ESD-TR-66-137

(Final Report)

# THE APPLICATION OF LARGE-SCALE COMPUTERS TO U. S. AIR FORCE INFORMATION SYSTEMS

John B. Campbell
John P. McCabe
Essie S. Nevans

March 1966

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Mass. 01730

Project: 2801

Task: 01

Distribution of this document is unlimited.

ESRCP

(Prepared under Contract No. AF 19(628)-4963, by General Electric Company, Washington, D. C.)

When US Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

ESD-TR-66-137

(Final Report)

# THE APPLICATION OF LARGE-SCALE COMPUTERS TO U. S. AIR FORCE INFORMATION SYSTEMS

John B. Campbell
John P. McCabe
Essie S. Nevans

March 1966

DIRECTORATE OF COMPUTERS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Mass. 01730

Project: 2801

Task: 01

Distribution of this document is unlimited.

(Prepared under Contract No. AF 19(628)-4963, by General Electric Company, Washington, D. C.)

## FOREWORD

This final report has been prepared by the Information Systems Operation of the General Electric Company, Washington, D. C., and details the study efforts conducted during January 15, 1965, to January 15, 1966. This work consists of Task #01, Project #2801, of Contract No. AF 19(628)-4963. Lieutenant S. B. Carpenter, ESRCP, Electronic Systems Division, was Contract Monitor for this contract. The authors take this opportunity to express their appreciation of the able assistance given by: Messrs. Stanley B. Rosen and Frank A. Frangione of the General Electric Company; Lieutenant S. B. Carpenter of ESD as Technical Representative, and Lieutenant John Curtis, also of ESD, for finance system information.

## REVIEW AND APPROVAL

This technical report has been reviewed and is approved.

Stewart B. Carpenter

STEWART B. CARPENTER, Ist Lt., USAF
Contract Monitor, Computer Program Division

# ABSTRACT

Two centralized Air Force systems — one dealing with finance and one with personnel assignment — were used to study applications of large-scale computers. A generalized time-sharing computer system was modeled and simulation was made to measure query response time for various hypothetical conditions.

# TABLE OF CONTENTS

Section | Page

I. INTRODUCTION 1

II. STUDY OF A CENTRALIZED AIR FORCE PAY SYSTEM 3

1. Introduction 3
2. Characteristics of Study System 3
3. Alternative Storage Devices 5
3.1 Univac FASTRAND Drum 5
3.2 IBM 1302 Disc Model 2 7
3.3 IBM 2321 Data Cell 7
3.4 RCA Model 3488 8
3.5 General Electric Disc — DSU 250 9
3.6 Honeywell 1800 Disc 9
4. File Organization 9
5. Batch Processing 10
6. Conclusions and Recommendations 11

III. STUDY OF AIR FORCE PERSONNEL ASSIGNMENT SYSTEM 12

1. Introduction 12
2. System Design Requirements 12
3. System Configuration 14
3.1 Configuration A 14
3.2 Configuration B 17
3.3 Configuration C 18
3.4 Configuration D 20
4. Assignment Algorithm 20
5. Design Alternatives 21
6. Recommendations 22

IV. STUDY OF A TIME-SHARED PERSONNEL SYSTEM 24

1. Introduction 24
2. Centralized Computation System 24
2.1 Assumed System Functions 24
2.2 Description of Model Studied 25
2.3 Performance Characteristics to be Estimated 25
2.4 System Design and Load Factors 25
2.5 Assumptions on System Design 27
2.6 GPSS Simulation 28
2.7 Simulation Results 28
2.8 Discussion of Results 38
3. Decentralized Computation Model 41
4. Extension of the Model to Other Systems 42

TABLE OF CONTENTS (continued)

Page

Appendix I FILE PARTITIONING ANALYSIS 43

Appendix II A MATHEMATICAL MODEL OF A TIME-SHARING SYSTEM 54

Appendix III PERSONNEL ASSIGNMENT ALGORITHMS 61

REFERENCES 67

BIBLIOGRAPHY 68

# LIST OF ILLUSTRATIONS

| Figure | | Page |
|---|---|---|
| 1. | Military Pay System | 4 |
| 2. | Summary Chart | 6 |
| 3. | Recycling Available Men to Open Jobs | 13 |
| 4. | Configuration A | 15 |
| 5. | Configuration C | 19 |
| 6. | Summary Chart | 23 |
| 7. | GPSS Flow Chart | 26 |
| 8. | Simulation Flow | 29 |
| 9. | GPSS III Flow Chart | 32 |
| 10. | GPSS III Assembly Input for Run 3 | 36 |
| Table I. | Simulation Results for Three Cases Studied | 39 |

# SECTION I

# INTRODUCTION

During the past year, the Information Systems Operation of the General Electric Company has conducted a series of studies pertaining to the application of large-scale computer systems to information storage and retrieval. Specifically two Air Force functions were examined to determine the feasibility of centralizing the tasks at a computer center with remote access.

Several aspects are important in the specification of a large computer system. When the processing load is great, due to bulk of data or the frequency of querying, it is necessary that prime capability be efficiently utilized among tasks. As possible aids in the design of this type of system, these studies pursue gains in efficiency made possible by both (1) data organization techniques, and (2) on-line time sharing of data and processing equipment.

The applications examined: (1) an overall pay system, and (2) a system to aid in the assignment of personnel to jobs, proved interesting in their demands upon large-scale data-handling and manipulation capabilities. Since hypothetical requirements were frequently imposed on the systems to facilitate the studies, specific conclusions should not be drawn about either system.

The hypothetical Air Force pay system requires on-line querying by a large number of users of an extensive and highly dynamic data base. The matching of men and jobs by computer requires an assignment algorithm which will make consistent, acceptable assignments using the broadest spectrum of candidates. The personnel files used in assignment are also large and frequently updated. The timely communication and acceptance of updates add a significant load to both study systems.

These Air Force applications are examples of the classes of functions which can be handled using on-line, remote-inquiry computer systems. Examination of these tasks provided a basis for constructing a generalized time-shared computer system model, which is the third in a series of studies. Simulation cf the general model was performed to measure the adequacy of configurations for a set of hypothetical requirements.

Feasibility of both the pay and man-job match systems was shown and each was examined as a time-sharing type of application. The generalized time-sharing model showed centralization of all computational power to be more economical than distributing logical capability to remote stations.

Three supporting analytic studies were performed and are contained in Appendixes to avoid unnecessary detail in the body of the report. The first analytical study deals with a means for partitioning a large file to permit, in some cases, greatly reduced searching times. The second deals with a mathematical model for a time-shared computer system which allows for analytical calculation of processing times at each terminal as a function of system loading. The third investigates three computational algorithms for performing man-job match calculations. Estimates of processing times are given, and the methods compared.

The following sections address each study in detail. Specific results, conclusions, and recommendations are presented with each study. The three supporting analytic studies are referenced when appropriate.

# SECTION II

# STUDY OF A CENTRALIZED AIR FORCE PAY SYSTEM

## 1. Introduction

This section is devoted to a study to determine the feasibility of designing an Air Force Centralized Pay System involving multiple users on a real-time basis. The principal requirement is that of responding to rapid querying of an extremely large file. The system load of more than one query per second of a file whose size is approximately 2 billion characters rules out more conventional tape or drum configurations. This system is characteristic of many systems such as the following: 1) personnel systems for keeping financial records, 2) logistics system for keeping track of parts and/or reliability histories, and 3) large document storage and retrieval systems. Only the very recent market has seen the emergence of devices for randomly storing billions of characters at a cost per character that does not greatly exceed tape storage costs. The following paragraphs describe the characteristics of the Air Force pay system which was used for study, present various configurations which use the storage devices examined, and offer recommendations for the implementation of such a system.

## 2. Characteristics of Study System

The pay system has two functions — computing and authorizing pay for all military personnel, and compiling and presenting statistics on fund allocations. There are two principal files — a master file with one record for each individual, and the Summary File composed of fund totals. Attention focuses on the processing required by the incoming queries since it is this load which taxes the equipment most. Report generation and other routine and regular processing can be considered to be performed "off-line" as far as the system is concerned since there is no rigorous time constraint — night shifts can be used.

There are many users querying the computer system at the rate of 58,000 questions per day. (See Figure 1.) Each one of these requires retrieval from the 2 billion-character master file of one personnel record, averaging 2,000 characters.

Eighty-one percent of these or 47,000 make some change to the individual's record so that it must be read back into storage in its corrected form. The remaining 11,000 daily queries do not alter the data and no second access is needed. It is assumed that every change which is made to a personnel record will also cause a change in the Summary File in order to update the statistics compilation. This update will also require two accesses of

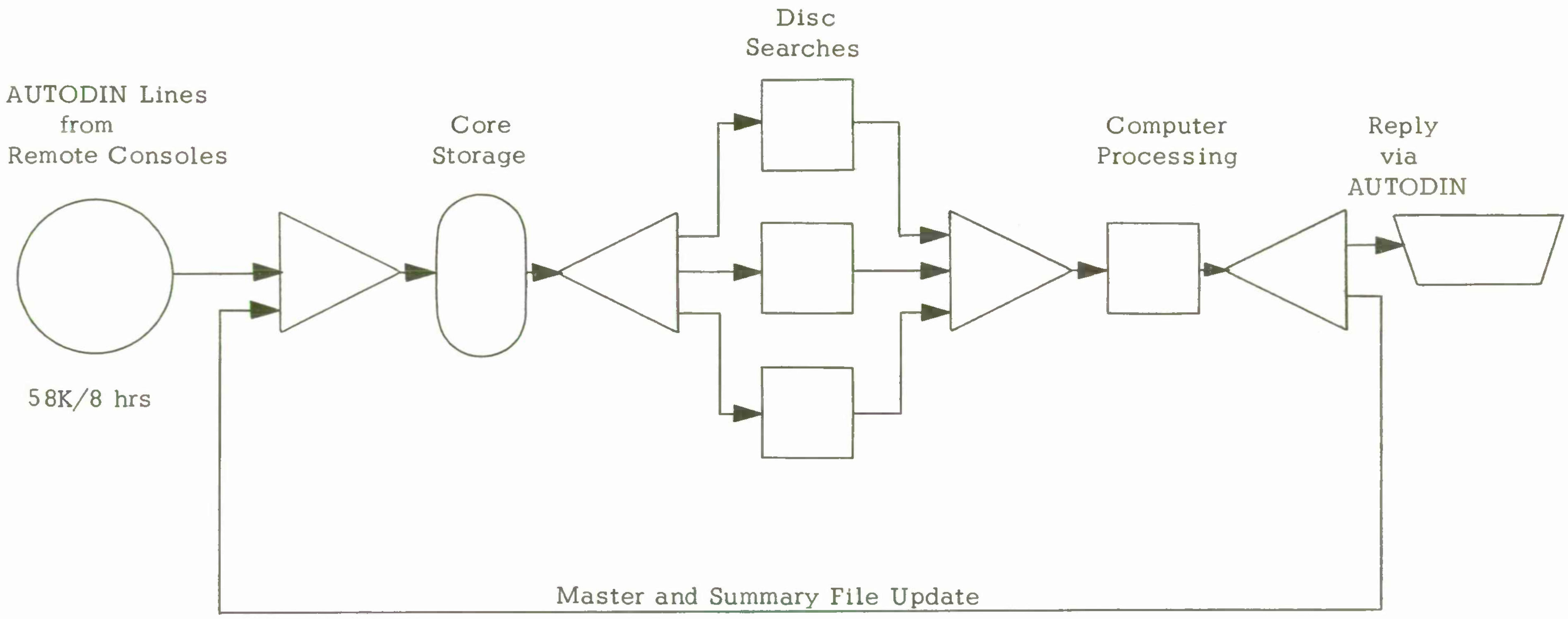


Figure 1. Military Pay System

the storage device so that the corrected entry can be recorded. Based on this load, the bulk storage device will see 105,000 search requests in the course of one day-shift. In addition, it will be accessed 94,000 times when the specific location has been previously determined. These "second" accesses occur when a corrected record is ready to be read back into place in the master file. In the case of using a drum or disc as the storage medium, one-half revolution time is the average access time for "second" accesses.

It is also assumed that only limited computational capability is required to perform these operations. The standard programs needed to compute pay under a variety of options are straightforward and are assumed to be held in core at all times and do not put additional searching load on the bulk storage devices. Assuming that the average number of instructions executed per query is less than 3,000, the average processing time per query will be less than 10 milliseconds for the range of central processors examined. Those computers offering peripheral storage in billions of characters operate within the time frame given above.

Since this is a feasibility study, the aspects of buffering, queue size, and overall time delays were not explicitly examined. It is assumed that the arrivals are evenly spaced over an eight-hour day and that they are stored in core upon arrival until the processor is ready to answer them.

## 3 Alternative Storage Devices

Characteristics and advantages of various bulk storage devices are given as follows.[1] The range of times within which the processing load would fall is presented. Transfer rates are given, but since access time is the overriding time constraint, read and write time can always be overlapped so as not to cause additional delay. This is also true of processing time as demonstrated above. A summary of equipment characteristics is provided in Figure 2.

### 3.1 Univac FASTRAND Drum

- The data file in the study system requires two subsystems for a total of 2.11 billion characters stored on 32 drum units with two controllers.
- The transfer rate is 183 kc.

[1] Information derived from appropriate manuals as listed in References [1] through [10].

| Name | No. of Controllers | (Char.) Size 6-bit | (kc) Trans. Rate | Read Time | Access Time (hrs) Worst | Access Time (hrs) Best | Percent Growth* |
|---|---|---|---|---|---|---|---|
| UNIVAC Drum | 2 | $2.11x10^{9}$ | 183 | .31 hr | 3.7 | 1.1 | 627 |
| IBM 1302 | 2 | $2.34x10^{9}$ | 183 | .31 hr | 7.1 | .25 | 3,100 |
| IBM 2321 | 1 | $2.5x10^{9}$ | 72 | .81 hr | 8.0 | 5 | 60 |
| RCA 3488 | 1 | $2.04x10^{9}$** | 80 | .72 hr | 8.0 | 2.5 | 220 |
| GE Disc | 2 | $2 x 10^{9}$ | 220 | .26 hr | 3.8 | .04 | 19,900 |
| Honeywell Disc | ? | $2.4x10^{9}$ | 43 | 1.36 hr | 4.5 | ? | ? |

* Percent Growth = $\frac{\text{8 hours - Best Case Time}}{\text{Best Case Time}}$

** 7-bit characters

Figure 2. Summary Chart

- It is possible to preposition the heads so that the average access time is 40 ms as opposed to 92 ms average access time without prepositioning.

- The maximum access time required for an eight-hour load is 3.7 hours assuming no overlap and no prepositioning except in the case of "second"accesses. Assuming that the messages arrive so that prepositioning can be used to the utmost, the minimum access time for the load is 1.1 hours. The real system would operate somewhere between these extremes.

## 3.2 IBM 1302 Disc Model 2

- Two disc subsystems would be required to hold the data base. Two controllers would handle 10 disc units with a total of 2.34 billion characters of storage.

- The transfer rate is 184 kc.

- Since each disc unit has four access mechanisms, there can be a maximum of 40 simultaneous search operations.

- The average access time is 180 ms but simultaneous searching when possible reduces the effective access time.

- Allowing "second" accesses to take 70 ms, and first accesses to take the maximum 180 ms, 7.1 hours would be required to handle an eight-hour load. If it were possible to initiate the 40 simultaneous seeks continually, the entire request load could be answered in 15 minutes.

## 3.3 IBM 2321 Data Cell

- Five 2321 systems connected to one 7631 File Control will provide 2.5 billion characters of storage.

- The transfer rate is 72 kc.

- The storage medium is magnetic strips which can be mounted for reading and writing and then stored Depending upon how much strip action is required, the access time can vary between 50 and 600 ms.

- The worst case value is derived from sequential searching in which the correct strip is never mounted. This maximum value is eight hours; it can never be greater than eight hours, for if the backlog was that large, some natural batching would have occurred so that in some cases there would be more than one requested record per strip. Since there are five systems available in this configuration, the strip action can be overlapped to give an effective average access time of 120 ms. "Second" accesses would be roughly 67 ms since the correct strip would always be mounted. Maximum use of the overlapped searches would allow the entire load to be handled in five hours.

- Because the strip handling causes wear, there is a finite lifetime of each strip meaning that replacement and maintenance must be provided.

## 3.4 RCA Model 3488

- Three 3488 sixteen-magazine units under control of one 380 Channel will provide 2.04 billion characters of storage on magnetic cards.

