   ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                 20            1747520.000                  670183.616

   TABLE ETIME
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                    CRM GPSS V/6000 VER. 2.0                    84/07/23.  08.37.02.
       ENTRIES IN TABLE            MEAN ARGUMENT         STANDARD DEVIATION
                    20             1702600.000                  654998.305
       TABLE NPAC
       ENTRIES IN TABLE            MEAN ARGUMENT         STANDARD DEVIATION
                    20                  19.000                       6.996
                                                                                                      257
                                                                                                      258
                       CLEAR       X1-X3,X11-X12                                                      259
                       INITIAL     X4,24                **     TWENTY FOUR SLICES     **              260
                       START       20,,,,PSLICE                                                       261
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                      CRM GPSS V/6000 VER. 2.0                                  84/07/23. 08.37.20.
FACILITIES
      FACILITY        AVERAGE        NUMBER       AVERAGE       SEIZING     PREEMPTING
                    UTILIZATION      ENTRIES     TIME/TRAN     TRANS NO.    TRANS NO.

         405            0.004            29       400.000
         406            0.004            29       400.000
         407            0.008            59       400.000
         408            0.008            59       400.000
         409            0.008            59       400.000
         410            0.004            29       400.000
         411            0.008            59       400.000
         412            0.004            29       400.000
         413            0.004            29       400.000
         414            0.004            29       400.000
         415            0.004            29       400.000
         416            0.004            29       400.000
         417            0.005            39       400.000
         418            0.007            49       400.000
         419            0.007            49       400.000
         420            0.004            29       400.000
         421            0.005            39       400.000
         422            0.001             9       400.000
         423            0.003            19       400.000
         424            0.008            59       400.000
         505            0.003            20       400.000
         506            0.002            17       400.000
         507            0.005            39       400.000
         508            0.006            41       400.000
         509            0.005            39       400.000
         510            0.003            21       400.000
         511            0.006            40       400.000
         512            0.003            21       400.000
         513            0.003            18       400.000
         514            0.003            21       400.000
         515            0.003            19       400.000
         516            0.003            21       400.000
         517            0.003            24       400.000
         518            0.005            33       400.000
         519            0.004            32       400.000
         520            0.003            21       400.000
         521            0.003            24       400.000
         522            0.001             7       400.000
         523            0.002            11       400.000
         524            0.005            38       400.000
         605            0.005            37       400.000
         606            0.005            35       400.000
         607            0.010            71       400.000
         608            0.009            67       400.000
         609            0.010            73       400.000
         610            0.005            37       400.000
         611            0.010            73       400.000
         612            0.005            35       400.000
         613            0.005            36       400.000
         614            0.005            38       400.000
         615            0.005            34       400.000
         616            0.005            34       400.000
         617            0.007            49       400.000
         618            0.008            57       400.000
         619            0.008            56       400.000
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                          CRM GPSS V/6000 VER. 2.0                          84/07/23. 08.37.20.
   FACILITY          AVERAGE          NUMBER      AVERAGE       SEIZING     PREEMPTING
                   UTILIZATION        ENTRIES    TIME/TRAN     TRANS NO.     TRANS NO.

        620           0.005              34       400.000
        621           0.007              52       400.000
        622           0.002              13       400.000
        623           0.003              22       400.000
        624           0.010              69       400.000
        705           0.002              14       400.000
        706           0.002              14       400.000
        707           0.004              29       400.000
        708           0.004              29       400.000
        709           0.004              29       400.000
        710           0.002              14       400.000
        711           0.004              29       400.000
        712           0.002              14       400.000
        713           0.002              14       400.000
        714           0.002              14       400.000
        715           0.002              14       400.000
        716           0.002              14       400.000
        717           0.003              19       400.000
        718           0.003              24       400.000
        719           0.003              24       400.000
        720           0.002              14       400.000
        721           0.003              19       400.000
        722           0.001               4       400.000
        723           0.001               9       400.000
        724           0.004              29       400.000
       1005           0.000              14        20.000
       1006           0.000              14        20.000
       1007           0.000              29        20.000
       1008           0.000              29        20.000
       1009           0.000              29        20.000
       1010           0.000              14        20.000
       1011           0.000              29        20.000
       1012           0.000              14        20.000
       1013           0.000              14        20.000
       1014           0.000              14        20.000
       1015           0.000              14        20.000
       1016           0.000              14        20.000
       1017           0.000              19        20.000
       1018           0.000              24        20.000
       1019           0.000              24        20.000
       1020           0.000              14        20.000
       1021           0.000              19        20.000
       1022           0.000               4        20.000
       1023           0.000               9        20.000
       1024           0.000              29        20.000
       1105           0.035               5     20000.000
       1106           0.014               2     20000.000
       1107           0.063               9     20000.000
       1108           0.077              11     20000.000
       1109           0.063               9     20000.000
       1110           0.042               6     20000.000
       1111           0.070              10     20000.000
       1112           0.042               6     20000.000
       1113           0.021               3     20000.000
       1114           0.042               6     20000.000
       1115           0.028               4     20000.000
```

```
CONCORDIA UNIVERSITY. GPSS V/6000                      CRM GPSS V/6000 VER  2.0                              84/07/23.  08.37.21.
   FACILITY          AVERAGE          NUMBER      AVERAGE       SEIZING     PREEMPTING
                   UTILIZATION        ENTRIES    TIME/TRAN      TRANS NO.    TRANS NO.