- The transfer rate is 80 kc.

- Preselection of a second card can overlap the feed, spin, and return of the currently drum-mounted card. This selection process takes a constant 170 ms so that preselection can save a significant portion of access time.

- Access time varies between 30 and 465 ms depending on the amount of card manipulation required.

- Without preselection, accessing for an eight-hour request load would require eight hours. As discussed, accessing cannot be greater than eight hours without finding more than one desired record per card in some cases; this natural batching would be possible with a backlog. Making full use of preselection, the minimum access time for the load is 6.3 hours. The very best case is overlap searching in all three units simultaneously — this reduces the total access time to 2.5 hours. All "second" accesses would require 30 ms which is half the drum revolution time.

- Card handling will cause wear requiring maintenance and replacement. Sample failure times are given as 30,000 extractions of a card or 100,000 continuous revolutions while mounted on the drum.

## 3.5 General Electric Disc — DSU 250

- Eight disc units under direction of two controllers would be required for 2 billion characters of storage.
- Transfer rate is 220 kc.
- Each disc unit has 16 independent actuators which can be commanded to seek. Therefore, the system can have 128 concurrent searches.
- Average access time with no overlap is 90 ms. "Second" accesses will be less than half of this. Assuming no overlap searching, an eight-hour load will require 3.8 hours. At the other extreme, full use of overlap capability would allow the load to be handled in just a fraction of an hour.

## 3.6 Honeywell 1800 Disc

- Almost 2.5 billion characters of storage is contained in three units.
- Average access time is 110 ms. Therefore, without overlap of any kind the daily request load would take 4.5 hours of accessing. Detailed information was not available, but utilizing only the simultaneity of the three units operating in parallel, one hour would be needed.

## 4. File Organization

One of the most important features of any query system is the organization of the files. Since each query in the study system refers to a unique individual, the file is ordered by a serial number. This serial number is used as a primary address locator by revealing what disc, card, strip, or area should be searched. A table lookup in core could be performed on the leading digits of the number. There could perhaps be a secondary address locator on each disc, card, etc., which could be examined to find the exact location desired. A high-speed drum containing a complete address lookup could also be used as a locator. The Univac FASTRAND Drum has content search instructions which could be employed here to save reading an address locator into core and finding the desired address.

Storage capacities given in the preceding section are stated in 6-bit characters (RCA 3488 has 7-bit characters). It should be noted that the IBM equipment (1302 Disc and 2321 Data Cell) have an 8-bit mode also Double numeric digits in the data file can be stored in one 8-bit character as opposed to two 6-bit characters. Any numeric data coded in this way will occupy one-fourth less space. The data will have to be tagged to be recognized, but since this system is not processing limited, the extra processing will not be detrimental. This form of packing would only be advantageous if it could reduce the file size by an amount equal to or greater than a storage unit.

5. Batch Processing

If the constraint of real-time operation were relaxed, some of the devices described could be used much more effectively. If all the incoming requests were ordered and batched and stored until the end of the day, the processor would be free for the majority of the day to perform other tasks. For example, if the file was stored on the RCA 3488, one complete pass could be made by each card with an average of 4.7 requested records per card. This would reduce the entire accessing and processing time to about one hour. (Best case with continuous processing for this device is 2.5 hours.) There is additional time to be added to this hour — the time to sort or order the queries by serial number for the final processing. Queries could be stored on a drum in such a manner that sorting and merging would not be necessary as it would be if they were to be held on tape until final processing.

Cost can be reduced by batching because it would only be necessary to have one RCA 3488 unit instead of three. The magazines are interchangeable so that when all processing of the first block of 16 magazines is completed, the next block can be loaded in its place. Also the cards will get less wear by being accessed only once a day. Maintenance is thus reduced.

The previous also applies to the IBM Data Cell which has removable blocks of magnetic strips.

It must be kept in mind that with batching there is an automatic one-day delay (or whatever batching time is chosen) before answers are received. If requests are batched and disc or drum-type storage is being used, the accessing delay time will be much closer to that stated for best case for these devices (Figure 2) than could be achieved without batching.

The Summary File represents a compilation of statistics which is not often queried. The desirability of accessing this file for updating more than once per query is certainly questionable in terms of the low number of queries addressed to it. The updating of this file could be done once a day, or at the time it is queried. All changes would be saved, accumulated, and applied in one pass of the file. If this batching is not allowed, and it is determined that there are many Summary File accesses per query, it would perhaps be advisable to put this file on a high-speed drum. The Summary File is relatively small in size compared to the master file and it could be effectively updated in parallel with the query processing. This would of course reduce the access times given for the various systems by almost one-half.

## 6 Conclusions and Recommendations

The objective of this study was to show feasibility of building the sort of query system described. This has been demonstrated in terms of the variety of equipment configurations which would satisfy the system requirements. Economic feasibility remains to be studied.

Since the transfer time for 2,000 character records is much smaller than the access time in the study system, it would in general be impossible to utilize more than the minimum number of channels. No time will be saved by adding additional capability in this area.

It will be possible to interleave other processing tasks which do not require bulk storage access with the continuous query handling. This will increase the productive output of the system and therefore its efficiency.

Referring to Figure 2, the percentage growth listed for each device is calculated in terms of additional request load; it is then really a reflection of the number of bulk storage accesses which can be tolerated in an eight-hour period This growth capability shown on the figure relaxes some of the limitations of the assumptions made. For example, suppose it is determined that an extra access per query must be made in order to locate the record desired This will add to access time, but not enough to place any of the devices in a marginal performance area. Similarly, more accesses to the Summary File per query could be tolerated.

In order to make the final decisions of system configuration, cost must be applied and the other uses of the system must be defined. If the equipment is to serve no other needs, then the least expensive system which can possibly provide response to the query load expected should be chosen. If, however, there are other tasks to be performed, capability greater than needed just for the query handling will be necessary. In the extreme, if the query handling is to be of only secondary importance in the system, batching of requests should be considered. Since there is a wide range of equipment available, it will be possible to match closely the functions of the system.

# SECTION III

# STUDY OF AIR FORCE PERSONNEL ASSIGNMENT SYSTEM

## 1. Introduction

This section is devoted to the study of an Air Force personnel assignment system, the goals of which are to identify potential problem areas in a full-scale system and to suggest guidelines for system design which are valid for most large-scale computer systems. Concentration is focused on two areas — file organization and computational requirements.

The central organizational concept, on which the proposed automated system is based, is to have comprehensive career records of Air Force personnel[2] and codified job descriptions to which the individuals' qualifications may be matched. Updated man and job files would be continually available for the assignment exercise which, when approved, would be entered into the permanent files as ordinary updating. Because of the magnitude of the file size (1 billion characters) and the assignment load (6000 individuals per week), file organization is of central importance.

The value of a full-scale system will be measured in terms of the degree to which it provides proper assignments, the time to accomplish this, and the cost to perform the task. The present study has not been directed at the effectiveness of assignments. However, some assignment algorithms are discussed in paragraph 4 of this section.

## 2. System Design Requirements

The Air Force man file is composed of 130,000 officer records and 720,000 airman records. The job file contains 850,000 records; each describes a unique job, e.g., weather officer, flight surgeon, mechanic, etc. Referring to Figure 3, the job file is searched on a regular basis and all open job records are extracted. Then the man file is searched to identify personnel available to assume the open jobs. These available jobs and men are then matched by qualifications and a matrix is constructed which shows the value of each man to each job. An optimization technique is then applied to the matrix to obtain an assignment of men to jobs. Subsequent updates reflect the assignments.

The man-job matrix size is a function of the cycle time (time since a particular job category was last searched for openings and assigned new

[2] All personnel levels except E8, E9, Colonel, and General.

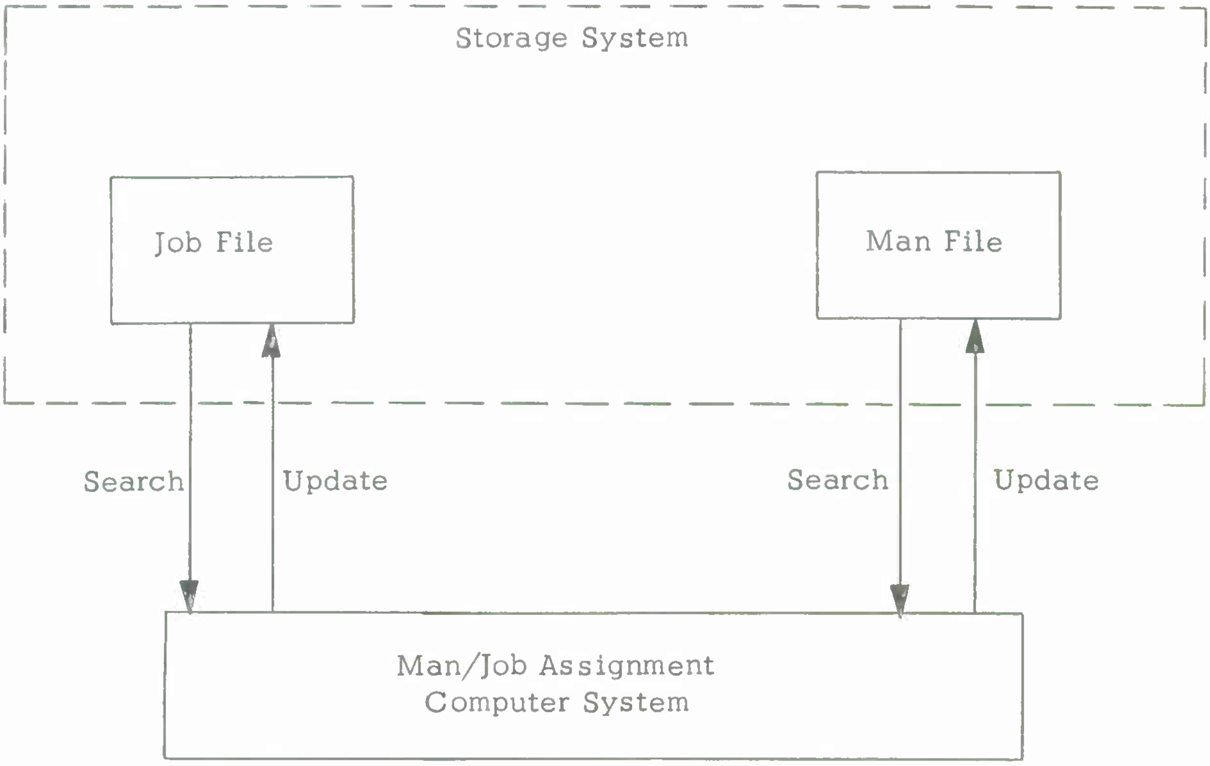


Figure 3. Recycling Available Men to Open Jobs

personnel). An individual's cycle time (time between new assignments) is assumed to be three years. Therefore, a third of all personnel in any job category are reassigned each year. Since personnel are not distributed evenly among the job categories, it will be necessary to recycle different job categories at different rates and with varying matrix sizes.

Processing required by an optimization procedure considered in a prototype system at ESD was to be approximately proportional to the cube of the matrix size. (See Decision Index Method, paragraph 4.) If the cycle time is long and the matrix large, the additional processing time may be prohibitive. However, the optimal matrix size with respect to processing time may result in a matrix so small that an individual is never considered for a large percentage of potential jobs.

Matrix size, processing time, and cycle time are three interrelated parameters which can be adjusted within limits to influence system performance. These relationships exist independent of the configuration of the system and therefore do not have to be defined in the early design stages.

## 3. System Configuration

Four alternative configurations are presented with their relative advantages. There exists another Air Force system which contains a master file of all personnel data with on-line updating. It was suggested by ESD that the system under study communicate with the master file system. The degree to which this supplied link is utilized represents the difference between the configurations described below.

The responsibility of updating the job file falls to the man/job system in all configurations, but it is assumed that this file is altered infrequently. One entry, social security number[3] of incumbent, intended to assure that one is not recycled to his own job, could be updated when the entire job category assignment has been approved. All changes can be made to the job category tape file at one time.

### 3.1 Configuration A

At one extreme of the scale suggested above lies the case where the man/job assignments are executed by a computer system which has no other tasks and which is self-sufficient in the sense that no data or programs are shared with any other system. It relies on the master file system only for the transmission of updating information, as seen in Figure 4. Whenever

[3] This assumes that all personnel have applied for and possess social security numbers.

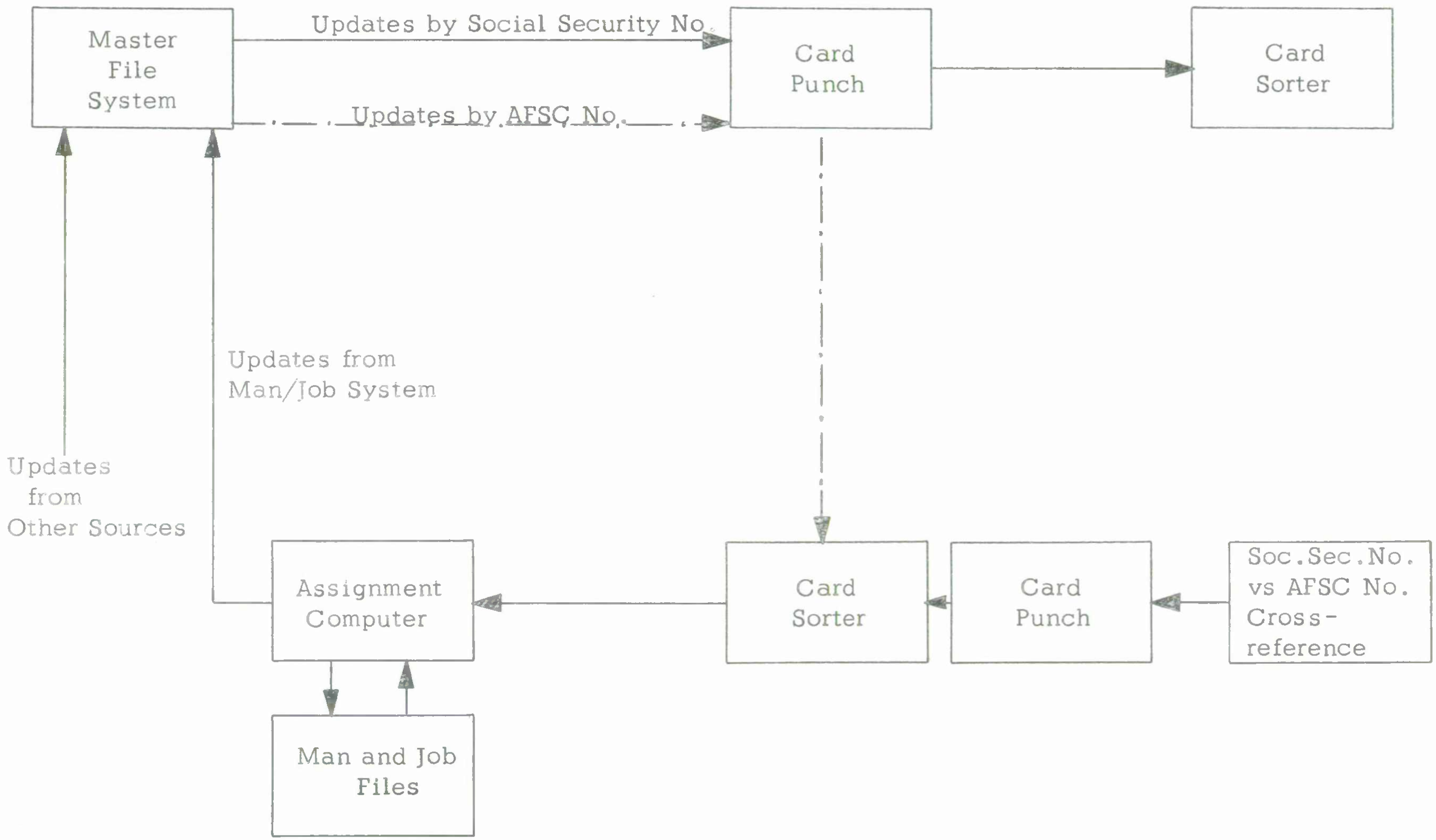


Figure 4. Configuration A

the master file system receives a file change, it is formatted and sent to the man/job system for incorporation into its files. The man/job system has no real-time requirements; it is only necessary that files be currently correct when they are being searched.

Since the files would not be shared, it would be feasible to organize them in the most advantageous manner with regard to the assignment problem. Different cycles or job categories are completely independent so that each one could act as a separate file. If the files were ordered according to job category number, it would be possible to deal only with relatively small parcels of information, and the total file size need be considered only for storage, not for processing. Separation of the file into categories would also facilitate additions and deletions. Further search efficiency would result from organizing the man file by social security number, or by ordering both man and job files by availability dates (e.g., vacancy date). This would minimize the number of man/job records which would have to be examined in the execution of a particular assignment task.

If the assignment matrix were limited to 100 jobs (men) at one time, there would have to be at least 60 assignment runs per week to accommodate the 6000 weekly assignments. The entire data base would occupy about 60 tapes, but if the file is structured, only appropriate ones would have to be mounted at one time. It would be advantageous to use more than the required number of tapes for storage so that a given category file could be found more quickly and less extraneous data would have to be read in searching for the appropriate file. This measure would also decrease the amount of information to be copied when writing a new tape with corrections.

The major obstacle encountered in this configuration is updating the data base. The update changes will have to be accumulated and incorporated into the data file before it can be searched for a reassignment cycle. If each individual's record is changed once a month, there will be approximately 42,000 updates per weekday. Assuming that the master file system is supplying them over one standard telephone line (2 kc character transfer rate), and assuming that the average update length is 80 characters long, it will require six hours of transmitting to convey the daily updates to the man/job system. An additional channel exclusively for this update link may be needed by the master file system. The master file is organized by social security number, whereas the man/job system's files are ordered by job category number, and perhaps further by availability date. Therefore, all incoming updates must be cross referenced and sorted before they can be incorporated into the tape files. The most efficient way of sorting such a vast amount of information is by sorting cards off-line as discussed below. The update message could be punched on a card as it arrives and then the cards could be sorted by social security number. These cards would then be matched against a cross-reference file of 15 million characters to obtain the proper category location for each one. Then new cards would be punched with the new information and sorted by category; they would then be merged with the appropriate tape files.

The scheme just mentioned has operational drawbacks; the double set of cards would number 84,000 per day. The sorting alone would require 25 hours a day necessitating the use of full shifts of two parallel sorting machines and personnel. This off-line processing load can be halved by requiring the master file system to send all update information coded by job category number (dotted path in Figure 4). In all probability this will require an extension to the personnel files in the master file system. The master file records are each just under 2,000 characters so that the addition of a 9-digit number (if not currently in each individual's record) does not constitute a large addition. If this arrangement could be made, the incoming updates could be punched on cards to be sorted by job category number.

It would be inefficient to update files which were not being examined if for no other reason than that the more than 60 tapes holding the file would have to be mounted and rewritten with the update information. An alternative is to save all cards pertaining to a job category until it is due to be recycled so that all accumulated updates could be incorporated in one pass of the tape. However, aside from the obvious card storage problem, there would be no way of establishing historical significance of any of the update information.

The configuration described is unwieldy because of its dependence on operating personnel. There are several kinds of errors to which this system will be prone. Sorting mistakes, where a card is out of order or where a card has been sorted into the wrong category, can usually be corrected manually by simply examining the rejected card. The errors which are more difficult to deal with are garbling and loss. In these cases it is impossible to recover the correct update from the card, and it will be necessary to have some other method of keeping track of the updates. One solution would be to write a transaction tape as the incoming update cards were being punched; this could be used for recovery if the file updating was performed frequently. Otherwise it would be impossible to know which tape would have the correct update being searched for.

If the master file system had access to each person's job category number, the master file could be requested to send an entirely new data base at regular intervals. This would result in a continually evolving data base. It would require about 240 hours of transmitting to convey the entire man file (2000 characters each) from the master file system to the man/job system.

### 3.2 Configuration B

An alternative design is the same as shown in Figure 4 except that the billion-character man and job files are stored on a large disc rather than on tape. The master file system would be utilized in the same manner as in Configuration A — it would supply updating information. Sorting personnel and machines would no longer be necessary, but the computer system would be considerably more complex.

If updating information were sent coded by job category, the disc file could be immediately corrected. However, if the master file did not contain this information, the updates could be sent by social security number and written on tape. A job category number-social security number cross-reference file could be held on the disc and accessed to determine which records should be changed in the file. Based on an average disc access time of 100 msec., it would require 1.16 hours a day to consult the cross-reference file; this is the additional updating time needed if the job category number does not accompany an update.

It has only been recently that random-access storage of the magnitude required by the man/job file has been available. Storage equipment of this size requires fairly expensive, fast, and large central processors.[4] Incremental additions in storage are much less expensive per character than the original bulk needed. For this reason there is no advantage to having the cross-reference file, if necessary, on tape — it should be kept on the disc so that the random-access feature can be used. If this configuration is implemented, the system will have more capability than is necessary to perform the man/job assignments. Unless there is other processing which could be done on this system, the efficiency of this configuration may be very low.

## 3.3 Configuration C

Shifting more of the responsibility onto the master file system would allow the man/job system to concern itself chiefly with processing the assignment matrix. In this case, the man/job system would function as a remote console communicating with the master file system for data but performing its own processing. In review, the master file system has a 2000-character record for each member of the Air Force ordered by social security number and the capability for on-line updating of its own file. The master file would be relied upon to supply individual man file records on demand as each category is recycled as shown in Figure 5. A cross-reference file could be constructed which would be ordered by job category number. This tape would be searched for the appropriate category, and the successive social security numbers of personnel in this job category would be sent to the master file system for direct accessing from its storage. The individual records would then be transmitted to the man/job system where they would

[4] See Section II.3 for a survey of available storage equipment in the size range required.

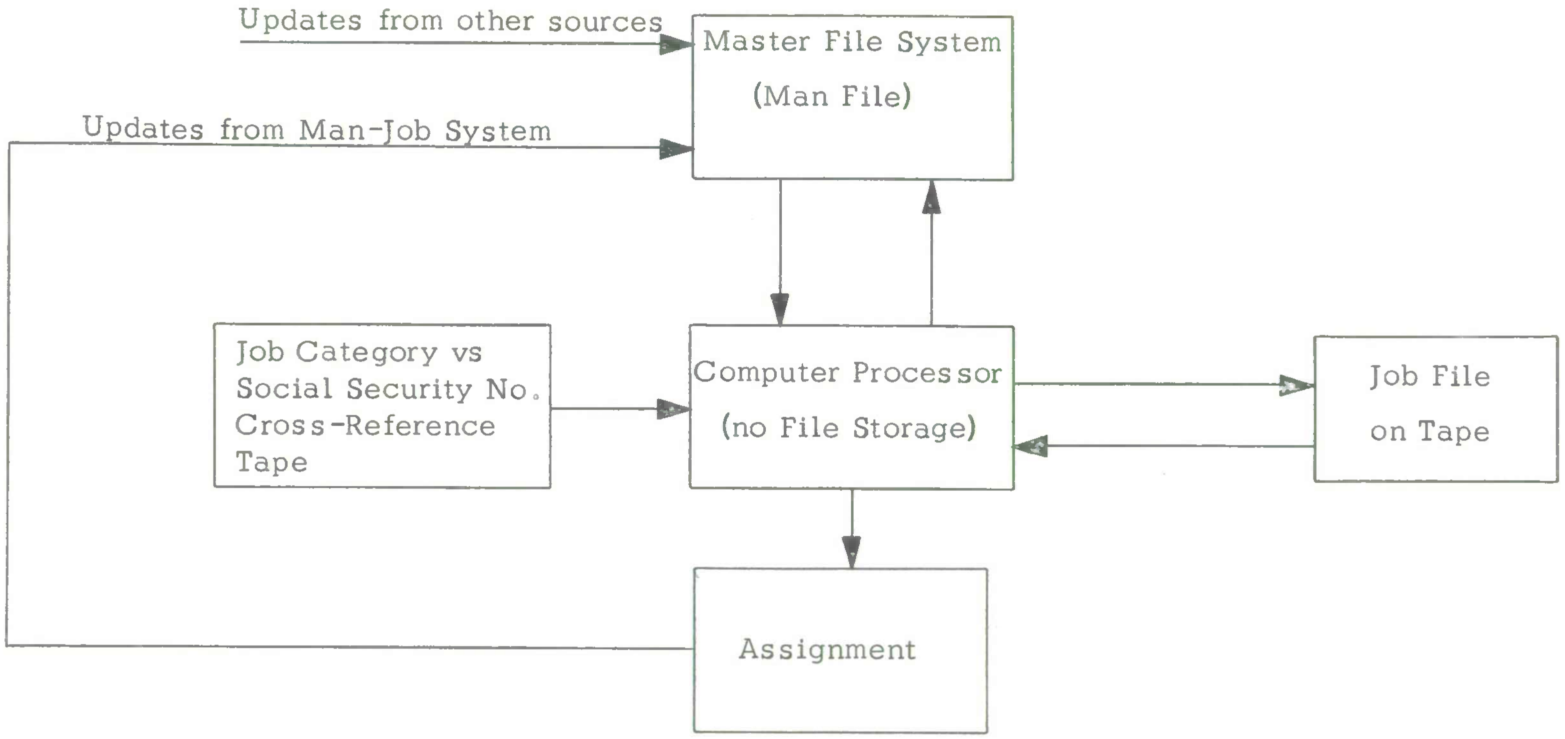


Figure 5. Configuration C

be written on tape. This subfile tape would then be used for processing of the assignment matrix. As in ANY configuration, the approved assignments must be sent back to the master file system as a normal update.

An advantage of this configuration is that the man/job system is relieved of all update processing. If, as is suggested here, the master file system were to transmit only the relevant man file records over one standard (2 kc character transfer rate) telephone link, an entire week's assignment load (6000 records of 2000 characters each) could be transmitted in one and two-thirds hours. With the removal of update loads the need for sorting machines and personnel vanishes.

A possible drawback is that the master file man record will probably have to be extended in order to cover all information required for the matching of qualifications for assignment. The prototype man/job system contains 867-character man records; some of these characters will definitely be redundant, but not all.

### 3.4 Configuration D

The logical extreme of the utilization of the postulated master file system is to allot the entire assignment task to it. Incremental storage additions to its 2 billion-character file will be relatively inexpensive per character and the tasks presently envisioned for the system[5] allow assumption of additional processing loads. This configuration makes the man/job system look like a remote console with only input/output capability.

## 4. Assignment Algorithm

The assignment problem can be stated as follows: given n people and n jobs available for assignment, determine the "best assignment" such that only one person is assigned to one job. "Best assignment" refers to some maximization of the payoff values assigned to each man/job combination. Three such optimization techniques were investigated briefly and are listed below.

The Decision Index Method[11][6] consists of successive modifications of a man/job payoff matrix. The first step is to compute the Decision Index for each element of the matrix and then select the element with the highest Decision Index as the first assignment. That row and column are then deleted from the original value matrix and a new set of indexes is calculated. The highest index is again chosen as the assignment. The procedure is continued

[5] See Section II.3.

[6] Numbers in brackets designate references included at the end of this report.

until all personnel or jobs have been assigned. Personnel assignment resulting from this technique is not optimal; however, the logic and computations are relatively simple. To obtain a timing estimate, a 100 x 100 matrix is chosen as an example. The first assignment requires $n^2 = 10,000$ Decision Index computations and 10,000 comparisons to select the maximum value. To compute the next assignment there are $(n - 1)^2$ computations and comparisons. The total number is the sum of the squares = $n^2 + (n-1)^2 + \ldots + 3^2 + 2^2$ which is asymptotic to $n^3/3$ for large n. For the 100 x 100 example, there are 376,699 computations and an equal number of comparisons to be made. Depending on the speed of the processor chosen, time can be estimated; conservatively estimating 1 msec for each operation, the entire matrix could be solved in about 12 minutes. This technique requires a working storage of $2n^2$ locations (20,000 for 100 x 100 size matrix).

The Hungarian Method[12] is a way of manipulating a value matrix to yield a solution which is optimal in the sense that the payoff of the assignment matrix is truly maximized. Because the calculations are simple and there are fewer of them than in the previous example, it is assumed that this technique would require less processing time.

The personnel assignment problem can also be formulated as a classical linear programming problem — the transportation problem.[13] It has been studied extensively and optimal solutions exist.

## 5. Design Alternatives

The configuration of the proposed man/job matching system must be viewed within the context of shared responsibilities with the master file. The respective delegation of duties hinges on the degree to which the study system is balanced in size and speed to the master file system. If, at one extreme, the man/job system is designed to have its own data base storage, handle its own processing, and rely on the master file system only for updating information, it becomes an independently scheduled and autonomous computer system. At the other end of the spectrum, the man/job system is a console; in this case its limited capability should be fully utilized in performing the assignments and answering queries, or assuming any other new functions. Since it is so dependent on the master file system for data and updating, the console should do all the processing possible so as to be a minimum burden to the larger system. In other words, the more balanced the two systems are, the more the total load should be distributed. If the systems have vastly different capabilities, the smaller one should be fully utilized, leaving the larger system more flexible.

The characteristics to be considered in choosing a system configuration have both quantitative and qualitative trade-offs. The salient features are total data storage and transmission equipment required, resource utilization, and cost. Attention must be paid to the capacity for expansion; often growth is not in predictable directions. The original designers of the master file system discussed here may not have envisioned the man/job match function which the system will serve. A rigid configuration like A allows very little expansion in size and almost no alteration in the role of the system. On the other hand, if the man/job system looks more like a console, time-sharing its tasks with a larger system, there are immediate advantages in flexibility and the ability to adapt the system to new demands. For example, there may be future need for querying of the master file system via the consoles. As another example, the console might serve as a statistics gathering center.

## 6. Recommendations

Referring to the Summary Chart (Figure 6), Configuration D or C is recommended because of its capacity for growth in any direction, its freedom from the update load and its independent operational procedures. This design also represents more efficient use of resources in that the man files do not have to be maintained redundantly, and the additional updating traffic is avoided.

| Configuration | A | B | C | D |
|---|---|---|---|---|
| Man/job system file storage required | 60 tapes (Job file Man file, cross reference, if needed.) | One billion character disc | 5 tapes (Job file tapes—working tape for man file, cross reference tape) | None |
| Data transfer time from master file to man/job system | > 6 hours per day in updating | > 6 hours per day in updating | 1-2/3 hours per week to send requested man file records | Query and response —transmission time |
| Man/job system equipment required | > 3 tape drives — card punch —2 sorters—telephone line to master file simple computer | One billion character disc with accompanying computer—telephone line to master file | 3 tape drives — telephone line to master file — simple computer | Remote console for querying — telephone line to master file |
| Man/job system degree of manual operation | 2 full shifts/day of sorters (prone to errors) | Minimum | Tape mounting required | None |
| Man/job system use | Maximum | More capability than required for man/job task, ∴ may be low utilization | Computer could be chosen to perform only man/job task ∴ high utilization | Master file system would have greater use |
| Update Processing load on the man/job system | Responsible for incorporating all updates sent from Master File | Updates made to disc files upon arrival from Master File | None | None |

Figure 6. Summary Chart*

* Based on 3-year duty cycles, and one update/month/man.

# SECTION IV

# STUDY OF A TIME-SHARED PERSONNEL SYSTEM

## 1. Introduction

This section will describe an analytical study of some of the performance and design characteristics of an on-line, time-shared personnel system which could be implemented using presently available equipment.

The original intent of the study was to compare the performance and cost of two general types of implementation of such a system. Both types were characterized by a centralized data storage facility which could be accessed and modified by a large number of geographically dispersed users who communicate with the system in an on-line fashion. The first type of system design provided for centralized computation as well as data storage, with no computational capability at the remote terminals; the second type provided for centralized data storage and retrieval, but computational capability to operate on retrieved data was dispersed to the remote sites.

As the performance data was determined for the applications in question, the clear superiority of the first type of system, that of centralized computation as well as data storage, was made quite evident and consequently emphasis was placed on this type. Paragraph 3 of this section will discuss the reasoning to justify this decision.

## 2. Centralized Computation System

The characteristics of the hypothetical system studied with this model are not peculiar to personnel management systems, but are applicable to many other types of command and control functions. Principally, these characteristics are: multiple simultaneous users; large, structured, random-access data base (billions of characters); high update load; priority schemes; and low proportion of computational load compared with data base maintenance load. The bulk data storage is assumed to be provided by disc or drum units.