      1116            0.042          [illegible]   20000.000
      1117            0.028          [illegible]   20000.000
      1118            0.056              8        20000.000
      1119            0.049              7        20000.000
      1120            0.042              6        20000.000
      1121            0.028              4        20000.000
      1122            0.014              2        20000 000
      1123            0.007              1        20000.000
      1124            0.056              8        20000 000
      1205            0.001             15          200.000
      1206            0.001             15          200.000
      1207            0.002             30          200.000
      1208            0.002             30          200 000
      1209            0.002             30          200.000
      1210            0.001             15          200.000
      1211            0.002             30          200.000
      1212            0.001             15          200.000
      1213            0.001             15          200 000
      1214            0.001             15          200.000
      1215            0.001             15          200.000
      1216            0.001             15          200.000
      1217            0.001             20          200.000
      1218            0.002             25          200.000
      1219            0.002             25          200.000
      1220            0.001             15          200 000
      1221            0.001             20          200.000
      1222            0 000              5          200.000
      1223            0.001             10          200.000
      1224            0.002             30          200.000
      1305            0.561             15       107253.333
      1306            0.473             15        90400.000
      1307            0.803             30        76680.000
      1308            0.765             30        73026.667
      1309            0.852             30        81373.333
      1310            0.617             15       117920 000
      1311            0.768             30        92306.667
      1312            0.456             15        87200.000
      1313            0.451             15        86160.000
      1314            0.377             15        72080.000
      1315            0.488             15        93306.667
      1316            0.385             15        73573.333
      1317            0.650             20        93180.000
      1318            0.765             25        87712.000
      1319            0.715             25        81936.000
      1320            0.481             15        91973 333
      1321            0.553             20        79240.000
      1322            0.214              5       122400.000
      1323            0.440             10       126080.000
      1324            0.931             30        88920.000
TABLES
    TABLE TTIME
    ENTRIES IN TABLE        MEAN ARGUMENT        STANDARD DEVIATION
                 20           1746550.000                593001.686
    TABLE ETIME
```



```
CONCORDIA UNIVERSITY, GPSS V/6000            CRM GPSS V/6000 VER  2.0                    84/07/23. 08.37 2
     ENTRIES IN TABLE          MEAN ARGUMENT       STANDARD DEVIATION
                   20            1700800.000               576258.129
     TABLE NPAC
     ENTRIES IN TABLE          MEAN ARGUMENT       STANDARD DEVIATION
                   20                 19.500                    7.592
                                                                                                   262
                         END                                                                       263
                                                                                                   264
```

CONCORDIA UNIVERSITY

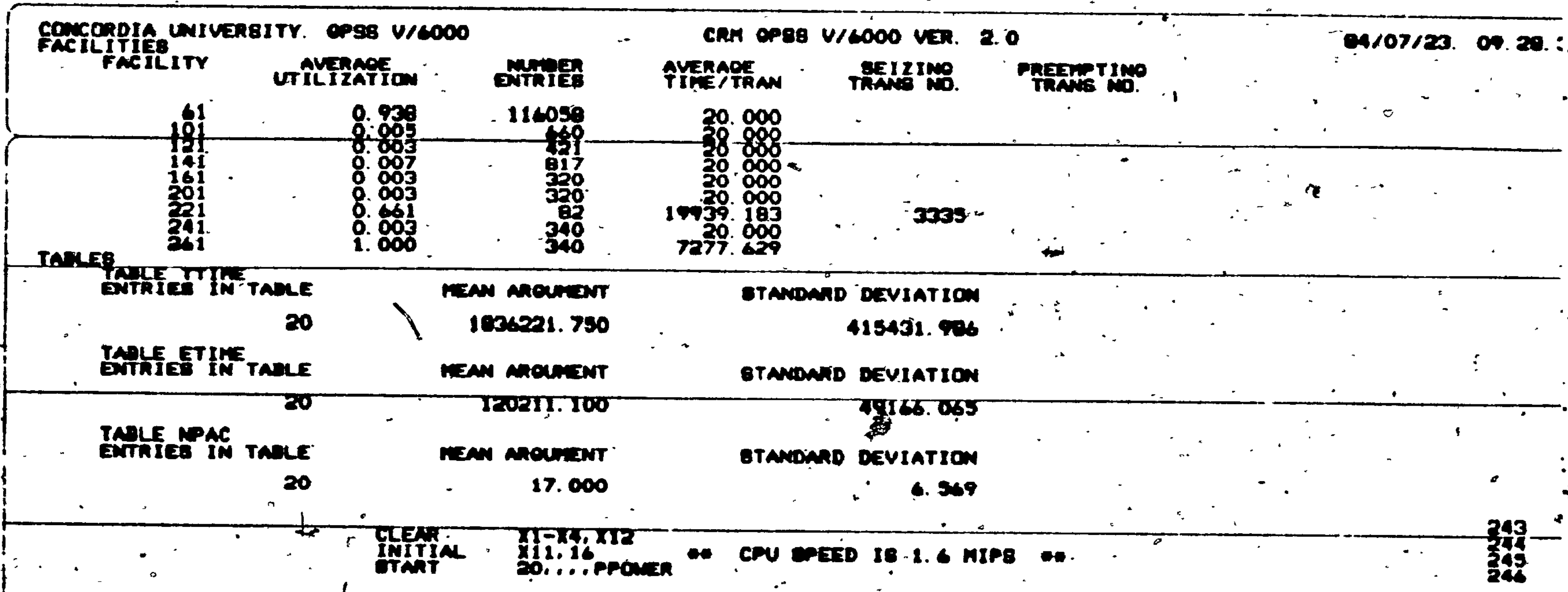
PPOWER
HPL= 20

```
CONCORDIA UNIVERSITY. GPSS V/6000                 CRM GPSS V/6000 VER. 2.0                 84/07/23. 08 30.51.
            BLOCK
            NUMBER  *LOC   OPERATION  A,B,C,D,E,F,G,H,I,J     COMMENTS                                       CARD
                                                                                                          NUMBER
     2              ETIME  TABLE      PF17,0,100,20       ** EXEC TIME PER PROGRAM  **                       220
     3              NPAC   TABLE      PF12,0,5,8       **  NUMBER OF PACKET GROUPS PER PROGRAM **            221
            135            TERMINATE  1                                                                      222
                                                                                                             223
                                                                                                             224
                           RESET                                                                             225
                           INITIAL    X4,1                                                                   226
                           INITIAL    X11,4              **   CPU SPEED IS 0.4 MIPS  **                      227
                           REPORT     PPOWER                                                                 228
                    FAC    TITLE      ,FACILITIES                                                            229
                    TAB    TITLE      ,TABLES                                                                230
                    TAB    INCLUDE    T1-T4/1,2,3,4                                                          231
                           ENDREPORT                                                                         232
                           START      20,,,,PPOWER                                                           233
                                                                                                             234
```

CONCORDIA UNIVERSITY   com