### 2.1 Assumed System Functions

As applied to the personnel management function, the system would provide a centralized facility to serve both personnel and payroll users. The functions of the system can be categorized broadly as: (1) Man/Job assignments; (2) Updates to Centralized Data Base; and (3) Queries of Data Base. The man/job assignments involve many separate steps. First, the personnel and job files must be searched for available men and jobs. These will be ordered in matrix form, to which optimization methods can be applied. This last step represents the only significant computational load considered for the system, and it will be assumed that such computations will be performed as a "background" task for the processor while it is waiting for inputs from consoles or for data to be retrieved from the bulk storage unit. Updates require no significant computer processing, but do require two accesses of the bulk storage to find and correct the update item. Queries represent multiple retrievals from the disc unit, with a moderate amount of computer processing applied to each retrieval, using Boolean comparisons of the retrieved records with the request formulation.

### 2.2 Description of Model Studied

A functional model of the centralized system is shown in Figure 7. The shapes of the functional blocks correspond to the conventions of the GE System Modeling and Simulation Technique.[7] Circular shapes represent originations in the data flow; oval shapes represent temporary storage or buffering; rectangular shapes represent processing facilities, with consequent delays; fan-in and fan-out shapes represent merging and routing functions; and trapezoidal shapes represent terminations for data flow.

### 2.3 Performance Characteristics to be Estimated

The performance characteristics of interest are as follows:

- Average response time for normal users and priority users;
- Statistics of utilization factors for central processor and disc access channels; and
- Statistics of queue lengths at buffers.

### 2.4 System Design and Load Factors

Values for the above performance characteristics were estimated by performing simulation runs on the model, with the following as design constants, or variable parameters:

- Arrival Load for Queries, treating all users as a single composite input load.

[7] For a description of this technique, see ESD TDR-63-612.

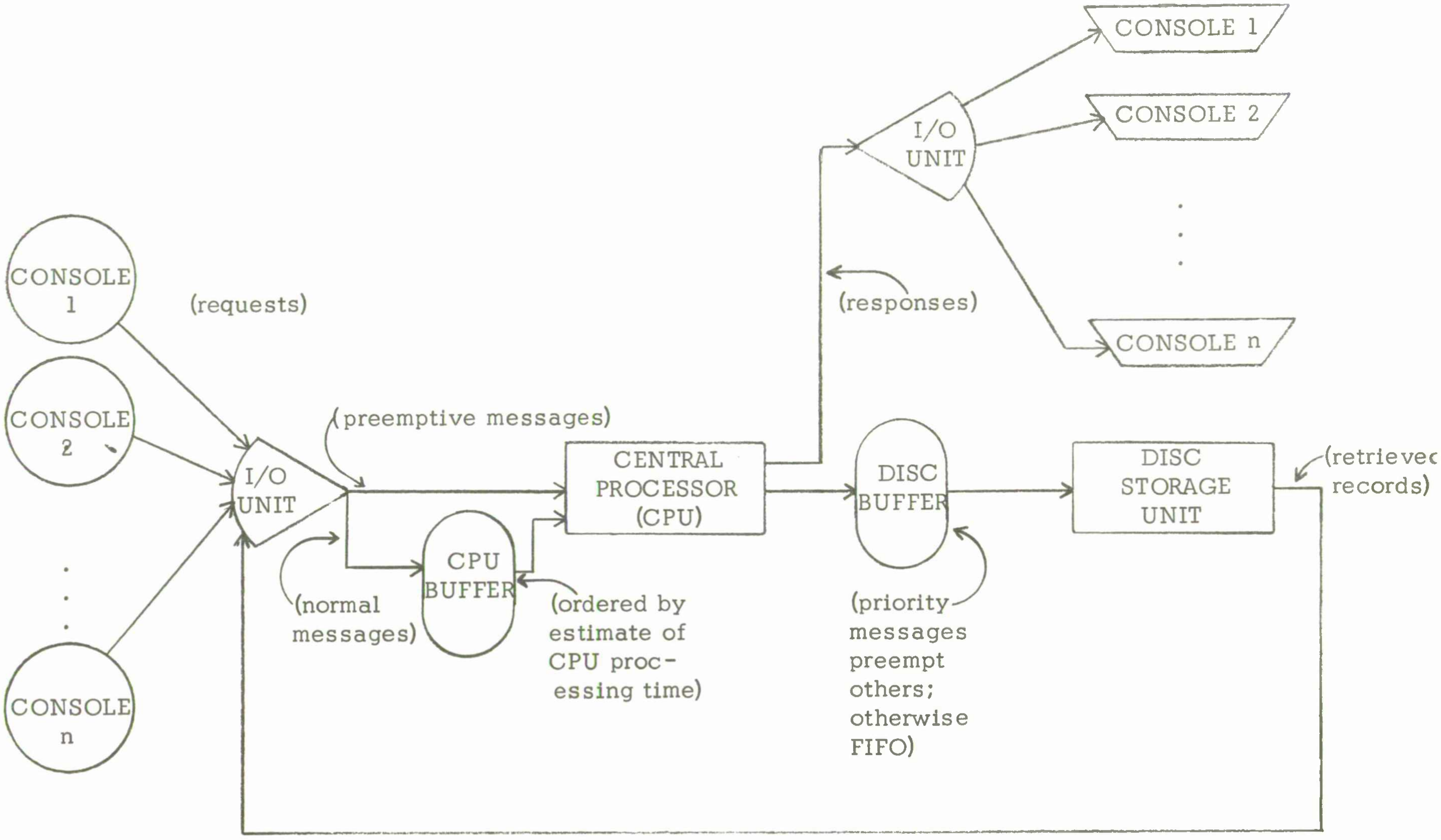


Figure 7. GPSS Flow Chart

- Percentage of Query Types: Retrievals, Updates, and Computation Requests[8]

- Number of disc accesses required to satisfy retrieval queries (influences the number of computer/disc interchanges necessary)

- Amount of data transferred from disc to CPU

- Number of access channels between CPU and disc

- Number of search arms on disc (determines number of simultaneous accesses)

- CPU processing time for each query type

- Disc search time per access

- Disc to CPU transfer time as function of data block length

## 2.5 Assumptions on System Design

In performing the simulation study, a hypothetical design was assumed having the following characteristics:

- Individual, non-shared communication channels between consoles and CPU;

- No delays in transfer of query to CPU;

- Queues for computer service are ordered FIFO; but

- Retrieval-type queries are allowed to interrupt service on a computation;

- When CPU traps to an interrupt, results of the computations in process are not lost.

- All data and programs are stored randomly on a random-access disc.

[8] In the actual simulation, only retrievals and updates were considered as query types; the capability for computations can be estimated by examining the available CPU time not used for these types, and considering that background computation can be carried out during these intervals.

- Priority schemes not based on the originator of the query, except for the special takeover type, in which the extraordinary priority might be assigned through the importance of the requestor.

In addition, for purposes of the simulation, the following computational assumptions were made:

- Interval between arrivals of queries from consoles is exponentially distributed;
- Delay at the disc while transferring data to the CPU in response to retrieval requests is uniformly distributed;
- Time to transfer records in response to update requests is constant;
- Number of accesses of disc to satisfy retrieval requests is uniformly distributed with mean of 7, and required CPU processing time resulting from each access is uniformly distributed with a mean of 10,000 μsec.
- Times for executive housekeeping and initiation of I/O are included in CPU processing time distribution for each loop.
- No errors or failures take place.
- Updates require only two disc accesses, and essentially no processing time.

## 2.6 GPSS Simulation

The model described in paragraph 2.2 of this section was rendered into the form required for computer simulation by the GPSS III (General Purpose System Simulator) program for the IBM 7094. The rendition is organized around the sequence of steps given in Figure 8 and results in the flow chart of Figure 9. The computer input listing corresponding to this flow chart is given as Figure 10.

## 2.7 Simulation Results

A limited number of simulation runs were made, to estimate performance for a nominal system, and determine instances under which saturation would result. The results from three runs are of most interest. These runs are summarized as follows:

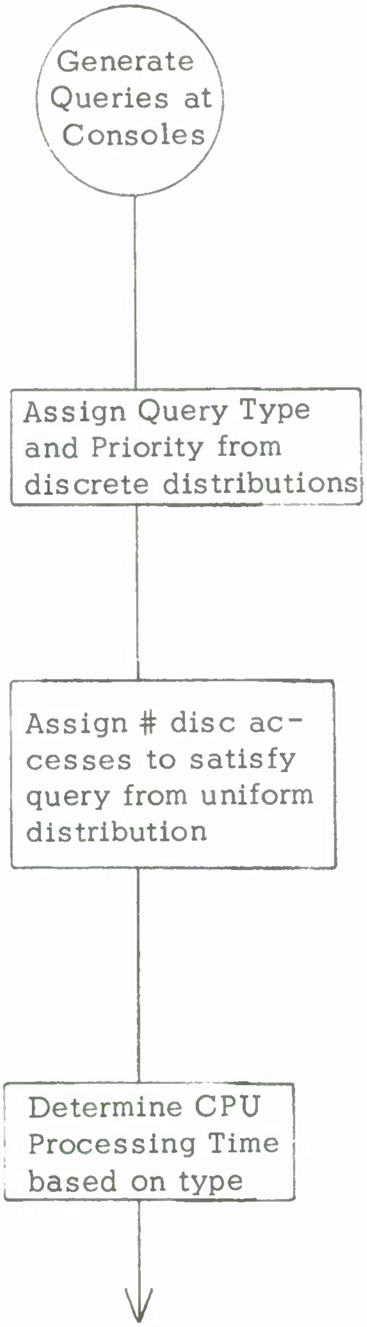
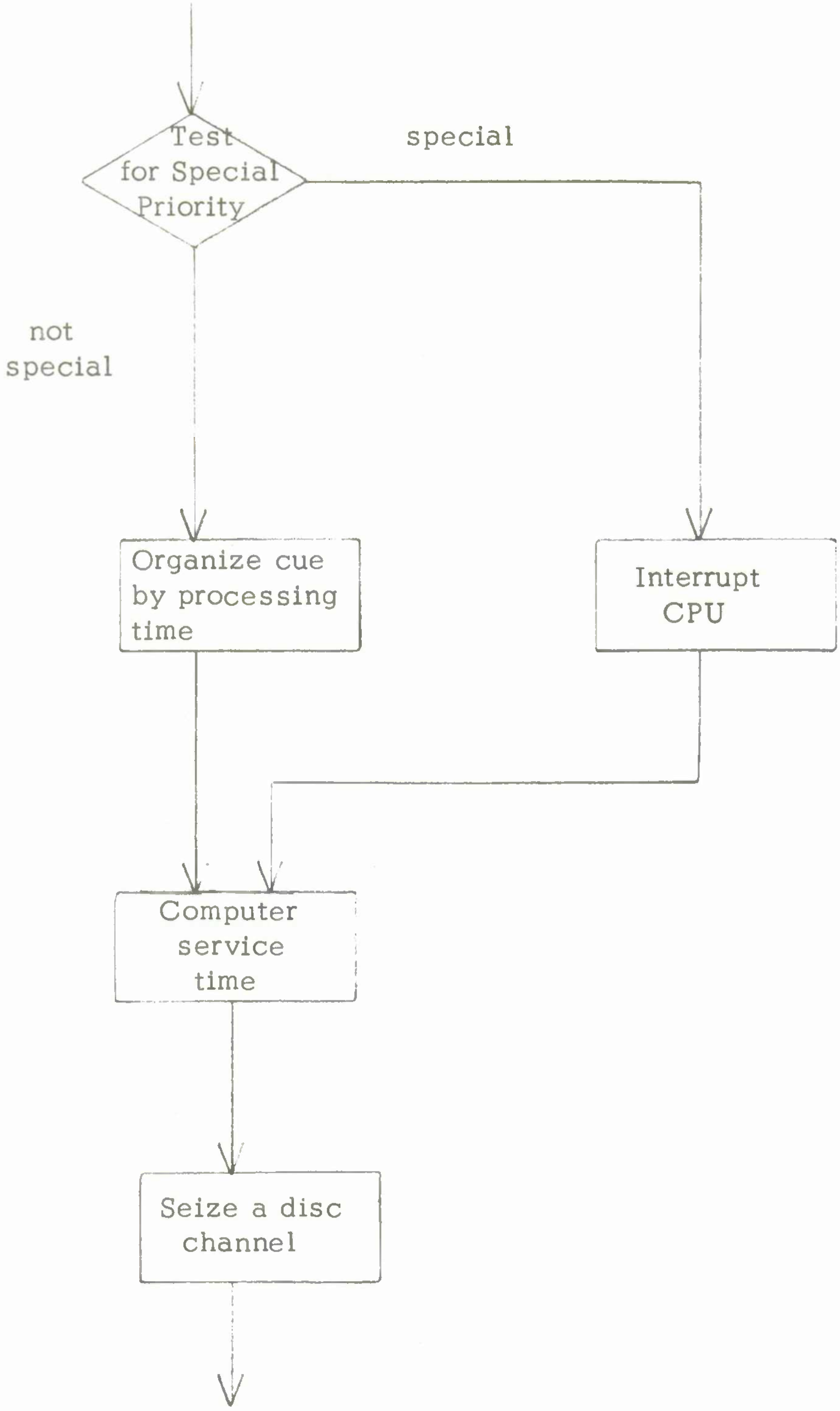

Generate Queries at Consoles
Assign Query Type and Priority from discrete distributions
Assign # disc accesses to satisfy query from uniform distribution
Determine CPU Processing Time based on type

Figure 8. Simulation Flow


Test for Special Priority
special
not special
Organize cue by processing time
Interrupt CPU
Computer service time
Seize a disc channel

Continued

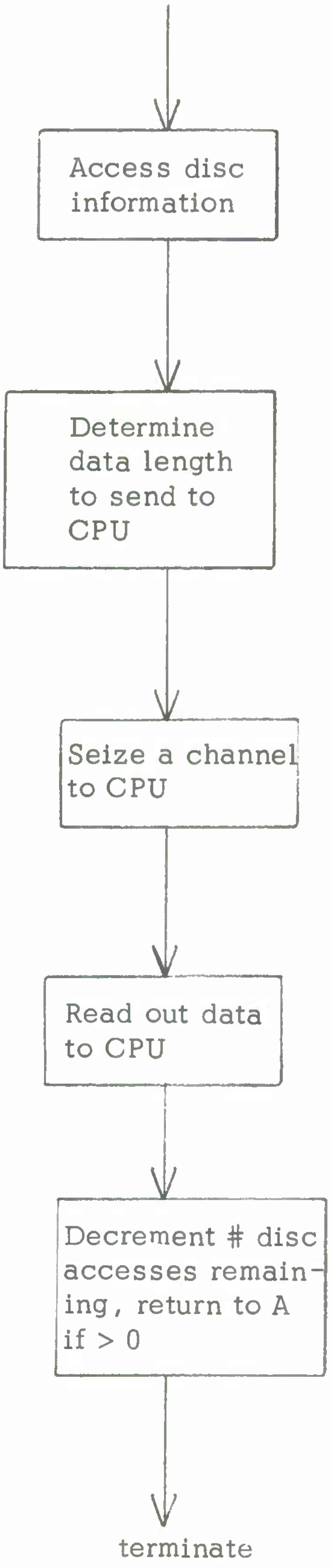

Access disc information
Determine data length to send to CPU
Seize a channel to CPU
Read out data to CPU
Decrement # disc accesses remaining, return to A if > 0
terminate

Figure 8. Concluded

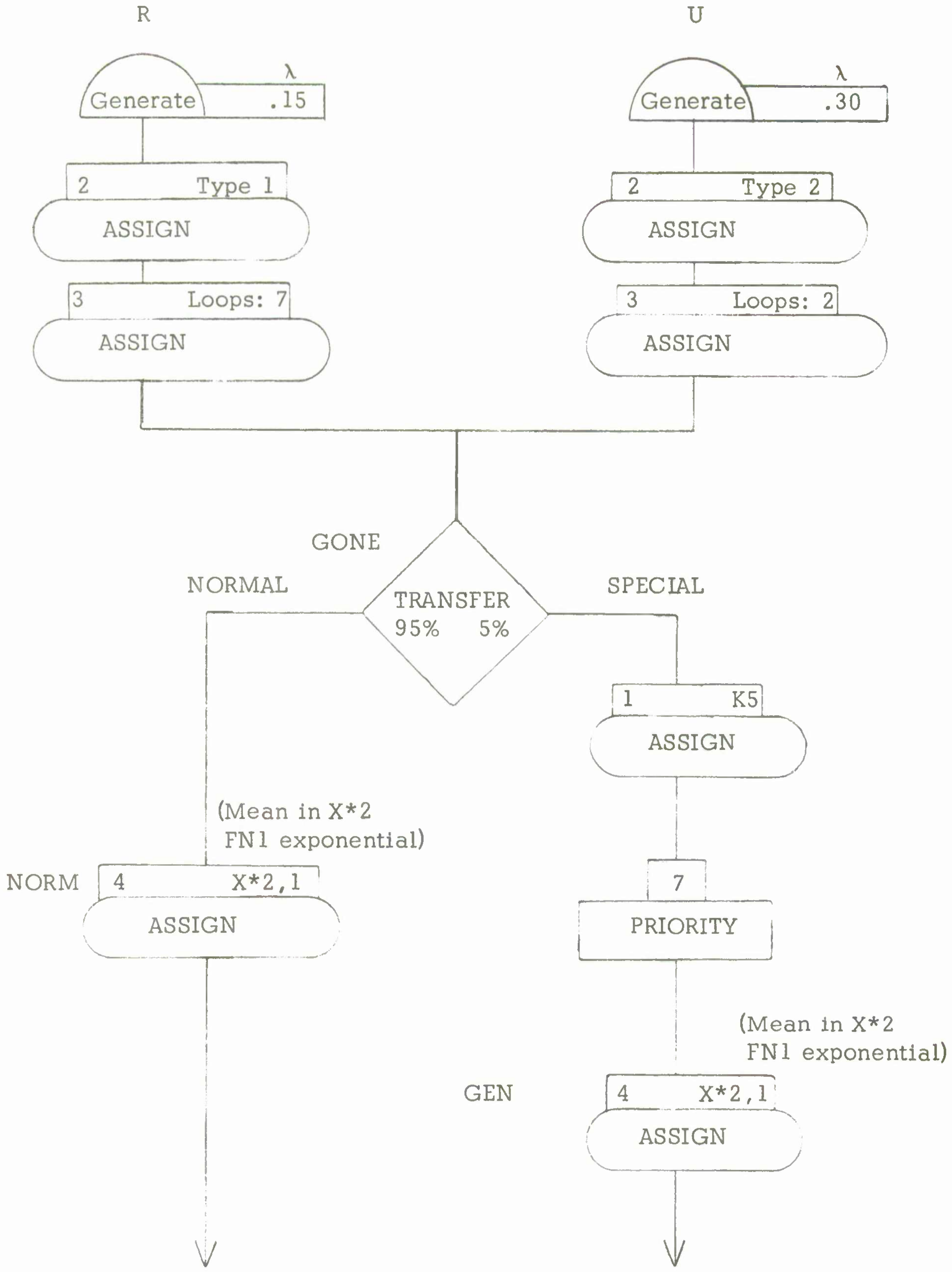
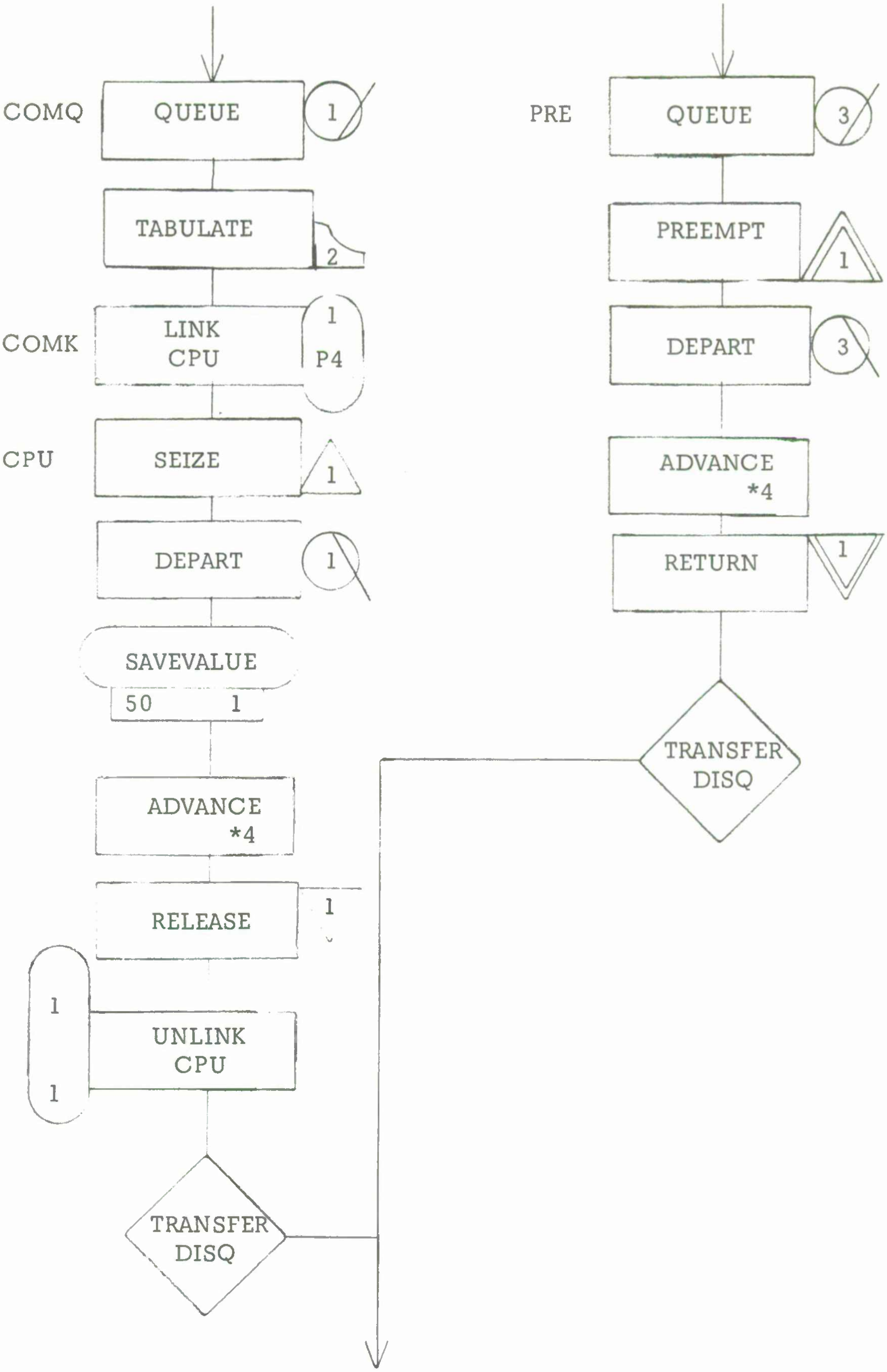
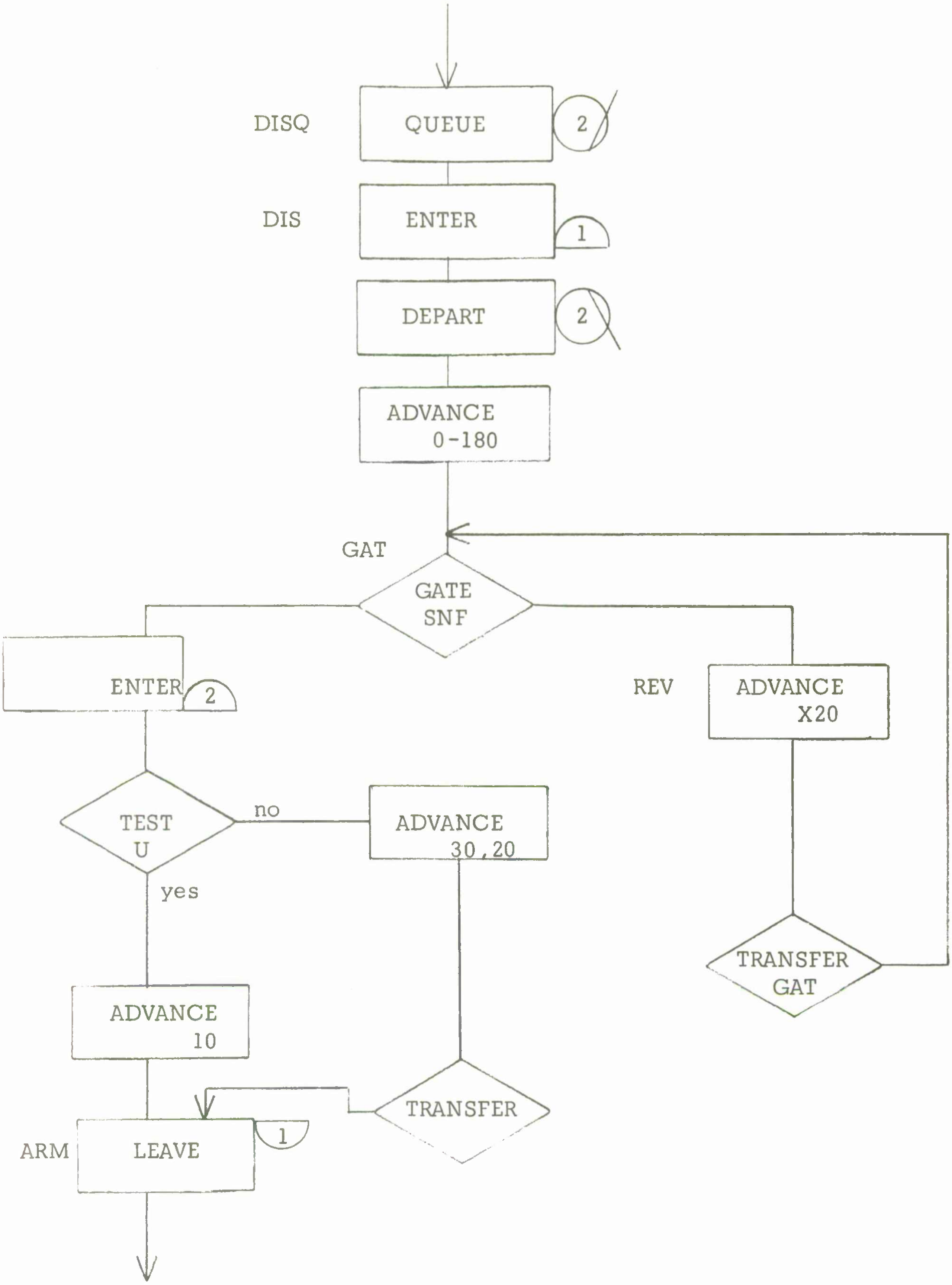


Figure 9. GPSS III Flow Chart


COMQ
QUEUE
1
TABULATE
2
COMK
LINK
CPU
1
P4
CPU
SEIZE
1
DEPART
1
SAVEVALUE
50
1
ADVANCE
*4
RELEASE
1
1
UNLINK
CPU
1
TRANSFER
DISQ
PRE
QUEUE
3
PREEMPT
1
DEPART
3
ADVANCE
*4
RETURN
1
TRANSFER
DISQ

Continued


DISQ
QUEUE
2
DIS
ENTER
1
DEPART
2
ADVANCE
0-180
GAT
GATE
SNF
ENTER
2
REV
ADVANCE
X20
TEST
U
no
ADVANCE
30,20
yes
TRANSFER
GAT
ADVANCE
10
TRANSFER
ARM
LEAVE
1

Continued

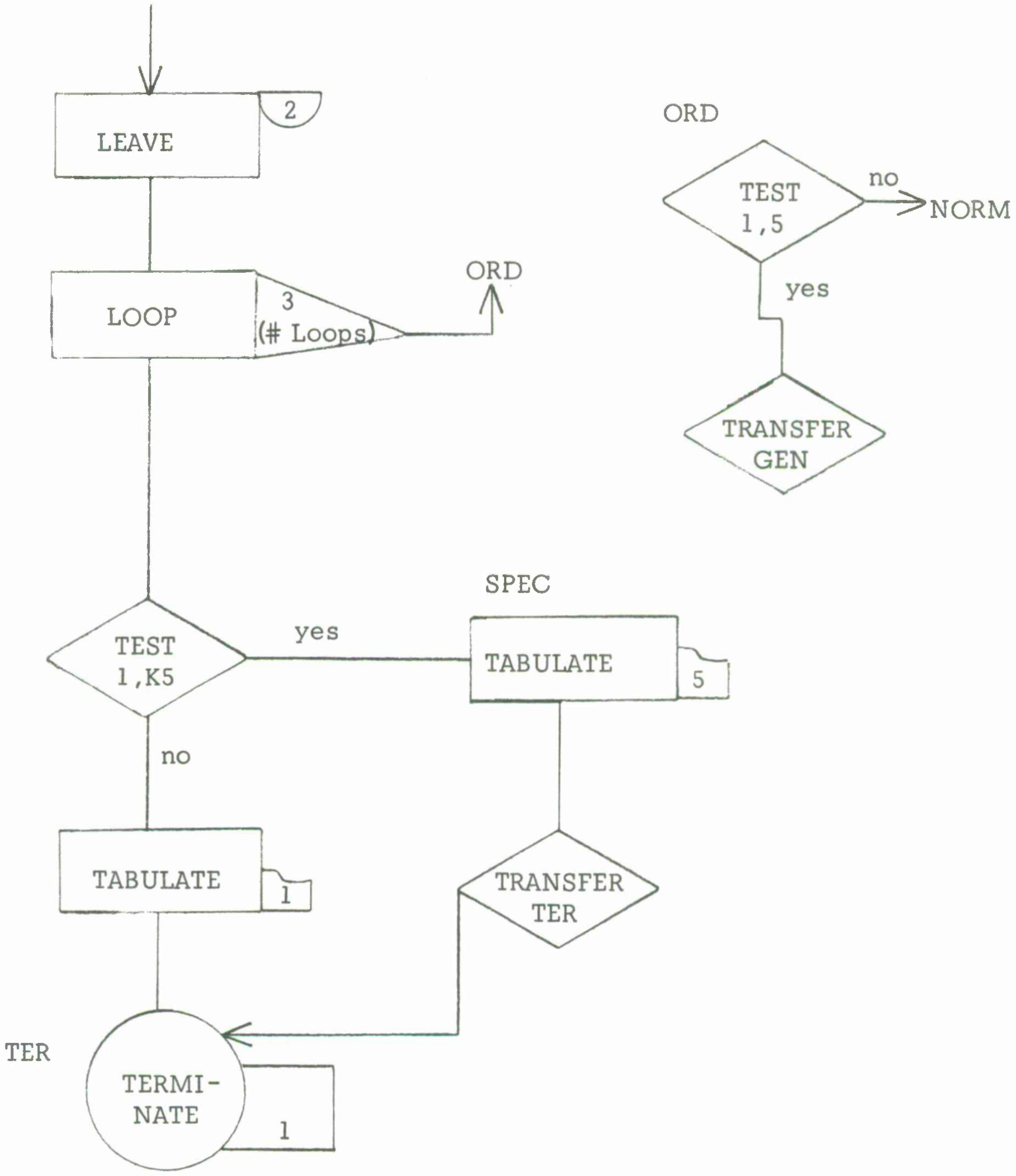
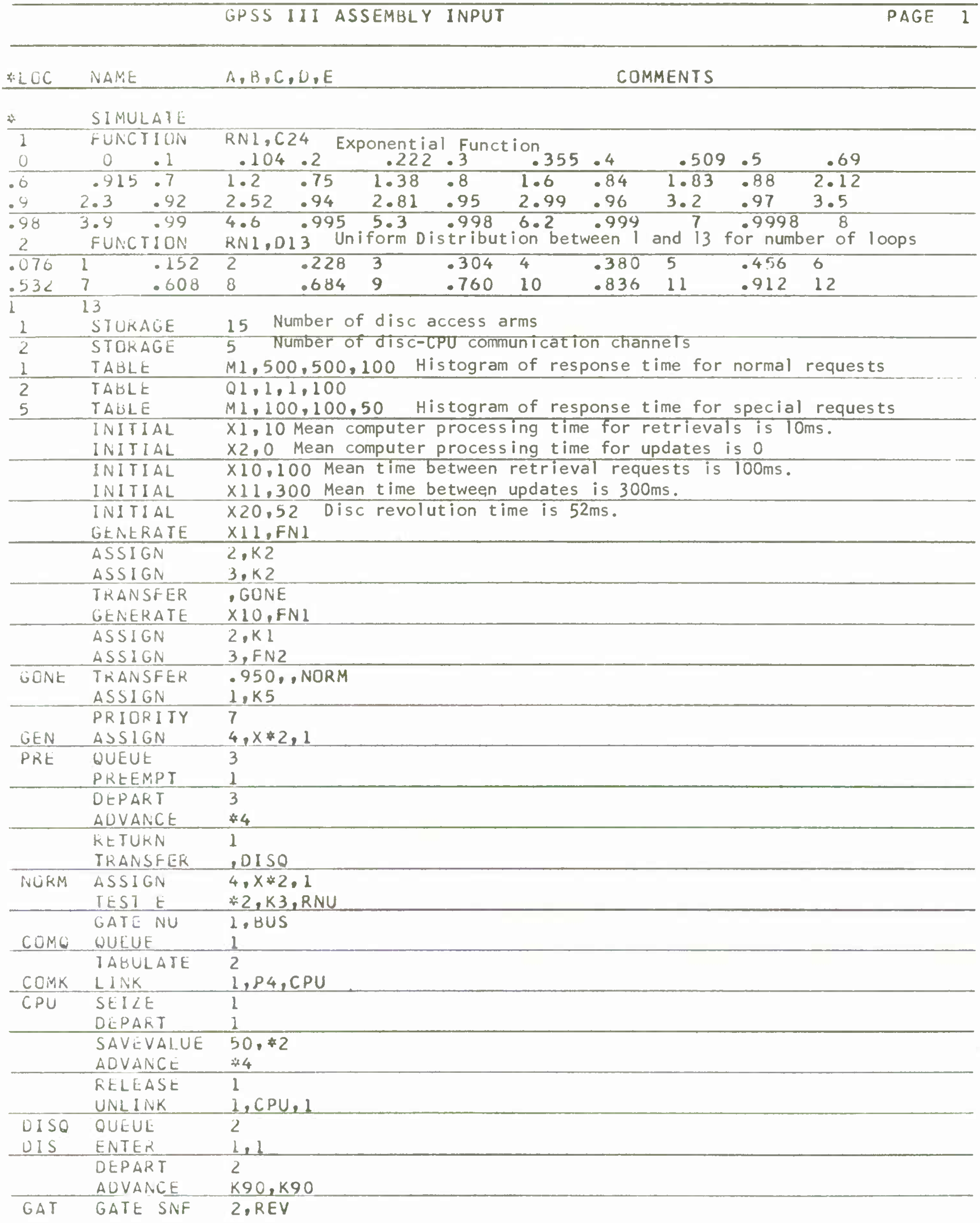