```
CONCORDIA UNIVERSITY  GPSS V/6000                 CRM GPSS V/6000 VER. 2.0                         84/07/23. 08.55.07.
FACILITIES
    FACILITY        AVERAGE        NUMBER      AVERAGE       SEIZING     PREEMPTING
                  UTILIZATION     ENTRIES     TIME/TRAN     TRANS NO.    TRANS NO
         61          0.947        380848        20.000
        101          0.002           700        20.000
        121          0.001           448        20.000
        141          0.002           869        20.000
        161          0.001           340        20.000
        201          0.001           340        20.000
        221          0.219            88     20000.000
        241          0.001           360        20.000
        261          1.000           360     22348.444
TABLES
    TABLE TTIME
    ENTRIES IN TABLE          MEAN ARGUMENT         STANDARD DEVIATION

                  20            6315039.000                1224427.001

    TABLE ETIME
    ENTRIES IN TABLE          MEAN ARGUMENT         STANDARD DEVIATION

                  20             398400.000                 167464.183

    TABLE NPAC
    ENTRIES IN TABLE          MEAN ARGUMENT         STANDARD DEVIATION

                  20                 18.000                      6.366

                  CLEAR       X1-X4,X12                                                                  235
                  INITIAL     X11.8        **  CPU SPEED IS 0.8 MIPS  **                                 236
                  START       20,...,PPOWER                                                              237
                                                                                                         238
```

```
CONCORDIA UNIVERSITY.  GPSS V/6000              CRM GPSS V/6000 VER. 2.0                                84/07/23. 09.17.01.
FACILITIES
   FACILITY        AVERAGE        NUMBER        AVERAGE        SEIZING        PREEMPTING
                 UTILIZATION      ENTRIES      TIME/TRAN       TRANS NO.      TRANS NO.

       61           0.961         194577         20.000
      101           0.003            700         20.000
      121           0.002            439         20.000
      141           0.004            853         20.000
      161           0.002            340         20.000
      201           0.002            340         20.000
      221           0.390             79      20000.000
      241           0.002            360         20.000
      261           1.000            360      11244.222
TABLES
   TABLE TTIME
   ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION

                 20          3300928.000                  524327.542

   TABLE ETIME
   ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION

                 20           198750.000                   52958.400

   TABLE NPAC
   ENTRIES IN TABLE        MEAN ARGUMENT          STANDARD DEVIATION

                 20               18.000                       5.712

                                                                                                      239
                 CLEAR        X1-X4,X12                                                               240
                 INITIAL      X11,12      ** CPU SPEED IS 1.2 MIPS **                                 241
                 START        20,...,PPOWER                                                           242
```



```
CONCORDIA UNIVERSITY. GPSS V/6000                    CRM GPSS V/6000 VER. 2.0                      84/07/23. 09.28.:
FACILITIES
     FACILITY          AVERAGE         NUMBER       AVERAGE      SEIZING     PREEMPTING
                     UTILIZATION       ENTRIES     TIME/TRAN    TRANS NO.     TRANS NO.
         61             0.938          116058        20.000
        101             0.005             440        20.000
        121             0.003             421        20.000
        141             0.007             817        20.000
        161             0.003             320        20.000
        201             0.003             320        20.000
        221             0.661              82     19939.183        3335
        241             0.003             340        20.000
        261             1.000             340      7277.629
TABLES
     TABLE TTIME
     ENTRIES IN TABLE          MEAN ARGUMENT              STANDARD DEVIATION

                   20            1836221.750                      415431.986
     TABLE ETIME
     ENTRIES IN TABLE          MEAN ARGUMENT              STANDARD DEVIATION
                   20             120211.100                       49166.065
     TABLE NPAC
     ENTRIES IN TABLE          MEAN ARGUMENT              STANDARD DEVIATION