Figure 9. Concluded

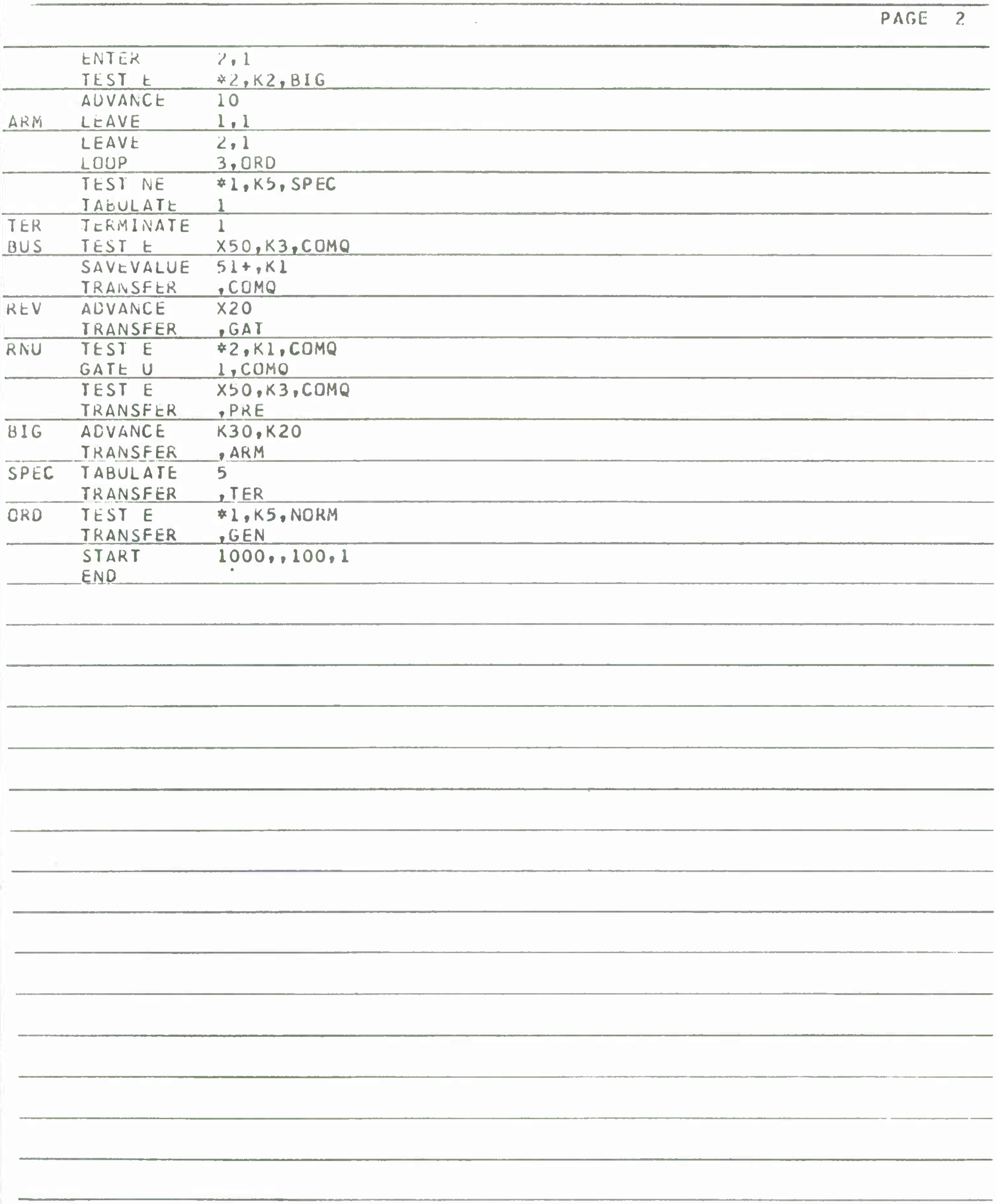


```
                    GPSS III ASSEMBLY INPUT                                      PAGE  1

*LOC    NAME        A,B,C,D,E                            COMMENTS

*       SIMULATE
 1      FUNCTION    RN1,C24  Exponential Function
 0       0    .1     .104 .2      .222 .3      .355 .4      .509 .5      .69
.6      .915  .7    1.2   .75   1.38   .8    1.6   .84   1.83  .88   2.12
.9     2.3    .92   2.52  .94   2.81   .95   2.99  .96   3.2   .97   3.5
.98    3.9    .99   4.6   .995  5.3    .998  6.2   .999    7   .9998   8
 2      FUNCTION    RN1,D13  Uniform Distribution between 1 and 13 for number of loops
.076   1      .152  2     .228  3      .304  4     .380  5     .456  6
.532   7      .608  8     .684  9      .760  10    .836  11    .912  12
1      13
 1      STORAGE     15   Number of disc access arms
 2      STORAGE     5    Number of disc-CPU communication channels
 1      TABLE       M1,500,500,100  Histogram of response time for normal requests
 2      TABLE       Q1,1,1,100
 5      TABLE       M1,100,100,50   Histogram of response time for special requests
        INITIAL     X1,10 Mean computer processing time for retrievals is 10ms.
        INITIAL     X2,0  Mean computer processing time for updates is 0
        INITIAL     X10,100 Mean time between retrieval requests is 100ms.
        INITIAL     X11,300 Mean time between updates is 300ms.
        INITIAL     X20,52  Disc revolution time is 52ms.
        GENERATE    X11,FN1
        ASSIGN      2,K2
        ASSIGN      3,K2
        TRANSFER    ,GONE
        GENERATE    X10,FN1
        ASSIGN      2,K1
        ASSIGN      3,FN2
 GONE   TRANSFER    .950,,NORM
        ASSIGN      1,K5
        PRIORITY    7
 GEN    ASSIGN      4,X*2,1
 PRE    QUEUE       3
        PREEMPT     1
        DEPART      3
        ADVANCE     *4
        RETURN      1
        TRANSFER    ,DISQ
 NORM   ASSIGN      4,X*2,1
        TEST E      *2,K3,RNU
        GATE NU     1,BUS
 COMQ   QUEUE       1
        TABULATE    2
 COMK   LINK        1,P4,CPU
 CPU    SEIZE       1
        DEPART      1
        SAVEVALUE   50,*2
        ADVANCE     *4
        RELEASE     1
        UNLINK      1,CPU,1
 DISQ   QUEUE       2
 DIS    ENTER       1,1
        DEPART      2
        ADVANCE     K90,K90
 GAT    GATE SNF    2,REV
```

Figure 10. GPSS III Assembly Input for Run 3



```
                                                                        PAGE  2
        ENTER         2,1
        TEST E        *2,K2,BIG
        ADVANCE       10
ARM     LEAVE         1,1
        LEAVE         2,1
        LOOP          3,ORD
        TEST NE       *1,K5,SPEC
        TABULATE      1
TER     TERMINATE     1
BUS     TEST E        X50,K3,COMQ
        SAVEVALUE     51+,K1
        TRANSFER      ,COMQ
REV     ADVANCE       X20
        TRANSFER      ,GAT
RNU     TEST E        *2,K1,COMQ
        GATE U        1,COMQ
        TEST E        X50,K3,COMQ
        TRANSFER      ,PRE
BIG     ADVANCE       K30,K20
        TRANSFER      ,ARM
SPEC    TABULATE      5
        TRANSFER      ,TER
ORD     TEST E        *1,K5,NORM
        TRANSFER      ,GEN
        START         1000,,100,1
        END
```

Figure 10.  Concluded

Run 1: Lightly loaded system limited disc access equipment;

Run 2: Same disc access equipment, heavier load, saturated condition; and

Run 3: Heavy load, added access capability, improved performance.

Table 1 gives a summary of these results, together with a listing of the values of the design and load factors for each run. It is not claimed that these runs examine the performance of a spectrum of possible designs; they were conducted to illustrate the magnitude of loads that could be handled with currently available equipment configurations. The conclusions which can be drawn from these results are presented in the following paragraphs.

## 2.8 Discussion of Results

The results of most significant interest pertain to the response time and utilization columns. The response time results show the relatively marginal value of incorporating elaborate priority schemes for expediting access to the CPU by special users. For the system studied, and any similar system requiring sharing of both a CPU and data bank, the processor is rarely fully utilized (as may be seen from the utilization columns). However, it seems to be common in the design of such systems to devote much attention to multi-level priority access schemes for CPU service while in fact the real bottleneck appears at the access to the data bank. Referring to Table I, it is seen that the average service time for both normal and special requests, for Runs 1 and 3 where no saturation was experienced, can be calculated by multiplying the average access time by the number of accesses required for the requests. For Run 1, the average data retrieval time was 100 msec (90 for access, 10 for readout); for Run 3, the average retrieval time was (for R-type requests) 130 msec. For R-type requests, which predominate two to one for Run 1 and three to one for Run 3, the corresponding average response times would be .7 and .91 seconds, assuming no waiting in queues. Factoring in the shorter response times for update-type requests, it is easy to see that the actual average response times of .67 and .84 seconds for the two runs represent almost entirely the summation of access delays while going through the necessary calls on bulk memory to satisfy the requests, and that little delays were experienced in queues. No priority algorithms can speed this process, since the speed with which the disc arm finds the proper location for the address being accessed cannot be increased.

The conclusion to be drawn from the results presented is that in a multi-user, time-shared storage and retrieval system, using contemporary equipment configurations, the speed of computation is far out of proportion

| | Mean Tir<br>between<br>(R = Retr:<br>(U = Upd<br>Exponent<br>Distribut | ent<br>puter<br>ation | Percent<br>Channel<br>Utilization | Computer<br>Queue<br>Size | | Disc<br>Queue<br>Size | |
|---|---|---|---|---|---|---|---|
| | | | | Avg | Max | Avg | Max |
| Run 1 | 150 msec<br>300 msec | .9 | 54.0 | 0.3 | 7 | 0.1 | 7 |
| Run 2 | 100 msec<br>300 msec | .0 | 76.0 | 0.3 | 6 | 21.1 | 59 |
| Run 3 | 100 msec<br>300 msec | .0 | 44.0 | 0.9 | 10 | 0.1 | 5 |

* Based on outpu

Note 1: This resu

to the speed of access to the data base. Priority algorithms are not applicable, since they can only be applied to the re-ordering of positions in queues, and this cannot solve the irreducible delays in data access.

Comparing Run 2 with Run 1, it is observed that the system can be driven into saturation by only a minor change in load and delay parameters. Run 1 displays characteristics of a highly underloaded system; buffers are rarely occupied and utilization rates are quite nominal. The difference between Run 1 and Run 2 is a 33 percent increase in input rate and a 20 percent increase in the access and delay time. This change was sufficient to make the system reach saturation after only 12 seconds of simulated operation. Based on this admittedly incomplete simulation, it might be concluded that systems having these general characteristics saturate when their input limit threshold is exceeded only briefly.

### 3. Decentralized Computation Model

At the initiation of this study, it had been planned to model and simulate two alternative implementations for personnel data management systems, each of which would offer centralized data storage to permit remote access to most current data, but which differed in the placement of the computation function. A system providing a small, fast computer at each remote site, communicating with centralized data storage, appeared as an attractive alternative to the system concept described in the preceding section. However, investigation of the power of the centralized computation system quickly revealed that the decentralized computers would prove far too expensive to be practical.

The system as described in the preceding section could easily accommodate hundreds of simultaneous users, each having the capability to perform substantial computational tasks on retrieved data. Its cost, including communications, has been estimated in the range of $2,000,000 to $3,000,000. Of this, only about 15 percent constitutes the cost of the actual CPU. To disperse this computational power in a number of remote locations requires installation of remote computers, each of which should be capable of executing the same computations as those performed centrally. In addition, the communication lines would have to be given additional capacity, since the disc-computer data transfers constitute much more data than the console-computer data transfers. Small, fast computers having the computational capability assumed for the centralized system, are priced in the $80,000 range without bulk memory. Hence, it is estimated that only for fewer than 6 simultaneous users would the decentralized computation alternative be economically justified, and the added communication costs might make the threshold even lower.

## 4. Extension of the Model to Other Systems

While the system model, described in paragraph 2 of this section, was constructed and tested for a typical personnel data management application, it is applicable without significant change to a large number of other types of data management systems, including a variety of command and control applications. The basic functions are found in other systems requiring multiple-user, on-line access to a large data base. Such requirements are found in logistic, air traffic control, surveillance, transportation management, intelligence, and many other similar systems. A modified form of the same model could be very useful for determining the capability of alternative processing and data handling subsystem designs for implementation of improved versions of any of these applications.

# APPENDIX I

## FILE PARTITIONING ANALYSIS

Suppose a file consisting of N records is divided into M subfiles. Let the subfiles be indexed from $i = 1$ to $i = M$. Let the length of the i-th subfile be $L_i$. Then

$$\sum_{i=1}^{M} L_i = N$$

Let us suppose that searching is done as follows:

A batch of n search qualifiers is served at a time. The search is done by going through the subfiles in order, from $i = 1$ to $i = M$. For each subfile, it is possible to determine whether any of the search qualifiers in the batch pertain to that subfile. If none do, one does not read any records from that subfile, but instead goes to the next subfile. If some search qualifiers do pertain to the given subfile, one reads the records from this subfile serially, until one has found all the pertinent records, and then one goes to the next subfile. (The remaining records of the given subfile are unread.) Each search qualifier describes exactly one record in the whole file. It is assumed that the n search qualifiers of a given batch describe n records in the file, i.e., no two qualifiers of a given batch describe the same record. It is further assumed that any record of the file is as likely to be described by an incoming search qualifier as any other record.

Let us define X as:

X = the number of records of the file which are read during the process of serving a batch of n search qualifiers.

It is desired to find

E(X) = the average value of X.

In order to do this, let us define

$X_i$ = the number of records of the i-th subfile which are read during the process of serving a batch of n search qualifiers.

Now since

$$X = \sum_{i=1}^{M} X_i$$

we have

$$\boxed{E(X) = \sum_{i=1}^{M} E(X_i)} \qquad (1)$$

where $E(X_i)$ is the average value of $X_i$. Thus the problem reduces to finding the $E(X_i)$.

In order to compute $E(X_i)$ we begin by computing $P(X_i \leq L_i - k)$, which is the probability that $X_i \leq L_i - k$ where k is any integer such that $0 \leq k \leq L_i$. To do this, number the search qualifiers from $j = 1$ to $j = n$, and let us define

$C_j$ = the event that the j-th search qualifier does not refer to the $(L_i - k + 1)$-th through $L_i$-th records of the i-th subfile.

Then

$$\begin{aligned} P(X_i \leq L_i - k) &= P(C_1 \cdot C_2 \cdots C_n) \\ &= P(C_1) \cdot P(C_2 \mid C_1) \cdots P(C_n \mid C_1 \cdots C_{n-1}) \\ &= \frac{N-k}{N} \cdot \frac{N-k-1}{N-1} \cdots \frac{N-k-(n-1)}{N-(n-1)} \end{aligned} \qquad (2)$$

For any k such that $0 \leq k < L_i$,

$$P(X_i = L_i - k) = P(X_i \leq L_i - k) - P(X_i \leq L_i - k - 1)$$

$$= \frac{N-k}{N} \cdots \frac{N-k-(n-1)}{N-(n-1)} - \frac{N-(k+1)}{N} \cdots \frac{N-(k+1)-(n-1)}{N-(n-1)}$$

$$= \frac{N-k}{N} \cdots \frac{N-k-(n-1)}{N-(n-1)} \cdot \left\{ 1 - \frac{N-(k+1)-(n-1)}{N-k} \right\}$$

$$= \frac{N-k}{N} \cdots \frac{N-k-(n-1)}{N-(n-1)} \cdot \frac{1+(n-1)}{N-k}$$

$$= \frac{n}{N} \cdot \frac{N-k-1}{N-1} \cdots \frac{N-k-(n-1)}{N-(n-1)} \tag{3}$$

which can now be used to compute $E(X_i)$:

$$E(X_i) = \sum_{k=0}^{L_i} (L_i - k) \cdot P(X_i = L_i - k)$$

$$= \sum_{k=0}^{L_i - 1} (L_i - k) \cdot P(X_i = L_i - k) \tag{4}$$

$$E(X_i) = \sum_{k=0}^{L_i - 1} (L_i - k) \cdot \frac{n}{N} \cdot \frac{N-k-1}{N-1} \cdots \frac{N-k-(n-1)}{N-(n-1)}$$

$$= \frac{n}{N} \cdot \frac{1}{(N-1) \cdots (N-(n-1))} \sum_{k=0}^{L_i - 1} (L_i - k)\,(n-k-1) \cdots (N-k-(n-1)) \,. \tag{5}$$

In order to evaluate the sum (5), let us put

$$P_n(k) = k \cdot (k-1) \cdots (k-(n-1))$$

and put

$$Q_n(k) = k \cdot (k-1) \cdots (k-(n-1)) \cdot (k-n)$$

$$Q_n(k) = P_n(k) \cdot (k-n) \qquad (7)$$

Lemma. For any integers A, B with $A \leq B$,

$$\sum_{k=A}^{B} P_n(k) = \frac{1}{n+1} \cdot \left\{ Q_n(B+1) - Q_n(A) \right\} \qquad (8)$$

Proof.

$$Q_n(k+1) - Q_n(k) = (k+1) \cdot k \cdots (k-(n-1)) - k \cdot (k-1) \cdots (k-n)$$

$$= k \cdot (k-1) \cdots (k-(n-1)) \left\{ (k+1) - (k-n) \right\}$$

$$= (n+1) \cdot k(k-1) \cdots (k-(n-1))$$

$$= (n+1) \cdot P_n(k) .$$

Hence,

$$\sum_{k=A}^{B} P_n(k) = \frac{1}{n+1} \cdot \sum_{k=A}^{B} \left\{ Q_n(k+1) - Q_n(k) \right\}$$

$$= \frac{1}{n+1} \left\{ Q_n(B+1) - Q_n(A) \right\}$$

Q.E.D.

In order to use the lemma in evaluating the sum in (5) we write

$$(L_i - k) = (L_i - N) + (N - k)$$

so that the sum in (5) may be written

$$\sum_{k=0}^{L_i-1} (L_i-k)\,(N-k-1)\cdots(N-k-(n-1))$$

$$= (L_i-N)\cdot \sum_{k=0}^{L_i-1} (N-k-1)\ldots(N-k-(n-1))$$

$$+ \sum_{k=0}^{L_i-1} (N-k)\,(N-k-1)\cdots(N-k-(n-1))$$

$$= (L_i-N)\cdot \sum_{p=N-1}^{N-L_i} P_{n-1}(p) + \sum_{p=N}^{N-L_i+1} P_n(p)$$

$$= \frac{L_i-N}{n}\cdot\left\{Q_{n-1}(N) - Q_{n-1}(N-L_i)\right\}$$

$$+ \frac{1}{n+1}\cdot\left\{Q_n(N+1) - Q_n(N-L_i+1)\right\}$$

$$= \frac{L_i-N}{n}\cdot N\cdot(N-1)\cdots(N-(n-1))$$

$$+ \frac{1}{n+1}(N+1)\cdot N\cdots(N-(n-1))$$

$$- \left\{\frac{L_i-N}{n}\cdot 1 + \frac{N_i-L+1}{n+1}\right\}\cdot(N-L_i)\cdots(N-L_i-(n-1))$$

$$= \frac{L_i-N}{n}\cdot N\cdot(N-1)\cdots(N-(n-1))$$

$$+ \frac{1}{n+1}\cdot(N+1)\,N\cdots(N-(n-1))$$

$$+ \frac{N-L_i-n}{n(n+1)}\cdot(N-L_i)\ldots(N-L_i-(n-1))$$

Substituting this expression for the sum into (5),

$$
\begin{aligned}
E(X_i) &= (L_i - N) + \frac{n}{n+1} \cdot (N+1) \\
&\quad + \frac{N-L_i-n}{n+1} \cdot \frac{N-L_i}{N} \cdots \frac{N-L_i-(n-1)}{N-(n-1)}
\end{aligned}
$$

$$
\begin{aligned}
E(X_i) &= L_i - \frac{N-n}{n+1} \\
&\quad + \frac{N-L_i-n}{n+1} \cdot \frac{N-L_i}{N} \cdots \frac{N-L_i-(n-1)}{N-(n-1)}
\end{aligned}
\tag{9}
$$

which can be written as:

$$
\boxed{
\begin{aligned}
E(X_i) &= L_i - \frac{N-n}{n+1} \\
&\quad + \frac{1}{n+1} \cdot \frac{(N-L_i)! \cdot (N-n)!}{N! \cdot (N-L_i-n-1)!}
\end{aligned}
}
\tag{9'}
$$

Putting

$$
K = \frac{(N-L_i)! \cdot (N-n)!}{N! \cdot (N-L_i-n-1)!}
\tag{10}
$$

we then have

$$
E(X_i) = L_i - \frac{N-n}{n+1} + \frac{K}{n+1}
\tag{11}
$$

If we substitute $L + (N-L)$ for $N$ in (11), we can rewrite (11) as

$$
E(X_i) = \frac{n}{n+1} \cdot (L_i + 1) - \frac{N-L_i-n}{n+1} + \frac{K}{n+1}
\tag{11'}
$$

## Computation for small n.

If n is small, there is no difficulty in making the computation directly. For instance, if $N = 1000$, $L_i = 100$, and $n = 3$, then substituting directly into (9):

$$E(X_i) = 100 - \frac{997}{4} + \frac{897}{4} \cdot \frac{900}{1000} \cdot \frac{899}{999} \cdot \frac{898}{998}$$

$$\begin{array}{lll@{\qquad}lll}
\log 900 & = & 2.9542425 & \log 1000 & = & 3.0000000 \\
\log 899 & = & 2.9537597 & \log 999 & = & 2.9995655 \\
\log 898 & = & 2.9532763 & \log 998 & = & 2.9991305 \\
\log 897 & = & \underline{2.9527924} & \log 4 & = & \underline{\phantom{0}.6020600} \\
 & & 11.8140709 & & & 9.6007560 \\
 & & \underline{\phantom{0}9.6007560} & & & \\
\end{array}$$

$$\text{arc log} \quad (2.2133149) = 163.424$$

$$E(X_i) = 100 - 249.25 + 163.424 = \boxed{14.174}$$

If this file of 1000 records consists of 10 equal subfiles, each of 100 records, then by (1):

$$E(X) = \sum_{i=1}^{10} E(X_i) = \sum_{i=1}^{10} 14.174$$

$$= 10 \times 14.174 = 141.74$$

## An inequality for $E(X_i)$

For some applications it can be useful to have a very rough idea of the size of $E(X_i)$. It is possible to obtain an inequality for $E(X_i)$ which might help in giving an estimate. Define

$N_i$ = the number of search qualifiers of a given batch, which describe records of the i-th subfile.

Then $N_i$ is an integer-valued random variable, with $0 \leq N_i \leq L_i$. Now

$$E(X_i) = \sum_{j=0}^{L_i} E(X_i \mid N_i = j) \cdot P(N_i = j)$$

Now it has been shown in (2) that

$$E(X_i \mid N_i = j) = \frac{j}{j+1} \cdot (L_i + 1),$$

hence

$$E(X_i) = (L_i + 1) \cdot \sum_{j=0}^{L_i} \frac{j}{j+1} \cdot P(N_i = j)$$

Now the function

$$g(j) = \frac{j}{j+1}$$

is concave on the interval $0 \leq j < \infty$; since one has $0 \leq P(N_i = j) \leq 1$ for each $j$ $(0 \leq j \leq L_i)$ and since $\sum_{j=0}^{L_i} P(N_i = j) = 1$, it follows that

$$\sum_{j=0}^{L_i} g(j) \cdot P(N_i = j) \leq g\left( \sum_{j=0}^{L_i} j \cdot P(N_i = j) \right)$$

(see reference [ 1 ] for example). Now

$$\sum_{j=0}^{L_i} j \cdot P(N_i = j) = E(N_i)$$

and it is clear that

$$E(N_i) = \frac{nL_i}{N}$$

thus

$$\boxed{E(X_i) \leq \frac{\left[\frac{nL_i}{N}\right]}{\left[\frac{nL_i}{N}\right] + 1} \cdot (L_i + 1)} \tag{12}$$

For example, let us consider the case which was treated previously: $N = 1000$, $L_i = 100$, $n = 3$. Then by (12),

$$E(X_i) \leq \frac{0.3}{0.3 + 1} \cdot (101)$$

giving

$$\boxed{E(X_i) \leq 23.308}$$

This may be compared to the exact answer obtained above, $E(X_i) = 14.174$.

Computation for moderately large n.

According to Stirling's formula,

$$M! \sim \sqrt{2\pi} \cdot M^{M+1/2} \cdot e^{-M} \tag{13}$$

or

$$\log (M!) \sim 1/2 \log 2\pi + (M + 1/2) \log M - M \log e, \tag{14}$$

for any integer $M$ which is sufficiently large. The relative error $R$ of Stirling's formula is estimated by the following inequalities

$$0 < R < \frac{1}{12n - 1} \tag{15}$$

so that if $M \geq 9$ then the relative error is $\leq 1\%$.

If $N$ is sufficiently larger than $L_i + n$, say if

$$N - L_i - n \geq 10$$

then $(N - L_i - n - 1)\,!$ in formula (10) may be approximated by (13) with small error, and indeed all the factorials in (10) may be so approximated. It will usually be the case that (16) holds. Rewriting (10),

$$\begin{aligned} \log K \quad = \quad & \log\,[\,(N - L_i)\,!\,] \;+\; \log\,[\,(N - n)\,!\,] \\ & - \log\,(N\,!) \;-\; \log\,[\,(N - L_i - n - 1)\,!\,] \end{aligned} \tag{17}$$

Substituting (14) into (17) and simplifying,

$$\boxed{\begin{aligned} \log K \quad \sim \quad & (N - L_i + 1/2)\;\log\,(N - L_i) \\ & + (N - n + 1/2)\;\log\,(N - n) \\ & - (N + 1/2)\;\log N \\ & - (N - L_i - n - 1/2)\;\log\,(N - L_i - n - 1) \\ & - \log e \end{aligned}} \tag{18}$$

This can also be written as

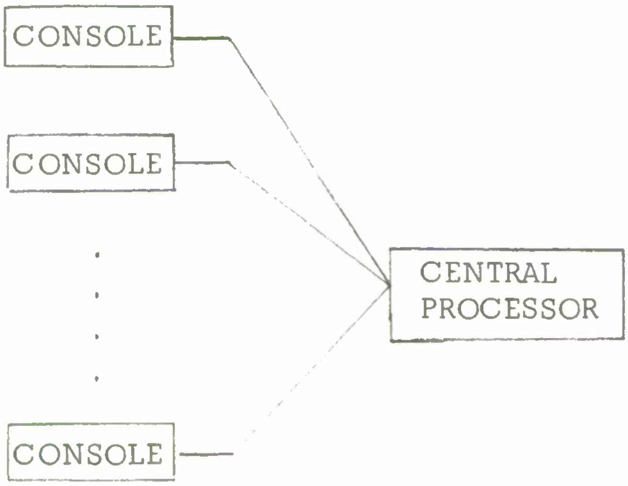
$$\boxed{\begin{aligned} \log K \quad \sim \quad & (N + 1/2)\cdot \left\{ \begin{array}{l} \log\,(N - L_i) + \log\,(N - n) \\ - \log N - \log\,(N - L_i - n - 1) \end{array} \right\} \\ & - L_i \cdot (\log\,(N - L_i) - \log\,(N - L_i - n - 1)) \\ & - n \cdot (\log\,(N - n) - \log\,(N - L_i - n - 1)) \\ & + \log\,(N - L_i - n - 1) - \log e. \end{aligned}} \tag{19}$$

The formula (19) is superior to formula (18) for actual computation. Of course we use (19) only when a direct computation using (9) is too difficult.

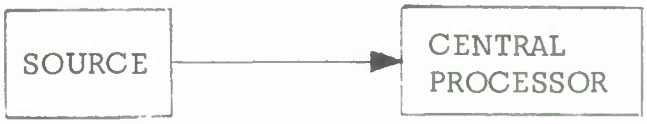
# APPENDIX II

## A MATHEMATICAL MODEL OF A TIME-SHARING SYSTEM

Scherr[14] has constructed a mathematical model of a time-sharing system having n consoles. In this model, each console is regarded as a source feeding the central processor. It is assumed that the central processor switches from console to console rapidly enough that one may consider that it is processing the requests from each console simultaneously; and that the central processor is devoting 1/m of its capability to each active console, if m is the number of active consoles. It is further assumed that if a particular console is not active, then the time interval between now and when it becomes active has an exponential distribution. The amount of processing required for a request from a console is assumed to have an exponential distribution. We can visualize Scherr's model as follows:



We would like to propose another model for a time-sharing system. In this model, we place the emphasis not on the consoles, but on the requests arriving at the central processor; this model may be visualized as follows:



Requests are assumed to originate according to the Poisson distribution, at a rate $\lambda$. It is assumed that the central processor deals with all requests simultaneously and that the amount of processing required by requests has an exponential distribution. Since the central processor's efficiency may vary depending on the number i of requests being processed, we will permit the processing rate to be dependent on i. We define $\mu_i$ to be the rate at which the processor serves requests if there are i requests being processed. A little more precisely, if there are i requests being processed and no more requests are received, then it is assumed that the time to first complete service has an exponential distribution with mean $\mu_i^{-1}$. Thus, $\mu_i$ is the rate of transition to the state where there are i-1 requests in process; each individual request is being processed at a rate $\mu_i/i$ .

Now in many time-sharing systems, much of the swapping is actually performed by peripheral equipment; in such systems, the central processor's efficiency may not depend very significantly on the number of requests. Even so, it is convenient as a mathematical device to permit $\mu_i$ to be dependent on i, as will be discussed later.

This model may be analyzed by using the method of the imbedded Markov chain. We define a state to be any (maximal) interval of time during which the number of requests in process remains constant. One state ends and another begins if either a new request arrives or a request finishes processing. Let $S_n$ denote the number of requests in process during the n-th state. Define:

$$P_{\ell,n} = P(S_n = \ell) \quad , \quad \ell \geq 0.$$

Define: $$\pi_\ell = \lim_{n \to \infty} \left\{ P_{\ell,n} \right\},$$

that is the $\pi_\ell$ are the limiting state probabilities. Define $P_\ell$ = the probability that if $\ell$ requests are in process, then the state terminates with the arrival of a new request. Define $Q_\ell$ = the probability that if $\ell$ requests are in process, then the state terminates by completing the processing of one of these requests. One has:

$$P_\ell = \frac{\lambda}{\lambda + \mu_\ell} \; ; \quad Q = \frac{\mu_\ell}{\lambda + \mu_\ell}$$

Of course $P_\ell + Q_\ell = 1$.