                   20                 17.000                           6.569

                     CLEAR        X1-X4,X12                                                               243
                     INITIAL      X11,16         **  CPU SPEED IS 1.6 MIPS  **                            244
                     START        20,,,,PPOWER                                                            245
                                                                                                          246
```

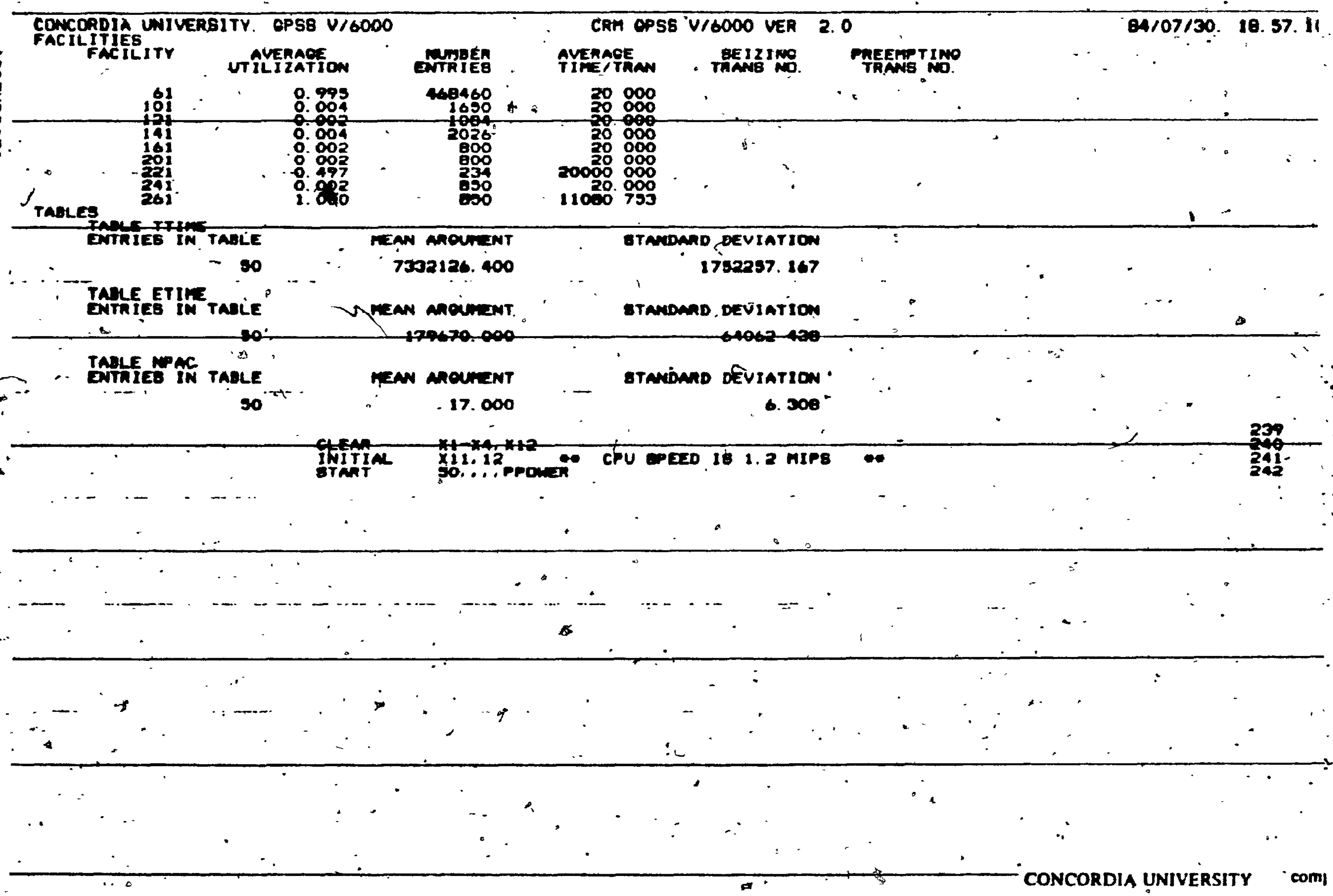
```
CONCORDIA UNIVERSITY. GPSS V/6000                  CRM GPSS V/6000 VER. 2.0                            84/07/23. 09.42.
FACILITIES
     FACILITY        AVERAGE        NUMBER      AVERAGE       SEIZING      PREEMPTING
                   UTILIZATION      ENTRIES    TIME/TRAN      TRANS NO.     TRANS NO.
            61         0.957         94681       20.000
           101         0.007           670       20 000
           121         0.004           440       20.000
           141         0.008           831       20.000
           161         0.003           325       20.000
           201         0.003           325       20.000
           221         0.970            96    19999 687         3343
           241         0.003           345       20 000
           261         1.000           345     5734.493
TABLES
     TABLE TTIME
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION

                   20'             1330124.500                   503185.914
     TABLE ETIME
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION

                   20               95537.500                    41668.355
     TABLE NPAC
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION

                   20                  17.250                        8.347

                        CLEAR      X1-X4,X12                                                                247
                        INITIAL    X11,20           ** CPU SPEED IS 2.0 MIPS **                              248
                        START      20....,PPOWER                                                            249
                                                                                                            250
```

```
CONCORDIA UNIVERSITY.  GPSS V/6000                    CRM GPSS V/6000 VER.  2.0                      84/07/23.  09.47.40.
FACILITIES
     FACILITY         AVERAGE          NUMBER        AVERAGE        SEIZING      PREEMPTING
                   UTILIZATION        ENTRIES      TIME/TRAN      TRANS NO.      TRANS NO.

          61            0.930           65385         20.000
         101            0.009             610         20.000
         121            0.005             384         20.000
         141            0.011             742         20.000
         161            0.004             295         20.000
         201            0.004             295         20.000
         221            0.996              70      19997.429          3198
         241            0.004             315         20.000
         261            1.000             315       4462.095
TABLES
   TABLE TTIME
   ENTRIES IN TABLE             MEAN ARGUMENT               STANDARD DEVIATION

                 20               1037340.000                       289308.926
   TABLE ETIME
   ENTRIES IN TABLE             MEAN ARGUMENT               STANDARD DEVIATION
                 20                 67190.000                        30439.273
   TABLE NPAC
   ENTRIES IN TABLE             MEAN ARGUMENT               STANDARD DEVIATION

                 20                    15.750                            6.935

                                                                                                       251
                     CLEAR          X1-X4,X12                                                          252
                     INITIAL        X11,24                                                             253
                     START          20....PPOWER          **  CPU SPEED IS 2.4 MIPS  **                254
                                                                                                       255
```

```
CONCORDIA UNIVERSITY.  GPSS V/6000                - CRM GPSS V/6000 VER. 2.0                     84/07/23. 09 59.20.
FACILITIES
     FACILITY          AVERAGE          NUMBER       AVERAGE        SEIZING       PREEMPTING
                     UTILIZATION        ENTRIES     TIME/TRAN       TRANS NO.      TRANS NO.
         61             0.967            70635        20.000
        101             0.010              720        20.000
        121             0.006              442        20.000
        141             0.012              911        20.000
        161             0.005              350        20.000
        201             0.005              350        20.000
        221             0.990               73     19809.726         3554
        241             0.005              370        20.000
        261             1.000              370      3945.773
TABLES
     TABLE TTIME
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION

                   20             1042500.600                   289642.584
     TABLE ETIME
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION
                   20               68991.450                    31021.379
     TABLE NPAC
     ENTRIES IN TABLE           MEAN ARGUMENT           STANDARD DEVIATION

                   20                  18.500                        6.708
                          END                                                                              256
                                                                                                           257
```

PPOWER
MPL=50

```
CONCORDIA UNIVERSITY. GPSS V/6000                CRM GPSS V/6000 VER. 2.0                 84/07/30. 16.03.1
           BLOCK                                                                                         CARD
           NUMBER  *LOC   OPERATION  A,B,C,D,E,F,G,H,I,J        COMMENTS                                 NUMBER
     2             ETIME  TABLE      PF17,0,100,20         ** EXEC TIME PER PROGRAM  **                   220
     3             NPAC   TABLE      PF12,0,5,8       **  NUMBER OF PACKET GROUPS PER PROGRAM **          221
           135            TERMINATE  1                                                                    222
                                                                                                          223
                                                                                                          224
                                                                                                          225
                          RESET                                                                           226
                          INITIAL    X4,1                                                                 227
                          INITIAL    X11,4            **   CPU SPEED IS 0.4 MIPS  **                      228
                          REPORT     PPOWER                                                               229
                   FAC    TITLE      ,FACILITIES                                                          230
                   TAB    TITLE      ,TABLES                                                              231
                   TAB    INCLUDE    T1-T4/1,2,3,4                                                        232
                          ENDREPORT                                                                       233
                   *      START      50,...PPOWER                                                         234
                                                                                                          235
                          CLEAR      X1-X4,X12                                                            236
                          INITIAL    X11,8         **  CPU SPEED IS 0.8 MIPS   **                         237
                          START      50,...PPOWER                                                         238
```

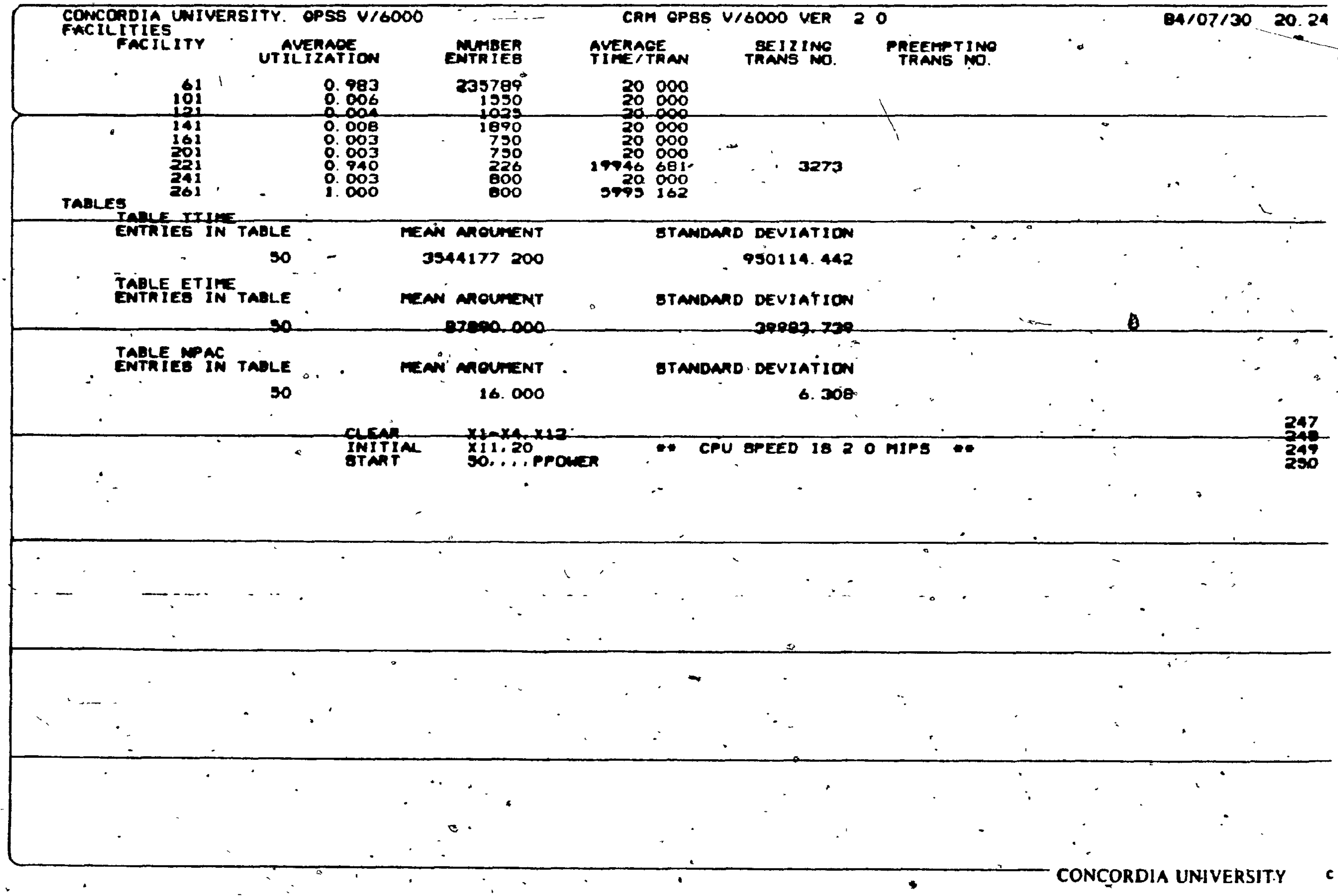


```
CONCORDIA UNIVERSITY. GPSS V/6000                    CRM GPSS V/6000 VER  2.0                                 84/07/30. 18.57.1
FACILITIES
    FACILITY         AVERAGE         NUMBER      AVERAGE       SEIZING       PREEMPTING
                   UTILIZATION       ENTRIES     TIME/TRAN     TRANS NO.     TRANS NO.

         61           0.995          468460       20 000
        101           0.004            1650       20 000
        121           0.002            1084       20 000
        141           0.004            2026       20 000
        161           0.002             800       20 000
        201           0 002             800       20 000
        221           0.497             234    20000 000
        241           0.002             850       20.000
        261           1.000             850    11080 753
TABLES
    TABLE TTIME
    ENTRIES IN TABLE              MEAN ARGUMENT              STANDARD DEVIATION

                  50                7332126.400                     1752257.167

    TABLE ETIME
    ENTRIES IN TABLE              MEAN ARGUMENT              STANDARD DEVIATION

                  50                 179670.000                       64062.438

    TABLE NPAC
    ENTRIES IN TABLE              MEAN ARGUMENT              STANDARD DEVIATION

                  50                     17.000                           6.308

                        CLEAR        X1-X4,X12                                                                             239
                        INITIAL      X11,12      ** CPU SPEED IS 1.2 MIPS    **                                            240
                        START        50,...,PPOWER                                                                         241
                                                                                                                           242
```

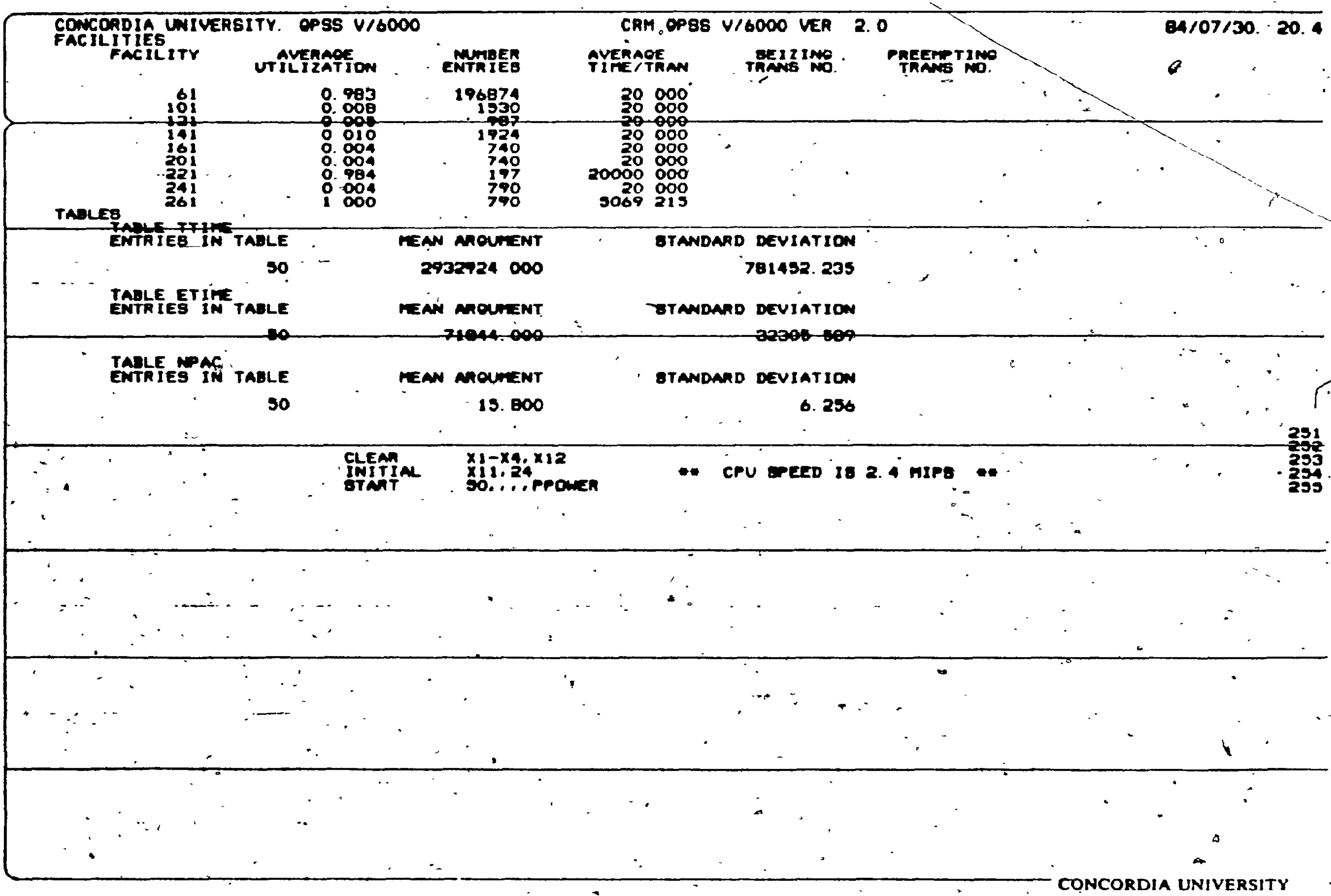
CONCORDIA UNIVERSITY com

```
CONCORDIA UNIVERSITY' GPSS V/6000                    CRM GPSS V/6000 VER. 2.0                        84/07/30. 20 00.59
FACILITIES
    FACILITY        AVERAGE        NUMBER      AVERAGE      SEIZING     PREEMPTING
                  UTILIZATION      ENTRIES    TIME/TRAN     TRANS NO    TRANS.NO.

       61            0.989         319546       20.000
      101            0.005           1630       20.000
      131            0.003           1070       20.000
      141            0.006           2031       20.000
      161            0.002            790       20.000
      201            0.002            790       20.000
      221            0.714            231    19963.087        3582
      241            0.003            840       20.000
      261            1.000            840     7689.451
TABLES
    TABLE TTIME
    ENTRIES IN TABLE         MEAN ARGUMENT          STANDARD DEVIATION

                  50           4947804.040                 1224021.803
    TABLE ETIME
    ENTRIES IN TABLE         MEAN ARGUMENT          STANDARD DEVIATION

                  50            120184.400                   49758.413
    TABLE NPAC
    ENTRIES IN TABLE         MEAN ARGUMENT          STANDARD DEVIATION

                  50                16.800                       6.682

                       CLEAR       X1-X4,X12                                                         243
                       INITIAL     X11,16       **  CPU SPEED IS 1.6 MIPS  **                        244
                       START       80,,,,PPOWER                                                      245
                                                                                                     246
```

CONCORDIA UNIVERSITY com



```
CONCORDIA UNIVERSITY. GPSS V/6000                CRM GPSS V/6000 VER  2 0                        84/07/30  20.24
FACILITIES
     FACILITY       AVERAGE         NUMBER       AVERAGE       SEIZING      PREEMPTING
                  UTILIZATION       ENTRIES      TIME/TRAN     TRANS NO.     TRANS NO.

         61         0.983          235789        20 000
        101         0.006            1550        20 000
        121         0.004            1025        30.000
        141         0.008            1890        20 000
        161         0.003             750        20 000
        201         0.003             750        20 000
        221         0.940             226     19946 681           3273
        241         0.003             800        20 000
        261         1.000             800      5995 162
TABLES
     TABLE TTIME
     ENTRIES IN TABLE            MEAN ARGUMENT              STANDARD DEVIATION

                   50              3544177 200                    950114.442

     TABLE ETIME
     ENTRIES IN TABLE            MEAN ARGUMENT              STANDARD DEVIATION

                   50                87880.000                     39983.739

     TABLE NPAC
     ENTRIES IN TABLE            MEAN ARGUMENT              STANDARD DEVIATION

                   50                   16.000                         6.308

                        CLEAR        X1-X4,X12                                                         247
                        INITIAL      X11,20           **   CPU SPEED IS 2 0 MIPS   **                   248
                        START        50,...PPOWER                                                      249
                                                                                                       250
                                                                              CONCORDIA UNIVERSITY   c
```



```
CONCORDIA UNIVERSITY. GPSS V/6000                  CRM GPSS V/6000 VER  2.0                          84/07/30. 20.4
FACILITIES
     FACILITY        AVERAGE        NUMBER       AVERAGE        SEIZING     PREEMPTING
                   UTILIZATION      ENTRIES     TIME/TRAN       TRANS NO.    TRANS NO.

          61           0.983         196874        20.000
         101           0.008           1530        20.000
         131           0.008            987        20.000
         141           0.010           1924        20.000
         161           0.004            740        20.000
         201           0.004            740        20.000
         221           0.984            197     20000.000
         241           0.004            790        20.000
         261           1.000            790      5069.215
TABLES
     TABLE TTIME
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                   50           2932924.000                 781452.235
     TABLE ETIME
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                   50             71844.000                  32305.509
     TABLE NPAC
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                   50                15.800                      6.256

                                                                                                            251
                                                                                                            252
                    CLEAR        X1-X4,X12                                                                  253
                    INITIAL      X11,24            **   CPU SPEED IS 2.4 MIPS   **                          254
                    START        50,,,,PPOWER                                                               255
```

CONCORDIA UNIVERSITY

```
CONCORDIA UNIVERSITY. GPSS V/6000                   CRM GPSS V/6000 VER. 2 0                          84/07/30. 20 57.3
FACILITIES
    FACILITY        AVERAGE          NUMBER        AVERAGE        SEIZING      PREEMPTING
                  UTILIZATION        ENTRIES      TIME/TRAN       TRANS NO      TRANS NO.

          61          0.988          158563          20.000
         101          0.010            1610          20 000
         121          0.006             789          20 000
         141          0.012            1961          20.000
         161          0.005             780          20 000
         201          0 005             780          20 000
         221          0.997             160       19997.262          3120
         241          0.005             830          20 000
         261          1.000             830        3864 964
TABLES
    TABLE TTIME
    ENTRIES IN TABLE          MEAN ARGUMENT           STANDARD DEVIATION

                  50           2072089.240                   750287.895

    TABLE ETIME
    ENTRIES IN TABLE          MEAN ARGUMENT           STANDARD DEVIATION

                  50             87221.860                    29357.489

    TABLE NPAC
    ENTRIES IN TABLE          MEAN ARGUMENT           STANDARD DEVIATION

                  50                16 600                        7.918

                      END                                                                                       256
                                                                                                                257
```

MPL=50 PPOWER
CPTM = 3.2 MIPS

```
CONCORDIA UNIVERSITY. GPSS V/6000                 CRM GPSS V/6000 VER  2 0                                  84/08/21.  22.14.17.
FACILITIES
     FACILITY        AVERAGE        NUMBER      AVERAGE      SEIZING     PREEMPTING
                   UTILIZATION     ENTRIES     TIME/TRAN    TRANS NO.    TRANS NO.
          61          0.983        126773        20.000
         101          0.013          1650        20.000
         121          0.008           977        20.000
         141          0.016          2016        20.000
         161          0.006           800        20.000
         201          0.006           800        20.000
         221          0.993           128     19995.430        4564
         241          0.007           850        20.000
         261          1.000           850      3032.712
TABLES
     TABLE TTIME
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                   50             1751498.900                  511762.282
     TABLE ETIME
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION
                   50               43920.000                   21673.419
     TABLE NPAC
     ENTRIES IN TABLE          MEAN ARGUMENT          STANDARD DEVIATION

                   50                  17.000                       6.999
                         END                                                                                   260
                                                                                                               261
```

12.0 REFERENCES.

- (BAS-1).-T. R. Bashkow & H Sullivan.-"A large scale, homogenious, fully distributed parallel machine" I & II.- Proceeding in Computer Architecture.- 1978 (pp 105-124).

- (BAS-2).- T.R. Bashkow & H. Sullivan.- "Node design for a distributed terminal network".- IEEE on Distributed Processing Trends & Applications.- 1978 (pp. 84-91).

- (BER-1).- A. Bernstein.- "Interprocess Communication Facilities for Network Operating Systems" COMPUTER June 1974.

- (BRY-1).-R.E. Bryant.- "Simulation of packet communication architecture computer systems". MIT/LCS/TR-188.- Nov 1977.- (Available from U. de Montreal).

- (BRO-1).- S.Browns.- "Message based communication in a distributed module arquitecture".- M.C.S. Thesis.- Concordia University.- Aug. 1978.

- (DAV-1).- J. Davidson et al. "A generalized multi-microprocessor system" IEEE Distributed Processing Trends & Applications. 1978 pp. 92-99.

- (CAR-1).- L.R. Carter.- "An Analysis of Pascal Programs". UMI Research Press.- 1982

- (ENS-1).- P. H. Enslow.- "Multiprocessor Organization.- A survey".-Tutorial on Distributed Processing.- IEEE Catalog No. EHO 127-1.- 1978

- (FUL-1).- S. H. Fuller.- "Multi-Microprocessors: An overview and working example".- Tutorial on distributed processing.- IEEE Catalog No. 127-1 .- 1978.- pp. 368-380

- (GOO-1).-G. Goos.- "Software engineering, an advanced course".- F. L. Bauer, Springer Verlag 1975, pp. 29-46

- (HAL-1).-R. H. Halstead.- "Multiple processor implementation of a message-passing system".- Thesis disertation.- MIT/LCS/TR-198.- 1978

- (HAN-1).-P. B. Hansen.-"Operating Systems Principles".- Prentice Hall, 1973.

- (HOL-1).- R. C. Holt.- "Structured Concurrent Programming with O.S. aplications" Addisson & Wessley.- 1978

- (IBM-1).- I.B.M.- "Analysis of some queueing models in real-time systems", Tech. Publ. GF20-0007-1.- 1971

- (JAF-1).-H. Jafari.- "A new modular loop architecture for distributed computer systems".- Distributed Processing Trends & Applications.- IEEE 1978, pp. 72-77

- (JEN-1).- E. D. Jensen.- "The Honeywell experimental distributed processor. An overview".- COMPUTER, Jan. 1978, pp. 28-38.

- (JEN-2).-E.D.Jensen.- "Partitioning and assignment of distributed processing softare".- COMPCON 76, Sept. 1976, pp. 348-352.

- (JON-1).- A.K.Jones.- "STAR-OS, a multiprocessor operating system for the support of Task Forces".- GIGOPS.- ACM Proceedings on Op. Systems, 1979. pp. 117-127.

- (JON-2).-A.K.Jones: "Software management of Cm*: A distributed multiprocessor".- AFIPS.- National Computer Conference 1977. pp 92-105 pp. 657-663.-

- (KAR-1).-S.I. Kartashew.- "A multicomputer system with dynamic architecture".- IEEE Transactions on Computers.- Vol. C-28, No. 10 Oct. 1979.

- (LAG-1).-K. Lagally.- "Synchronization in a layered system".- Available from A. Diez.

- (LIU-1).-M.T.Liu.- "Distributed Loop Computer Networks".- Advances in Computers.- Vol 17.- Prentice Hall 1978.

- (MAD-1).- A.W. Madison.- "Characteristics of Program Localities". UMI Research Press.- 1982.

- (MAR-1).- J. Martin.- "Systems Analysis for Data Transmission". Prentice Hall.- 1972

- (MCK-1).- R.M.McKeag.- "THE Multiprogramming System".- Studies in operating systems.- Academic Press.- 1976 pp. 145-184.-

- (MON-1).-W.A.Montgomery.- "Robust Concurrency control for distributed information".- PHD Disertation.- MIT/LCS/TR-207 Dec 1978.

- (MYE-1).-G.J.Myers.- "Advances in Computer Architecture).-J. Wiley & Sons.- 1978.

- (NEW-1).-E.E. Newhall.- "Traffic flow in a distributed loop switching system".- Comp. communic. networks and Teletraffic".- Brooklyn April 1972 pp. 29-46.

- (ORG-1).-E.I. Organic.- "Computer System Organization . The B5700/B6700 Series".-Academic Press, 1973.

- (OUS-1).- J.K.Ousterhout.- "MEDUSA: An experiment in distributed operating systems structures".- Comm. ACM, Feb 1980, V23, No. 2, pp. 92-105.

- (PAR-1).-D. Parnas.- "On a Buzzword: Hierarchical structure".- Proceedings IFIP Congress 1974 pp. 336-339

- (PIE-1).-J.R.Pierce.- "How far can Data Loops go?.- IEEE Transactions on Communications.- Vol Com-20 No 3 June 1972 pp. 527-530.-

- (POH-1).- A.V. Pohm & O.P. Agrawal.- "High speed memory systems". Prentice Hall, 1983.

- (PRO-1).-W.G.Probst & G.V. Bochmann.- "Operating systems design with computer network communications protocols".- IEEE %th. Data Comm Symposium.- Utah 1977 pp. 4-19, 4-25.

- (RIC-1).-R.Rice.- "SYMBOL.- A major departure from classic software dominated von Neuman computer systems".- AFIPS Spring Conference 1971, pp. 575-587.

- (SV)-1).-L. Svobodoba, B. Liskow.- "Distributed Computer Systems: Stucture and semantics" MIT/LCS/TR-215, March 1979.- Available from U. de Montreal.

- (SWA-1).- R.J.Swan.- "The implementation of the Cm* multimicroprocessor".- AFIPS.- National Computer Conference 1977, pp. 645-655.

- (WUL-1).- W.Wulf, E. Cohen.- "Hydra: The kernel of a multiprocessor operating system" .- Comm. ACM, June 1974, pp. 337-345.