Under appropriate conditions, namely that the capability of the central processor exceeds the requirements of the incoming requests, the Markov chain is stable, and the $\pi_\ell$ are determined uniquely by the conditions:

$$\pi_\ell = Q_{\ell+1} \cdot \pi_{\ell+1} + P_{\ell-1} \cdot \pi_{\ell-1}; \quad \ell \geq 1.$$

$$\pi_0 = Q_1 \cdot \pi_1.$$

$$\sum_{n=0}^{\infty} \pi_n = 1.$$

Now it is one thing to say that the $\pi_\ell$ are uniquely determined, and it is quite another to actually compute the $\pi_\ell$. One may proceed as follows: define numbers $x_\ell$ recursively by:

$$x_0 = 1;$$

$$x_1 = Q_1^{-1};$$

$$x_\ell = Q_\ell^{-1} \cdot \left\{ x_{\ell-1} - P_{\ell-2} \cdot x_{\ell-2} \right\}, \quad \ell \geq 2.$$

If we define:

$$K = \sum_{\ell=0}^{\infty} x_\ell$$

then the $x_\ell$ are related to the $\pi_\ell$ by:

$$\pi_\ell = x_\ell \cdot K^{-1}.$$

Thus it is all a matter of computing K. If we compute $x_1, \ldots, x_N$, where N is large, then approximately:

$$K = \sum_{\ell=0}^{N} x_\ell$$

Therefore the key to the whole problem is to decide how large N must be in order to be able to use the above approximation. While we shall not enlarge further on this problem here, we will remark that for all cases of interest the problem is soluble, and therefore the $\pi_\ell$ are computable. Further, in many cases the $\pi_\ell$ can be computed exactly.

Let $\tau_\ell$ denote the average duration of a state during which there are $\ell$ requests being processed; then:

$$\tau_\ell = \frac{1}{\rho + \mu_\ell}$$

Let $\overline{\pi}_\ell$ denote the probability that at time t there are $\ell$ requests being processed, as $t \to \infty$. Then it may be shown that:

$$\overline{\pi}_\ell = 2\lambda \tau_\ell \cdot \pi_\ell$$

Thus the $\overline{\pi}_\ell$ may be computed.

As an example, let us consider the case where $\mu_i$ is independent of i; thus there is a $\mu$ such that:

$$\mu_i = \mu , \quad i \geq 0 .$$

It is not difficult to see that the $\overline{\pi}_\ell$ equal the state probabilities for a queue with single server, Poisson input, exponential service, so that in fact the problem is already solved. However, one can carry through the computations outlined above; the equations for the $\pi_\ell$ may be solved in closed form, giving:

$$\pi_0 = \frac{\mu - \lambda}{2\mu}$$

$$\pi_\ell = \frac{\mu + \lambda}{2} \cdot \frac{\lambda^{\ell - 1}}{\mu^{\ell + 1}} \cdot (\mu - \lambda)$$

The $\overline{\pi}_\ell$ are then:

$$\overline{\pi}_\ell = (\frac{\lambda}{\mu})^\ell \; (1 - \frac{\lambda}{\mu})$$

in agreement with results for the queueing problem.[15]

The state probabilities $\overline{\pi}_\ell$ give us a measure of information on the length of time a particular request may take. For example, suppose the efficiencies $\mu_\ell$ are independent of $\ell$, so that:

$$\mu_\ell = \mu \, , \quad \ell \geq 0 .$$

Now suppose we enter a request which requires much processing. If the request stays in the system for a long time, then we may consider that, as far as other requests are concerned, the primary effect of the first request is to reduce the processing capability of the system. Thus in effect we have a system in which:

$$\mu_\ell = \frac{\ell}{\ell + 1} \cdot \mu$$

If we compute the corresponding state probabilities $\overline{\pi}_\ell$, and if the amount of time required to process the request is T (assuming that the central processor works only on this request) then the expected time to process the request is approximately:

$$\frac{T}{\sum_{\ell=0}^{\infty} (\overline{\pi}_\ell / \ell + 1)}$$

From this we see that permitting $\mu_i$ to be dependent on i can be a useful mathematical device, even if there is no significant dependence in the actual system being considered.

# APPENDIX III

## PERSONNEL ASSIGNMENT ALGORITHMS

The following discussion of the personnel-assignment problem is limited to three techniques. Each method assumes that a matrix of values indicating the utility of person i on job j is available. In addition, these values are assumed to be error free.[9]

1. Decision Index[10]

This technique assigns "personnel to jobs in a way that will tend to maximize their productivity". This technique does not guarantee an optimal policy in assigning personnel. The procedure consists of defining a Disposition or Decision Index (DI), which alters each element of the value matrix ($C_{pq}$) in a designated manner. The two DI suggested in footnote 2 for the case of n persons and n jobs and m persons and n jobs ($m \neq n$) are, respectively:

$$DI_{pq} = \frac{1}{n(n-1)} \quad n\, C_{pq} - C_{p.} - C_{.q} + C_{..} \tag{1}$$

$$DI_{pq} = \frac{1}{n(m-1)} \quad m\, C_{pq} - C_{p.} - C_{.q} + C_{..} \tag{2}$$

[9] For a discussion of the personnel-assignment problem when the errors in these values are considered, see reference [16].

[10] For more detailed information, see reference [11].

where,

$$C_{..} = \sum_{p=1}^{n \text{ or } m} \sum_{q=1}^{n} C_{pq}$$

$$C_{p.} = \sum_{q=1}^{n} C_{pq}$$

$$C_{.q} = \sum_{p=1}^{n \text{ or } m} C_{pq}$$

$C_{pq}$ = productivity of the $p^{th}$ person on the $q^{th}$ job.

Once the DI have been calculated for each element of the matrix, several techniques are available to obtain the personnel assignments.

A. The first procedure is to compute the DI for each element of the matrix and then select the element with the highest DI as the first assignment. Delete that row and column from the original value matrix and recalculate a new set of DI. Select the element with the highest DI as the second assignment. Continue this procedure until all personnel (or jobs) have been assigned.

B. The second procedure is to calculate the original DI and select the highest DI for assignment. Select the second highest for the second assignment and continue making assignments in descending DI order without recomputing a DI matrix.

C. The third procedure consists of a mixture of technique A and technique B; that is, recomputing the DI matrix after every $K^{th}$ assignment.

The advantage of the Decision Index Technique is that the computations are relatively simple and in addition, the logic required in assigning personnel is simple. The disadvantage of this technique is that the personnel assignment policy is not always the optimal policy.

The time required to calculate a 100 x 100 matrix, if technique A is used, can be estimated as follows.

First assignment required $n^2 = 10,000$ DI computations and $n^2 = 10,000$ comparisons to select the maximum DI.

Total number of DI computations $= n^2 + (n-1)^2 + \ldots + 3^2 + 2^2$ = total number of comparisons.

$$\text{Number DI computations} = \frac{n(n+1)(2n+1)}{6} - 1 = \frac{100(101)(201)}{6} - 1$$

$$= 376,699.$$

Depending on the computer selected, an estimate of the time required to solve a 100 x 100 matrix can now be calculated. The required storage for the technique is $2n^2$.

## 2. The Hungarian Method[12]

The theoretical justification for this method is presented in reference [12]. This method requires an n x n value matrix of <u>positive</u>

integers. An initial cover and an initial set of independent marks must first be defined before entering the two main routines.

Let,

$$a_i \quad = \max r_{ij} \text{ for } i = 1, 2, \ldots, n \text{ (maximum for each row)}$$

$$b_i \quad = \max r_{ij} \text{ for } j = 1, 2, \ldots, n \text{ (maximum for each column)}$$

where

$$r_{ij} \quad = \text{the utility of person } i \text{ on job } j$$

$$a = \sum_{i=1}^{n} a_i \qquad\qquad b = \sum_{i=1}^{n} b_i$$

Then define values $u_i$ and $V_j$ (cover) as follows:

$$\text{If } a \leq b \text{ define } \begin{cases} u_i = a_i & \text{for } i = 1, \ldots, n \\ V_j = 0 & \text{for } j = 1, \ldots, n \end{cases}$$

$$\text{If } a > b \text{ define } \begin{cases} u_i = 0 & \text{for } i = 1, 2, \ldots, n \\ V_j = b_j & \text{for } j = 1, 2, \ldots, n \end{cases}$$

From the cover $\left\{ u_i, V_j \right\}$ and the value matrix $R = (r_{ij})$, a qualification matrix $Q = (q_{ij})$ is defined.

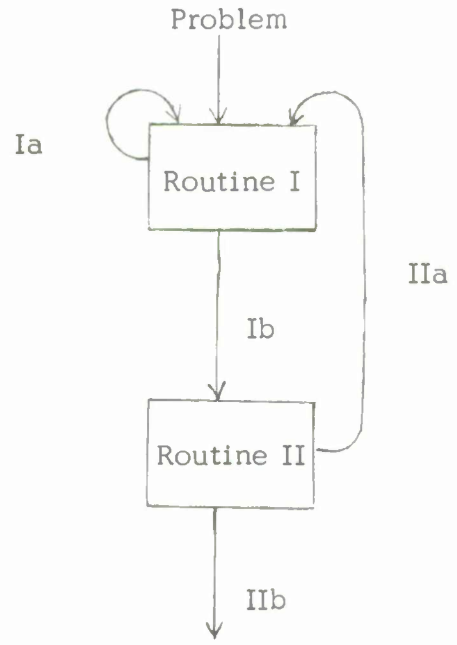
$$q_{ij} \quad = \begin{cases} 1 \text{ if } & u_i + V_j = r_{ij} \\ 0 & \text{otherwise} \end{cases}$$

The independent marks for this initial set are defined as follows:

1. If $a \leq b$, the rows are examined in order and the first 1 in each row without an independent mark (designated as 1*) in its column is changed to a 1*.

2. If $a \geq b$, the rows and columns in statement 1 are interchanged.

The relationship between the two routines[11] of this technique are shown below.



Since the "Hungarian Method" is an iterative process, an estimate of the time required to obtain a solution to an n x n matrix is difficult.

[11] For more detailed information, see reference [ 12] .

This method can be programmed on a small computer readily, and the relation between n and the mean processing time can be obtained. It is important to observe two important outcomes of the iterative method.

1) Every occurence of Ia increases the number of assignments (1*) by one,

2) Every occurrence of IIa decreases the current covering sum ($\Sigma u_i + \Sigma v_j$)

which assures a solution.

### 3. Linear Programming[13]

The personnel-assignment problem can be reformulated into a classical linear programming problem, the transportation problem. This problem has been studied extensively and optimal solutions do exist. A description of the technique is contained in reference [13], chapters 14 and 15, and shall not be repeated here.

Estimates of the computing time required to solve the assignment problem by linear programming can most likely be obtained with a more extensive literature search. Linear programs are now available at many research centers and time estimates should be available. The advantages of this technique are that optimal solutions are obtained and the technique is universally known.

# REFERENCES

[1] UNIVAC Bulletin, UP 2593.20, February 10, 1964.

[2] UNIVAC Bulletin, UP 3951.1, September 9, 1964.

[3] UNIVAC Bulletin, UP 2593.17, September 9, 1964.

[4] IBM 1302 Disc Storage, Models N1 and N2, File Number S360-07.

[5] IBM—Programming Support for the IBM 2321 Data Cell Drive— Preliminary Specifications, File Number 1410/7010-30.

[6] IBM 2321 Data Cell Drive with IBM 1410/7010 Data Processing Systems, File Number 1410-03.

[7] IBM 2321 Data Cell Drive, File Number S360-03.

[8] RCA Model 3488 Random Access Computer Equipment, General Information Manual, June 1964.

[9] RCA 3488 Timing Application Manual.

[10] Adams Associates. Computer Characteristics Quarterly, December 1964.

[11] Ward, J.H. Use of a Decision Index in Assigning AF Personnel, United States Air Force, Tech. Note WADC TN-59-38, August 1959.

[12] Kuhn, H.W. "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Vol.2, 1955. pp.83-97.

[13] Dantzig, George B.· Linear Programming and Extensions, RAND Corporation, 1963. AD 418 366.

[14] Scherr, A.L. An Analysis of Time-Shared Computer Systems, Thesis, Massachusetts Institute of Technology, June 1965.

[15] Saaty, T.L. Elements of Queuing Theory with Applications, McGraw-Hill, New York, 1965. p.95.

[16] King, Willis R. "A Stochastic Personnel-Assignment Model", Journal of the Operations Research Society of America, January-February 1965. pp. 67-81.

# BIBLIOGRAPHY

Adams Associates. Computer Characteristics Quarterly, December 1964.

Dantzig, George B. Linear Programming and Extensions, RAND Corporation, 1963.

Hardy, Godfrey N., Littlewood, J.E., and Polya, G. Inequalities, Cambridge University Press, 1934.

IBM 1302 Disc Storage, Models N1 and N2, File Number S360-07.

IBM 2321 Data Cell Drive, File Number S360-03.

IBM 2321 Data Cell Drive with IBM 1410/7010 Data Processing Systems, File Number 1410-03.

IBM—Programming Support for the IBM 2321 Data Cell Drive— Preliminary Specifications, File Number 1410/7010-30.

King, Willis R. "A Stochastic Personnel-Assignment Model", Journal of the Operations Research Society of America, January-February 1965. pp. 67-81.

Kuhn, H.W. "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Vol. 2, 1955. pp. 83-97.

RCA Model 3488 Random Access Computer Equipment, General Information Manual, June 1964.

RCA 3488 Timing Application Manual.

Saaty, T.L. Elements of Queuing Theory with Applications. McGraw-Hill, New York, 1965. p. 95.

Scherr, A.L. An Analysis of Time-Shared Computer Systems, Thesis, Massachusetts Institute of Technology, June 1965.

UNIVAC Bulletin, UP 2593.17, September 9, 1964.

UNIVAC Bulletin, UP 2593.20, February 10, 1964.

UNIVAC Bulletin, UP 3951.1, September 9, 1964.

Ward, J.H. Use of a Decision Index in Assigning AF Personnel, United States Air Force Tech. Note WADC TN-59-38, August 1959.

## DOCUMENT CONTROL DATA - R&D

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1. ORIGINATING ACTIVITY *(Corporate author)*<br>General Electric Company<br>Information Systems Operation<br>Washington, D.C. | 2a. REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b. GROUP<br>N/A |

3. REPORT TITLE

THE APPLICATION OF LARGE-SCALE COMPUTERS TO U.S. AIR FORCE INFORMATION SYSTEMS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

None

5. AUTHOR(S) *(Last name, first name, initial)*

Campbell, John B.
McCabe, John P.
Nevans, Essie

| | | |
|---|---|---|
| 6. REPORT DATE<br>MARCH 1966 | 7a. TOTAL NO. OF PAGES<br>74 | 7b. NO. OF REFS<br>16 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO.<br>AF 19(628)-4963 | 9a. ORIGINATOR'S REPORT NUMBER(S)<br>ESD-TR-66-137 |
| b. PROJECT NO.<br>2801<br>c. Task 01<br>d. | 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*<br>None |

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

| | |
|---|---|
| 11. SUPPLEMENTARY NOTES<br>None | 12. SPONSORING MILITARY ACTIVITY<br>Directorate of Computers<br>Electronic Systems Division<br>L.G. Hanscom Field, Bedford, Mass. |

13. ABSTRACT

Two centralized Air Force systems -- one dealing with finance and one with personnel assignment -- were used to study applications of large-scale computers. A generalized time-sharing computer system was modeled and simulation was made to measure query response time for various hypothetical conditions.

| 14. KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Data file structure | | | | | | |
| Man-job assignment | | | | | | |
| Modeling and simulation | | | | | | |
| Time-sharing computer | | | | | | |
| Remote consoles | | | | | | |

## INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, aubcontractor, grantee, Department of Defense activity or other organization (*corporate author*) iasulng the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" la included. Marking is to be in accordance with appropriate security regulationa.

2*b*. GROUP: Automatic downgrading ia specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital lettera. Titles in all caaea ahould be unclaasified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parentheais immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appeara on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers (*either by the originator or by the sponsor*), also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by aecurity claasification, using atandard statementa auch as:

(1) "Qualified requestera may obtain copiea of thia report from DDC."

(2) "Foreign announcement and diaseminatlon of thia report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copiea of this report directly from DDC. Other qualified DDC users ahall request through

________________________________________ ."

(4) "U. S. military agencies may obtain copiea of this report directly from DDC. Other qualified usera shall request through

________________________________________ ."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

________________________________________ ."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring (*paying for*) the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, *or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification ia required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional