1967

Fossal: a simulation model for the evaluation of design and management decisions of an on-line real-time information retrieval system

James Patrick Foster
Lehigh University

FOSSAL: A SIMULATION MODEL
FOR THE
EVALUATION OF DESIGN AND MANAGEMENT DECISIONS
OF AN
ON-LINE REAL-TIME
INFORMATION RETRIEVAL SYSTEM

by

James Patrick Foster

A Thesis

Presented to the Graduate Faculty

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Information Science

Lehigh University

1967

This Thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

Sept. 11, 1967
(date)

John M. Carroll
Professor in Charge

David J. Hillman
Head of the Department

ACKNOWLEDGEMENTS

The author wishes to extend his thanks to Professor Robert S. Taylor and Professor John M. Carroll for their assistance in the design of the simulator.

Thanks are also due to fellow graduate students, Ronald Anderson, James Green, and Anthony Amico, whose assistance helped overcome many problems.

Finally, the author wishes to thank his mother and father for their help in compiling this thesis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ABSTRACT

A general discussion of the principles of simulation is given, with emphasis toward an information retrieval system. The design of FOSSAL, the simulation model, is studied, with particular attention paid to the concept of programmable variables. An explanation is given for the use of each sampling distribution and all related data. The simulator is tested by using variables mostly from the Lehigh C.I.S. system. It is determined that 20 to 30 days of simulated time is necessary to obtain reliable results. With the parameters of the G.E. 225 Computer and the Lehigh C.I.S. System as input variables, an arrival rate of 0.25 queries per minute is the maximum that the system can handle if the tape subsystem is used, and a back-log of 30 minutes or less is desired at the end of a day. It is concluded that FOSSAL is extremely useful in helping a designer to evaluate an on-line real-time information retrieval system which has its information stored on a tape or disk subsystem.

I. - Introduction

Two points of extreme interest in the construction of any information system are those of, (A) the type and quantity of users; and (B) the potential (or lack of potential) of the components of the system. Certainly, no individual would be willing to make a decision to build or procure a system without having some indication of the above two factors. However, it is extremely difficult to predict the interrelationships between the users and the system without either observing a system already in use or by performing extensive tests on a prototype system. It is in performing tests on a prototype that simulation can be an invaluable tool. That is, a good simulation model can replace months of tests, as well as months of worry and financial drain.

Since the use of digital computers in the field of information retrieval has been mainly for experimental studies, it is difficult to obtain facts for management decisions by observing different retrieval systems in continuous operation. For this reason this research has undertaken the task of developing a model which can be used to evaluate an on-line real-time information retrieval system before the system is fully operational.

The most significant factor in a successful simulation is the design of a realistic and workable model. The proposed model is designed to operate in an on-line real-time (OLRT) environment, but no use is made of time-sharing techniques. The simulator does not make use of a conversational mode (inter-action between user terminal and computer during processing). In order that the simulator can be of use for more than one individual situation, the model has many interchangeable variables. The variables are parameters so that the system need not be re-programmed as each new situation arises. Thus, it is the aim of this research to design a simulation model that can be used by management to evaluate an on-line real-time document retrieval system in which the files are stored on either a disk or tape subsystem. Specifically, the document retrieval system must be one with coordinate indexed references to documents. In this research the evaluation deals mainly with queue length, response time, and system idle time.

II. - Design of Simulation Model

A. - General Configuration and Environment

The philosophy used in the design of the model is to give every user an equal priority of entering the system, and only penalize the longer requests when the processing time of the items in the ready queue has exceeded the threshold of the system. No emphasis is placed on the number of users entering the system since the prime concern of the simulator is the rate of arrival of the requests and not the particular terminal from which the query originated. The system (Figure 1) is designed to operate with three queues: a ready queue, a back-up queue, and an output queue.

Since no use is made of multi-programming, the central processor operates with only one search request at a time, and the inputs to the central processing unit (CPU) are received in a serial manner from the ready queue. The time needed to process any query is obtained from a sampling distribution, while the speed of the processor is read from a parameter card. Thus, it is immaterial what brand of processor is used as long as the cycle time of the processor is known.

It is of utmost importance that no periods of time exist when the central processor is idle if there are queries to be processed. To avoid such delays, a transfer control unit has been added. The transfer control unit is a software routine whereby a new query can be stored in the ready queue, or some retrieved document information can be transfered out of the system while the central processor is operating on a query.

During a simulation run, the document information files are

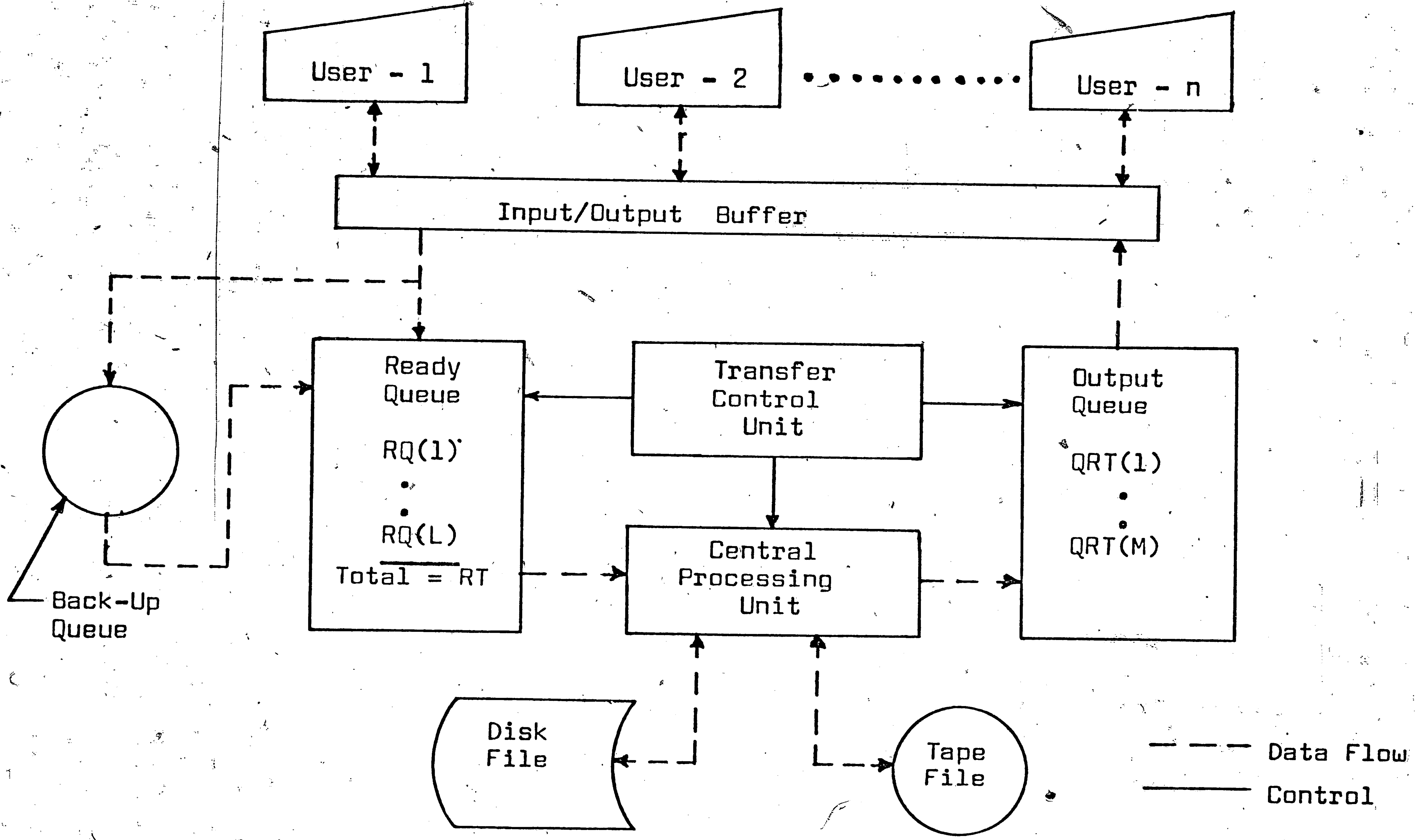


Figure 1. System Configuration

evaluated for only one type of storage. If one is interested in knowing the results for both disk and tape storage, then the simulation would have to be run twice.

The output queue stores all the retrieved information until it can be transmitted to the user via the input/output buffer.

B. - Queueing Mechanisms

As the queries arrive at the system, they are transferred from the I/O buffer to either the ready queue or the back-up queue depending upon the existing threshold value. The ready queue is a first-in-first-out (FIFO) queue which holds as many items as the threshold permits, unless there are a hundred items in the queue (Figure 2). For example, if the threshold of the system is set at 30 minutes and the total time to process the items in the ready queue is 22 minutes, then the next query to arrive in the buffer can be placed in the ready queue, no matter what the length of the new query. (Note: During actual operations the total process time could be estimated by the number of terms in the search). Had the

		1	2	3	4	5
IF	Search time 3.0 minutes	Y	Y	Y	N	N
	Process time of ready queue threshold	N	Y	Y	N	Y
	Items in ready queue 100	Y	Y	N	Y	-
THEN	Place new arrival in ready queue	X	X		X	
	Place new arrival in back-up queue			X		X

Figure 2. Decision Table for New Arrival

total processing time of the items in the ready queue exceeded the 30 minutes, the query would have been placed in the back-up queue provided no priority existed. A priority is given to those search requests that take less than three minutes of CPU time. Thus, even though the ready queue may have surpassed the threshold value, a query that takes only one minute to process will still be placed at the end of the FIFO ready queue. However, there is a limit of one hundred queries in the ready queue. Once the 100 limit has been exceeded, all queries are placed in the back-up queue.

The ready queue is designed as core storage, while the back-up queue is a tape unit. Items that are placed in the back-up queue are removed whenever the total processing time of the items in the ready queue drops below the threshold value of the system. The back-up queue also operates as a FIFO queue.

The output queue is used to store the retrieved information if the information is arriving from the CPU faster than the I/O buffer can transmit it at line speed to the user. Under normal conditions a query would retrieve enough documents to cause the output to back up, thus necessitating the use of the output queue.

C. - Disk Storage

For the efficient and economic operation of a file in an OLRT environment, the use of addressable bulk storage is almost a necessity. In this simulation a disk unit is used as addressable bulk storage on which are stored the serial and inverted files necessary to perform a Boolean search. The inverted file contains all the index terms in the system, with the documents characterized by each term being listed under the term. The serial file contains the bibliographic information for each document. If a two-term search is being evaluated, the two terms are located in the inverted file, the documents indexed by these two terms are read from the inverted file, and a check is made in the processor to see if there are any documents that satisfy the query. If a particular document is judged relevant to the query, the document is then located on the serial file (also on the disk), and transmitted to the user. As previously mentioned, the simulator is not designed to evaluate only a specific piece of equipment, rather there are many variable parameters read into the system, depending upon the user's specifications.

For each record that is accessed on the disk, the following times are encountered:

a. total access time (this includes the time to seek and position the arm, as well as the rotation of the disk).

b. latency or delay time.

c. read or write time.

When the simulator encounters a query that is requesting some type of information from the disk, the first calculations that are performed are those of fitting the users records to the physical records of the disk. For example, if the average number of documents listed under each term in an inverted file is 100, and each disk record can only handle 90 documents, then the simulator must be made aware of this fact (this necessitates the use of chaining). With this information of the IR records and the physical characteristics of the disk (i.e.-average access time, read/write, etc.), the proper processing can be performed on the search. A complete description of the equations used to calculate the retrieval time from a disk is given in Appendix 1.

D. - Tape Storage

Although a large OLRT system with many users would probably not be dependent on a tape system as its only mass storage device, it is possible that a tape system has its place in a small OLRT system which does not demand sub-minute retrieval times.

The evaluation of the tape system is performed on a serial file only. This file is organized by increasing document number, and contains the bibliographic information plus the terms that characterize the document. When performing a search on this serial file, the entire file must be read to locate the relevant documents for any type of search. A complete description of the retrieval equations for the tape system is given in Appendix 2.

E. - Variable Parameters of the Model

Even though the information retrieval model is designed with certain types of queues and storage media, there are enough variables which a user can assign to permit the model to be a very flexible and useful tool. The variable parameters fall into the following categories:

1. System Characteristics
2. Sampling Distributions
3. Document Collection
4. Central Processor Characteristics
5. Disk Characteristics
6. Tape Characteristics

7. Characteristics of the Terminal Equipment
8. Programming Characteristics

These categories are enumerated showing the symbol that the user must recognize to enter the simulation model.

1. System Characteristics:
   a. ISTART - the starting point for the random number generator.
   b. AMIN - the number of minutes in each simulated day.
   c. KDAYS - the number of days for which the simulation is run.
   d. MSM - the storage media desired.
   e. TH - threshold value desired on the ready queue in minutes.
2. Sampling Distributions:
   a. EXM - mean of the exponential distribution for interarrival time in minutes.
   b. S1M - mean of the Poisson distribution for a one-term Boolean search in documents.
   c. S2M - mean of the Poisson distribution for a two-term Boolean search in documents.
   d. S3M - mean of the Poisson distribution for a three-term Boolean search in documents.
   e. S4M - mean of the Poisson distribution for a four-term Boolean search in documents.
   f. S5M - mean of the Poisson distribution for a five-term Boolean search in documents.
   g. HDM - mean of the exponential distribution for the holding time in the output queue.
3. Document Collection:
   a. BCPD - number of bibliographic characters per document.
4. Central Processor Characteristics:

a. CTT - time to perform a computer machine instruction (cycle time) in microseconds.

5. Disk Characteristics:

a. DAT - disk access time in milliseconds (includes seek and positioning time).

b. DDT - disk delay time after positioning in milliseconds.

c. DRW - disk read/write speed in microseconds per word.

d. WPR - words per disk record.

e. DDR - number of documents held by one record on the disk.

f. DAD - average number of documents for each term of the inverted file on the disk.

6. Tape Characteristics:

a. THS - tape transfer rate in kilocycles.

b. TCM - character mode of tape.

c. TREC - number of records stored on tape.

d. WPRT - number of words per record.

e. TSST - tape handler start-stop time in seconds.

f. ATTR - one-half the average number of terms in a tape record of the serial file.

g. BF - the tape blocking factor.

7. Characteristics of Terminal Equipment:

a. TR - transmission rate of user's equipment in bits per second.

b. BPC - number of bits per character used by terminal equipment.

8. Programming Characteristics:

a. TNI1 - the number of machine instructions necessary to complete a one-term Boolean search after the term is retrieved from the serial file on the tape system.

b. TNI2 - instructions for a two-term Boolean "AND" search (tape system).

c. TNI3 - instructions for a three-term Boolean "AND" search (tape system).

d. TNI4 - instructions for a four-term Boolean "AND" search (tape system).

e. TNI5 - instructions for a five-term Boolean "AND" search (tape system).

f. DNI1 - the number of machine instructions necessary to complete a one-term Boolean search after the term is retrieved from the inverted file on the disk system.

g. DNI2 - instructions for a two-term Boolean "AND" search (disk system).

h. DNI3 - instructions for a three-term Boolean "AND" search (disk system).

i. DNI4 - instructions for a four-term Boolean "AND" search (disk system).

j. DNI5 - instructions for a five-term Boolean "AND" search (disk system).

III. - Discussion of Sampling Distributions

A. Interarrival Time

The simulator starts and continues with the arrival of requests from the users at remote terminals. Even though it is possible for two requests to arrive at the system at the same instant, only one input line is scanned at any given moment, so the net effect is that only one request arrives at a time. This simulation is unclocked or event oriented. To determine the time between two arrivals, the interarrival time is sampled from an exponential distribution (1). Since the random number generator produces numbers with a uniform distribution, it is necessary to transform the numbers. An exponentially distributed random number, u, is obtained from a uniformly distributed random number, y, by the following transformation:

$$u = -(1/\lambda)\ln y$$

where λ=mean arrival rate (arrivals per unit time).

Using a mixed multiplicative congruential random number generator with $0.0001 \leq y \leq 0.9999$, the interarrival time lies between $(0 < u \leq 9.2104)1/\lambda$.

B. Type of Boolean Search

In tests performed by Herner (2) on the types of searches requested from people in a scientific environment, it was found that the number of concepts in the questions varied from one to five with the following percentages for each type:

Number of Concepts	1	2	3	4	5	6
Percent of Questions	12.1	47.2	30.3	8.5	1.9	0

However, when one is concerned with a multiple term search, he is interested in knowing how the search is constructed; that is, what type of Boolean construction is used. In the same study, Herner (3) found that the majority of the requests appeared as "AND" statements. Thus, in this simulation all searches are "AND" searches. Occasionally, a user may wish to search on an author's name rather than on a specific index term. Such a search is treated as a one-term search in this simulation. To find the type of request for the model during a simulation cycle, a sample from the random number generator (uniformly distributed) yields a particular request with the above percentages (see Appendix 3).

C. Number of Recalled Documents

A primary concern in calculating the retrieval time of a request is the number of documents that satisfy the query. The total number of recalled documents varies for each one term search just as much as the number varies between a one-term and a five-term search. The number of recalled documents is assumed to have a Poisson distribution no matter how many terms are in the query. A different mean is used for each one of the five Boolean search distributions. However, the mean of each of these Poisson distributions varies from one document collection to another and is left as a variable for the user to insert in the simulator (see Appendix 6).

D. Output Queue, Holding Time and Response Time

A computer system operates at a greater speed than the transmission lines that carry the data to and from the computer, so in many cases a back-log of output data exists. In order to prevent the central processor from being held back, the system model is designed with an output queue and buffer. The buffer is a device used to change the speed of the data to transmission line speed, and the output queue is used to hold those documents that have been retrieved for the specific queries in the central processor.

The time that any processed query (the sum of the retrieved documents) spends in the output queue is obtained by sampling from an exponential distribution. Burke (4) has found that the output of a queueing system with Poisson input and exponential holding times is again Poisson. Since the model is concerned with the time between departures from the system, interdeparture time, use is made of the exponential distribution. The holding time in the output queue, time between departures, is considered to be the time that the retrieved documents wait in the queue and the time to transfer all the documents in a serial manner to the buffer. An explanation of the equations used in the output queue is given in Appendix 4.

IV. - Explanation of Simulation Program

The main program and all subroutines that are used in this simulation are written in Fortran II and were executed on the G.E. 225 Computer in the Lehigh University Computer Laboratory. However the programs can be loaded and run on any computer that has Fortran II capability and a core storage of 4K words. The simulated run time can be varied considerably by the person doing the evaluation. The simulated run time is one working day, but the number of minutes in this working day is an input parameter. Also, the number of days that the simulation can be re-cycled is a parameter. To execute a simulation on the G.E. 225, 0.5 to 1.0 minutes of computer time is necessary for each simulated day, depending upon the arrival rate under investigation.

A complete listing of the computer program used in the simulation is presented in Appendix 5. The initial steps in the program are devoted to initialization of the parameters and to reading all the variables that a user is evaluating in the simulation. The format for the input variables is depicted in Figure 3 (Section II-E gives an explanation of the units of all the input variables).

The simulation is started by reading a random variable which represents the arrival of a query at the input buffer (Figure 4). The random number generator used in these tests was written to be used with Fortran II (Lehigh Computer Laboratory, #D2.513) and generates random numbers between 0 and .9999. The calling sequence is, Y = FLAT(ISTART). The ISTART location holds the octal equivalent of the Ith number generated which is used to generate the I + 1 number. The generation of the random numbers is by a mixed

Fortran Card No.	Input Variable Name	Computer Format	Columns
1000	ISTART	XXXX	1 to 5
1000	AMIN	XXXX.X	11 to 16
1000	KDAYS	XX	21 to 22
1001	EXM	X.XXXX	1 to 6
1002	HDM	X.XXXX	1 to 6
1002	CTT	XXX.XX	11 to 16
1002	TR	XXXX.	21 to 25
1002	BPC	XXXX.	31 to 35
1003	MSM	X	1
1003	TH	XXX.X	11 to 15
1003	BCPD	XXX.X	21 to 25
1004	S1M	XX.XXX	1 to 6
1004	S2M	XX.XXX	11 to 16
1004	S3M	XX.XXX	21 to 26
1004	S4M	XX.XXX	31 to 36
1004	S5M	XX.XXX	41 to 46
1005	DRW	XXXX.X	1 to 6
1005	DAT	XXXX.X	11 to 16
1005	DDT	XXXX.X	21 to 26
1005	DDR	XXXX.X	31 to 36
1005	DAD	XXXX.X	41 to 46
1005	WPR	XXXX.X	51 to 56
1006	THS	XXXX.X	1 to 6
1006	TCM	XXXX.X	11 to 16
1006	TREC	XXXX.X	21 to 26
1006	WPRT	XXXX.X	31 to 36
1006	TSST	XX.XXX	41 to 46
1006	ATTR	XXXX.X	51 to 56
1006	BF	XXXX.X	61 to 66
1007	TNI1	XXXX.X	1 to 6
1007	TNI2	XXXX.X	11 to 16
1007	TNI3	XXXX.X	21 to 26
1007	TNI4	XXXX.X	31 to 36
1007	TNI5	XXXX.X	41 to 46
1008	DNI1	XXXX.X	1 to 6
1008	DNI2	XXXX.X	11 to 16
1008	DNI3	XXXX.X	21 to 26
1008	DNI4	XXXX.X	31 to 36
1008	DNI5	XXXX.X	41 to 46

Figure 3. Format for Input Variables

multiplicative congruential generator of the form:

$$X_{i+1} = X_i(a) + c \quad (\text{mod } m)$$

In this simulation X_i was picked as 35791 to start the generator, $a = 2^8+1$, and $c = 223$. The random number, Y, is sampled from a flat (uniform) distribution, and is transformed to a random variable from an exponential distribution, U. At the time of arrival of any new query, the information system is updated before any new requests are processed.

The total idle time of the central processor is checked upon the arrival of each new query. If the arrival time of the new query, ART, is greater than the total search-process time of all the items in the system, SWT; the idle time counter, TIT, is increased by the difference between the arrival time and the total search-process time of the ready queue and central processor. Next, the arrival time is checked against the time of the query that is in the central processor. If the arrival time is greater than the total processing time of the central processor (Figure 5), the central processor is updated, and the arrival time is readjusted to account for the removal of the search time from the central processor. If the arrival time is less than that of the item in the central processor, the arrival time is subtracted from the process time and the update sequence is complete (Figure 5). The update sequence of the simulation is a feature which brings the system up to the latest point in time. This accounts for the processing in the system between arrivals.

If the value of ART is greater than zero after updating the central processor, the ready queue is updated; otherwise, the pro-

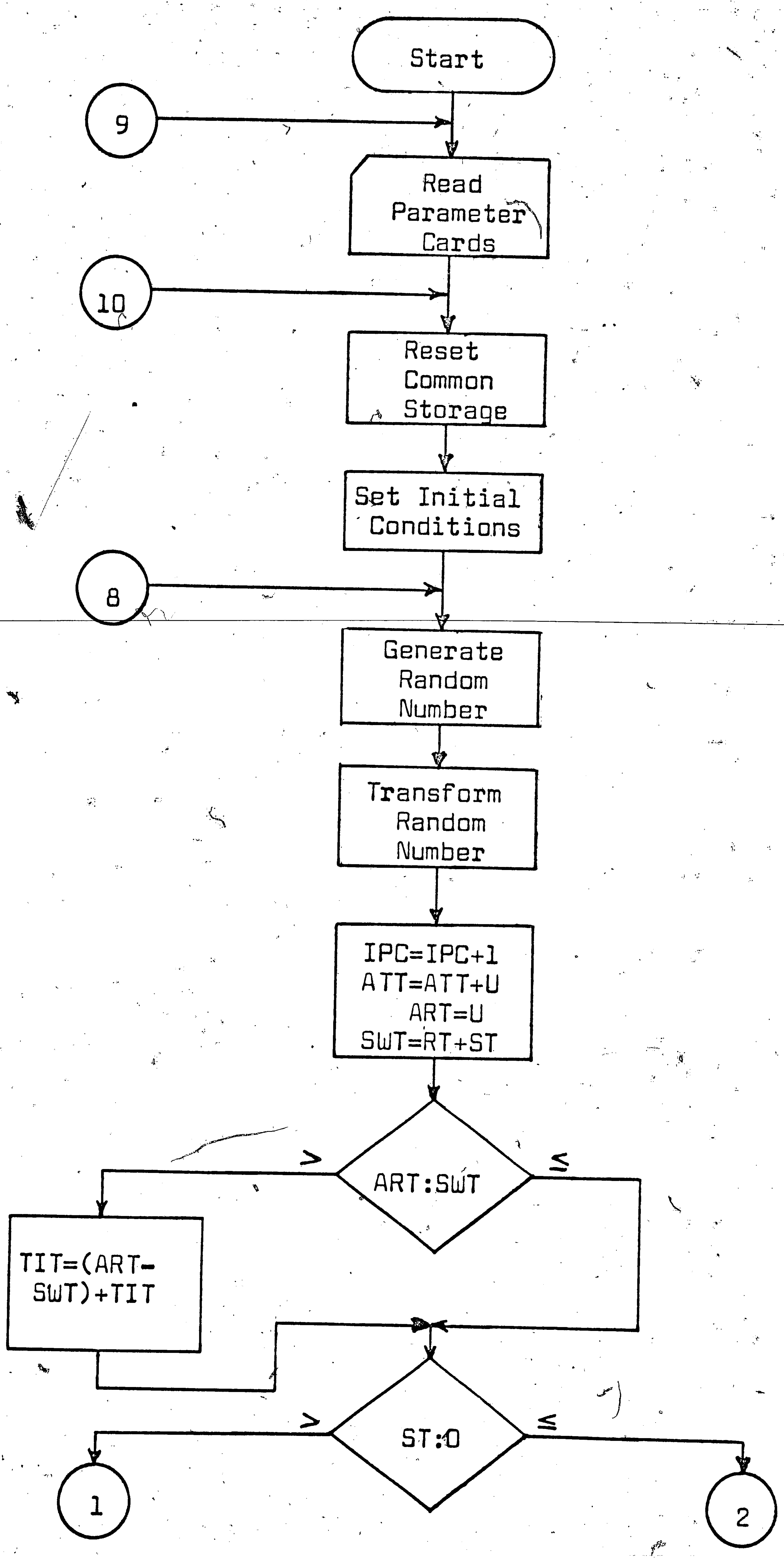


Figure 4. Flow Chart A - Initial Conditions and Calculations

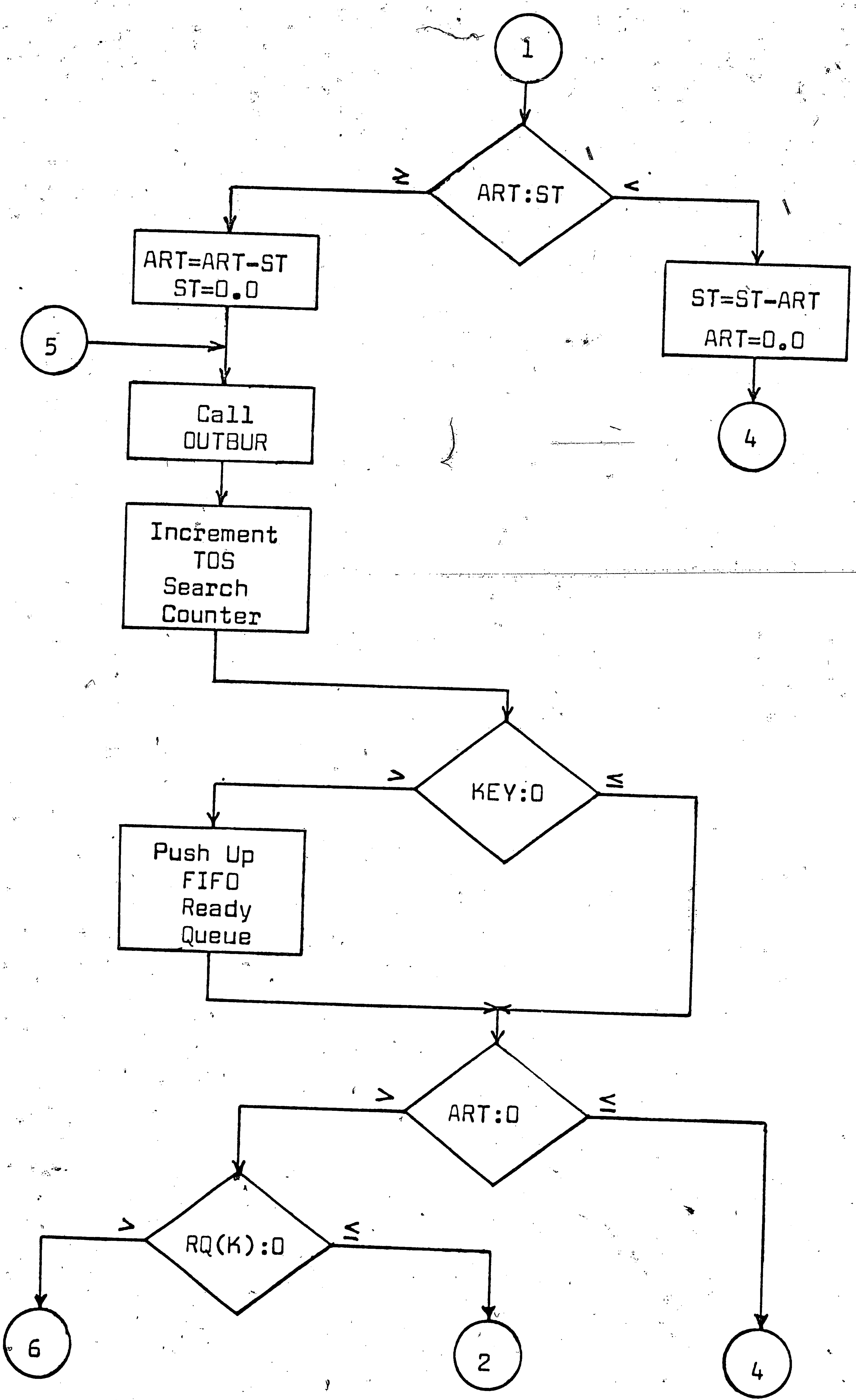


Figure 5. Flow Chart 8 - Update Central Processor

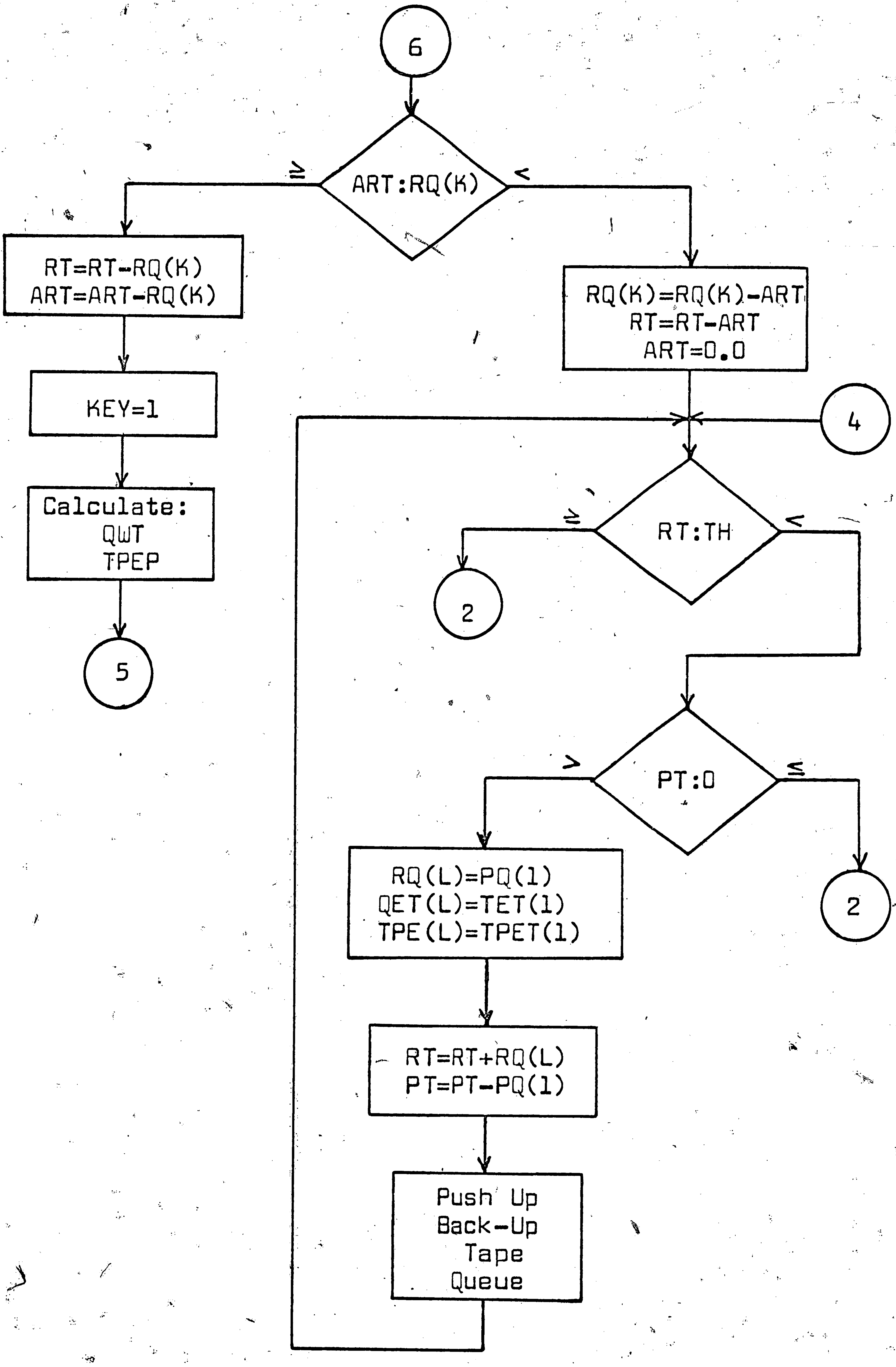


Figure 6. Flow Chart C - Update Back-Up and Ready Queues

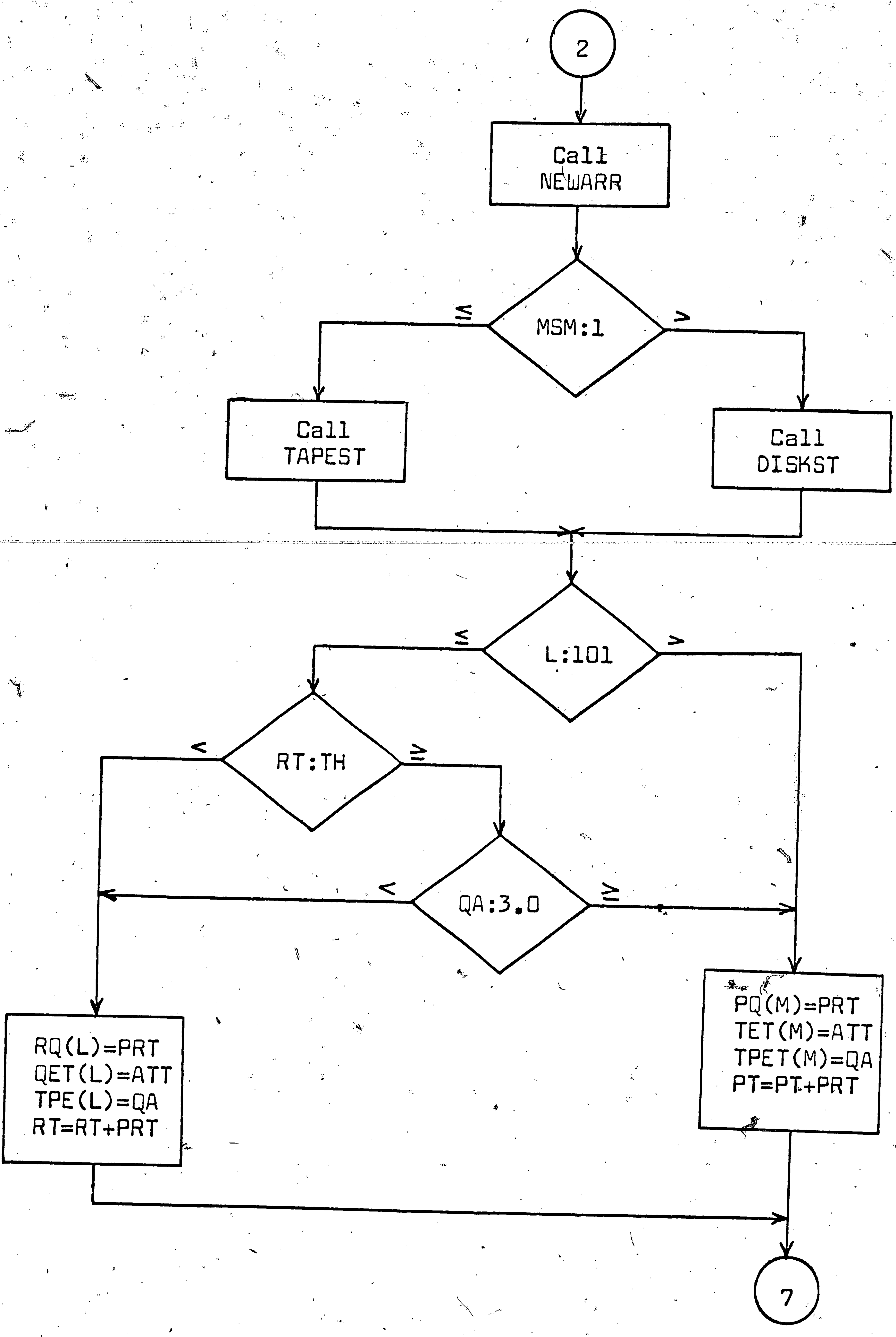


Figure 7. Flow Chart D - Selection and Placement of New Arrival

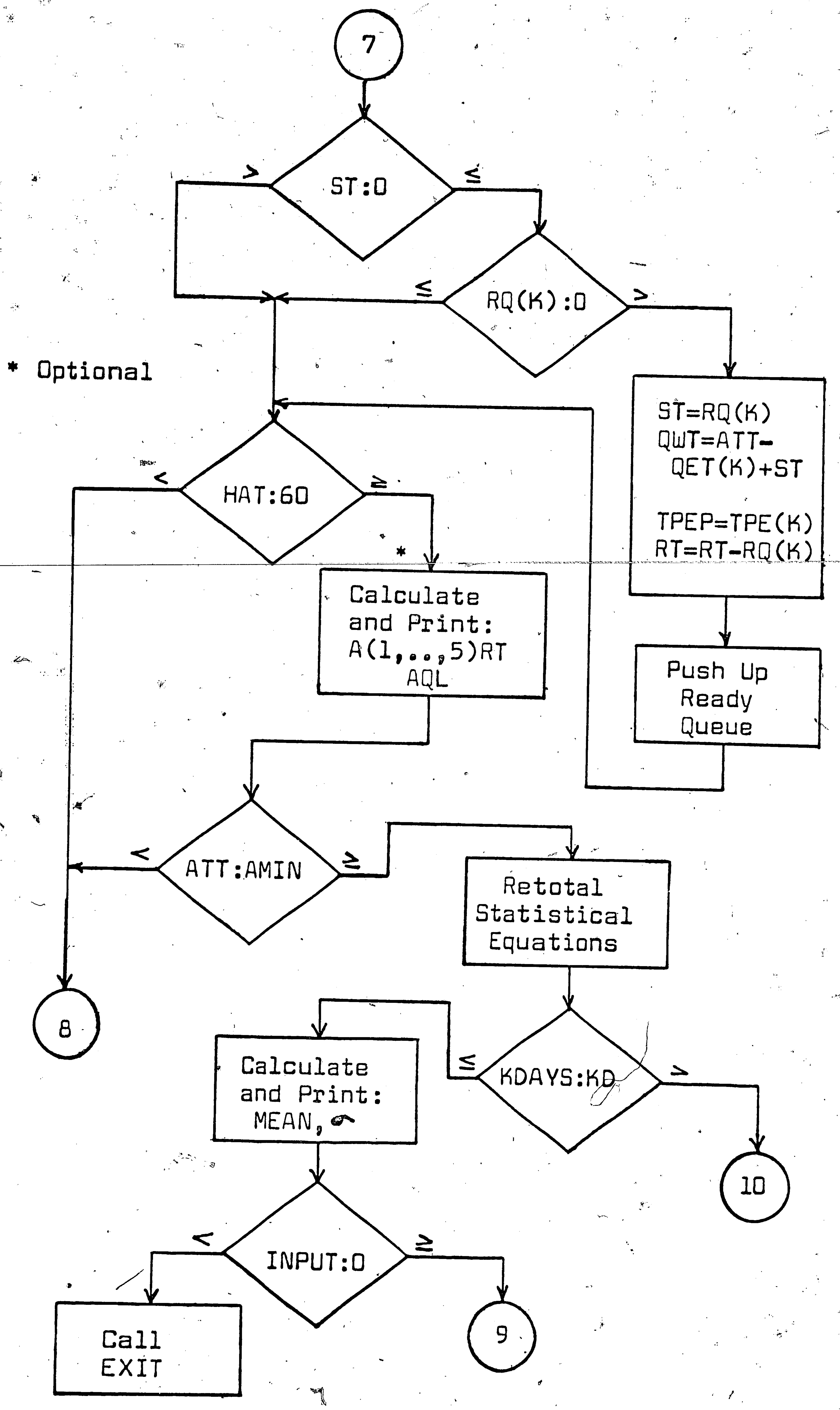


Figure 8. Flow Chart E - Placement of Query in CPU, and Print Routines

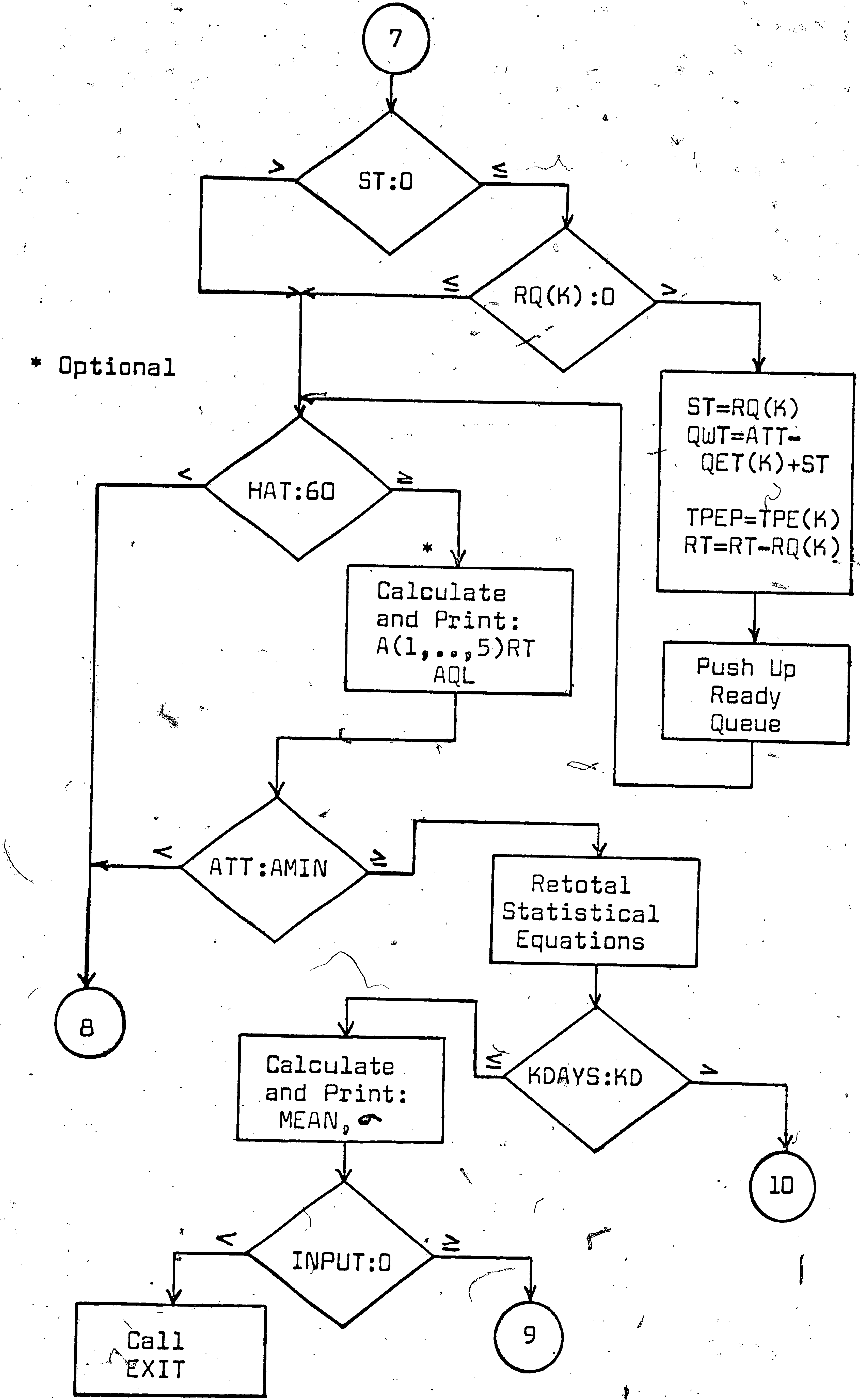


Figure 8. Flow Chart E - Placement of Query in CPU, and Print Routines

cessing of the new query is initiated. The ready queue is updated by removing the top (first) item from the queue and comparing it with the remaining arrival time (Figure 6). If the item removed is greater than the remaining arrival time, the arrival time is subtracted from the item, R(K); and R(K) is placed back in the top position of the ready queue. The processing time of the ready queue is retotaled, and the simulation continues by switching to the back-up tape queue update procedure.

If the process time of the top item that is removed from the FIFO ready queue is less than the remaining arrival time, the item is removed from the ready queue, processed, and the retrieved document information returned to the user via the output queue and buffer (Figure 6).

Any time a query is processed, the retrieved information is returned to the user via the output queue and buffer. This sequence and all related calculations are explained in Appendix 3, which is a subroutine, OUTBUR, called by the main program. The value of the response time, QRT, that is returned from this output subroutine includes the time that has elapsed between when the user's query entered the system and the last piece of document information is returned. This value is used later in the program to calculate the waiting times for the different types of searches. The response time of the five types of searches is stored in the counters T(1,..,5)RT.

The back-up tape queue procedure is the only remaining sequence for the simulator to operate on before proceeding to the processing of the new arrival. The back-up queue is used when the ready queue has a total time, RT, of the items in it of at least TH processing minutes; i.e., the item intended for the ready queue equals or exceeds 3.0 minutes, or there are already 100 items in the ready queue. The ready queue is core storage and is designed to hold 100 items. The value of the threshold, TH, is a variable which is assigned at the beginning of the evaluation. Thus, the back-up queue is only used as an overflow queue for large searches during peak periods for the central processor. If there are some queries on the tape unit queue, and the ready queue is updated to the point that the total time, RT, of the items is less than TH minutes, then an item can be removed from the back-up storage unit, provided the 100 positions in the core storage ready queue are not occupied. Since the back-up storage queue is designed as a FIFO queue, the top item in the back-up queue is removed and placed at the end of the ready queue which is also a FIFO queue (Figure 6).

When the updating of the system is completed, the simulation continues by processing the latest arrival. A special subroutine, NEWARR, is called to calculate the type of search, TOS; and the number of retrieved documents, W. The values of TOS and W are combined to form a composite value, QA (see Appendix 4). Once the values of W and TOS are known, the time to retrieve the information can be calculated. The main program calls either the tape or the disk subroutine depending upon which is selected in the input parameter cards. Appendix 1 gives the calculations necessary to find the time to retrieve a specific query from a disk system. Similarly, Appendix 2 produces such calculations for the tape subroutine. No matter which subroutine is called, the variable PRT, the process-retrieval time, is returned to the main program.

Before the new query is placed in the FIFO ready queue, the total time, RT, of the searches in the ready queue is checked to observe if the threshold, TH, has been exceeded during the previous cycle (Figure 7), or whether the ready queue contains 100 items. If the threshold has been exceeded, the new item is only placed in the ready queue if it is a search of short duration (less than 3.0 minutes), provided the ready queue does not already contain 100 items. A search of 3.0 minutes or greater, with RT exceeding TH minutes, is placed in the tape storage back-up queue until such a time is reached that the ready queue total, RT, drops below TH minutes.

With the ready queue of a FIFO type, a query that enters the ready queue is allocated a spot at the end of the queue, RQ(L). As soon as a new item is placed in the ready queue, a new total is calculated for RT. Besides storing the process-retrieval time, PRT, in the ready queue, the time that the query arrived in the system, ATT, and the value of QA are also retained (Figure 7). This information is retained for calculations further in the simulation sequence.

There exists a good possibility at this point in the simulation that the central processor is vacant due to the initial update procedure. Under such a condition, the system program is designed to remove the top item from the ready queue and move it to the central processor (Figure 8). The waiting time, QWT; the search-process time, ST; and the type of search, TPEP, are all transfered from the ready queue to the central processor for further calculations. The items in the ready queue are pushed up and the queue is retotaled before the simulation continues.

The simulation is now at a point where all the updating and processing is completed, and all that remains are the calculations on the stored data. The simulator has two options for printing out the calculated values: (1) data made available at the end of each simulated hour and day; and (2) data made available at the conclusion of the entire simulation. The first type of print-out is useful when the simulation is run only for a few simulated days, and as much information as possible is desired. On an hourly basis, the average ready queue length, AQL, and the average response time of each type of search, A(1,..,5)RT, are made available. At the conclusion of each simulated day the following parameters are printed:

IPC - the cycle number of the simulation.

RT - the total time needed to process all the queries in the ready queue.

PT - the total time needed to process all the queries in the back-up queue.

TIT - the total idle time that has accumulated in the system as of that moment.

ST - the time needed to process the query in the central processor.

CL - the number of queries in the ready queue.

CM - the number of queries in the back-up queue.

The second type of print-out occurs at the end of each simulation, and is in the form of statistical calculations. The final lengths of the ready queue and the back-up queue are expressed as a mean value with a standard deviation for as many days as the simulation is run. Similar values are given for the total idle time and the average ready queue length. A difference exists between the final length of the ready queue and the average ready queue length in that the average ready queue length is the mean value on an hourly basis, while the final ready queue length is the mean on a daily basis.

V. - An Example: An Information Retrieval System Evaluation By Simulation

Until now, in this paper, little if any mention has been made about the characteristics or parameters of a specific information retrieval system with which any reader of this paper may be familiar. All the discussion has focused on the design of the simulator model (FOSSAL). To prove the validity and usefulness of FOSSAL, certan parameters have been picked and submitted to FOSSAL for evaluation. The majority of the parameters that are used are taken from the Center for the Information Sciences Experimental Document (5) Retrieval System at Lehigh University, and any deviations from this system are noted.

It is the purpose of this example to find the maximum conditions under which the system can operate with a given set of parameters. Of course, with the output information that is received from FOSSAL, other comments and observations can be made. A maximum condition of a system can be obtained by varying the condition that is under test and holding all the other conditions constant. Since this simulator has eight categories of variables, there are at least eight areas that can be observed, studied, and maximized. The variables and the values of them that are used in this evaluation example are listed in Figure 9.

The simulator is run to check two important points; (1) to prove that the simulator gives results comparable to those of an actual system, and (2) to see how many runs must be made to reach a stable or steady state condition. Since the simulation is written for one day periods, a stable condition of the simulator is

Variable Name	Actual Value	Comments
ISTART	35791	See Section IV.
AMIN	480 min.	The evaluation is run for a period of one working day, so 480 minutes is used.
KDAYS	30 days	The equivalent of 30 working days.
MSM	1 or 2	When an evaluation of a tape system is in progress, a 1 must be used; 2 for a disk system.
TH	30 min.	This value for the threshold was picked for purposes of evaluation.
EXM	0.2 queries per min.	This is equivalent to one arrival every five minutes. It is only one of the values used in the evaluation.
S1M	5.0 doc. per search	See Appendix 6.
S2M	3.0 doc. per search	See Appendix 6.
S3M	1.0 doc. per search	See Appendix 6.
S4M	1.0 doc. per search	See Appendix 6.
S5M	1.0 doc. per search	See Appendix 6.
HDM	2.0 min. per document	This value is assumed for purposes of evaluation.
BCPD	144 char. per document	This is an average value taken from the Lehigh C.I.S. System.
CTT	18 microsec.	Taken from G.E. 225 Computer.
DAT	225 millisec.	Taken from G.E. disk subsystem.
DDT	5 millisec.	Taken from G.E. disk subsystem.

Figure 9. List of Input Variables

Variable Name	Actual Value	Comments
DRW	8.4 microsec.	Taken from G.E. disk subsystem.
WPR	64 words	Taken from G.E. disk subsystem.
DDR	50 documents	Taken from Lehigh C.I.S. System.
DAD	25 documents	Taken from Lehigh C.I.S. System.
THS	15 kilocycles	Taken from G.E. tape subsystem.
TCM	4.0	Taken from G.E. tape subsystem.
TREC	2200 records	Taken from Lehigh C.I.S. System.
WPRT	256 words	Taken from Lehigh C.I.S. System.
TSST	0.012 sec.	Taken from G.E. tape subsystem.
ATTR	3 terms	Taken from Lehigh C.I.S. System.
BF	4	Taken from Lehigh C.I.S. System.
TR	150 bits/sec.	A common teletype speed.
BPC	8 bits per character	This value is assumed since ASCII Code uses an 8-bit code.
TNI1	400 instruc.	Assumed value for evaluation.
TNI2	450 instruc.	" "
TNI3	500 instruc.	" "
TNI4	550 instruc.	" "
TNI5	600 instruc.	" "
DNI1	400 instruc.	" "
DNI2	450 instruc.	" "
DNI3	500 instruc.	" "
DNI4	550 instruc.	" "
DNI5	600 instruc.	" "

Figure 9 (continued). List of Input Variables one in which the simulator can give a good representation of a day with no deviations occurring from that value. Thus, the simulator is run up to 30 days in an attempt to locate a maximum and a stable condition.

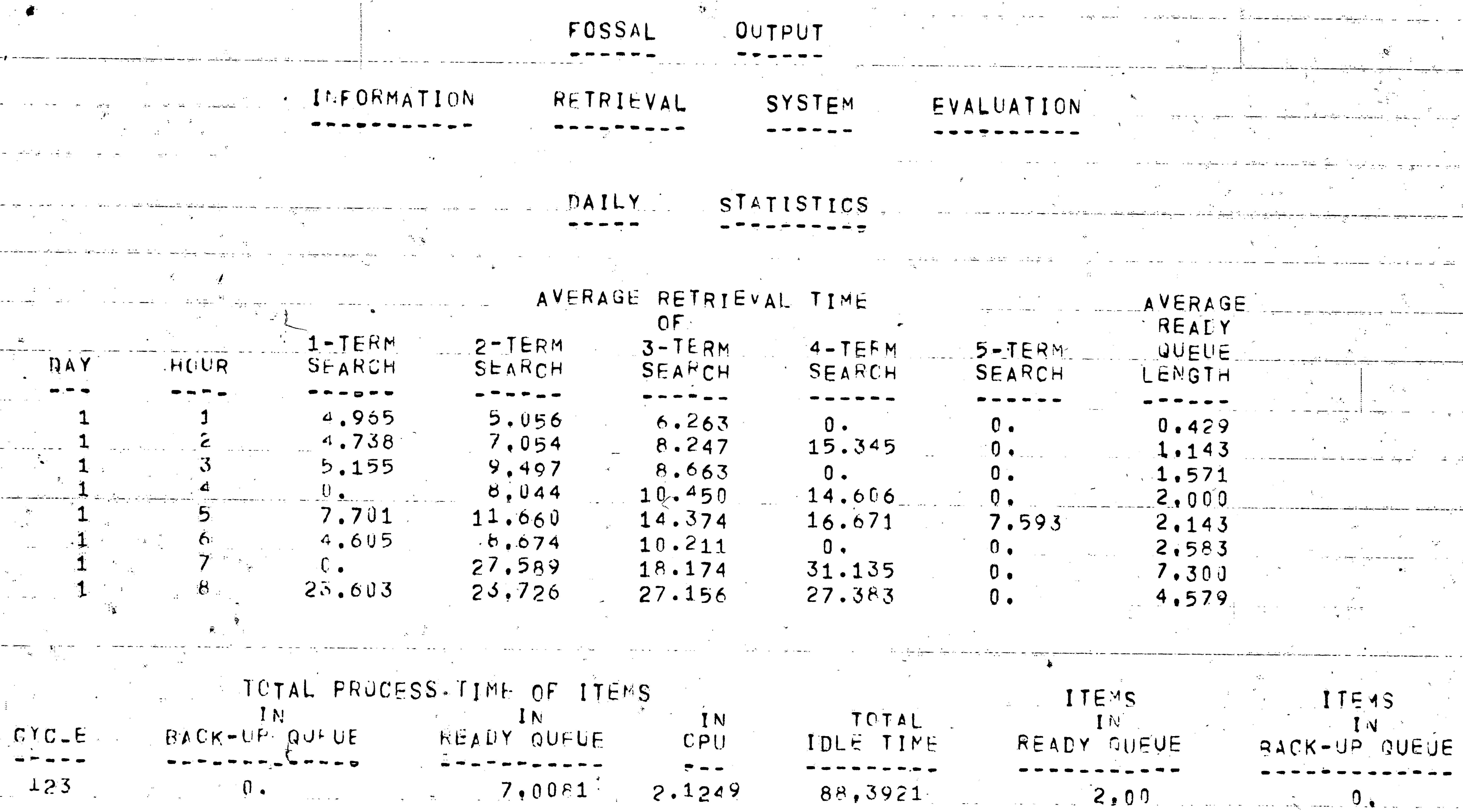
Once it is determined what the number of cycles necessary to obtain significant results is, the simulator is run with different arrival rates to check the effect of a varying λ on the system. All tests are performed using the tape subsystem.

For purposes of validating the simulator, and showing the power of its evaluation technique, it is felt that the above tests briefly described are sufficient. However, there is no limitation to the thoroughness that can be obtained in an evaluation by varying all the parameters used in FOSSAL.

VI. - Discussion of Results

As has been mentioned previously, the purpose of this research is to develop a simulation model that can be used to evaluate any on-line information retrieval system, rather then just generate facts about a specific system. However, one of the ways that a simulation program can be checked, is to compare the results of the program with known values. Where possible, this discussion will make comparisons between the simulated output and actual values.

Figure 10 shows the output of FOSSAL as it appears on an hourly basis. As previously mentioned, this output is optional, and would probably only be used when the simulation is run for a short period of time. These particular values are taken from what proved to be an acceptable arrival rate with respect to the system parameters of the simulator. It should be kept in mind that all the results discussed in this section are for the tape subsystem. The first two columns in Figure 10 show the specific time cycle for which the results apply. The next five columns give the average response time that each type of search would encounter during the designated time cycle. This average response time includes the time a query waits in the queue, the process time, the time the retrieved document information waits in the output queue, and the time to transmit the information to the user. For example, consider the average response time of a one-term search during the first day, first hour. The simulator produces a value of 4.965. When using the Lehigh C.I.S. retrieval system, 3.0 minutes is needed to search and process the serial file on tape, and additional time is needed to transmit whatever retrieved information

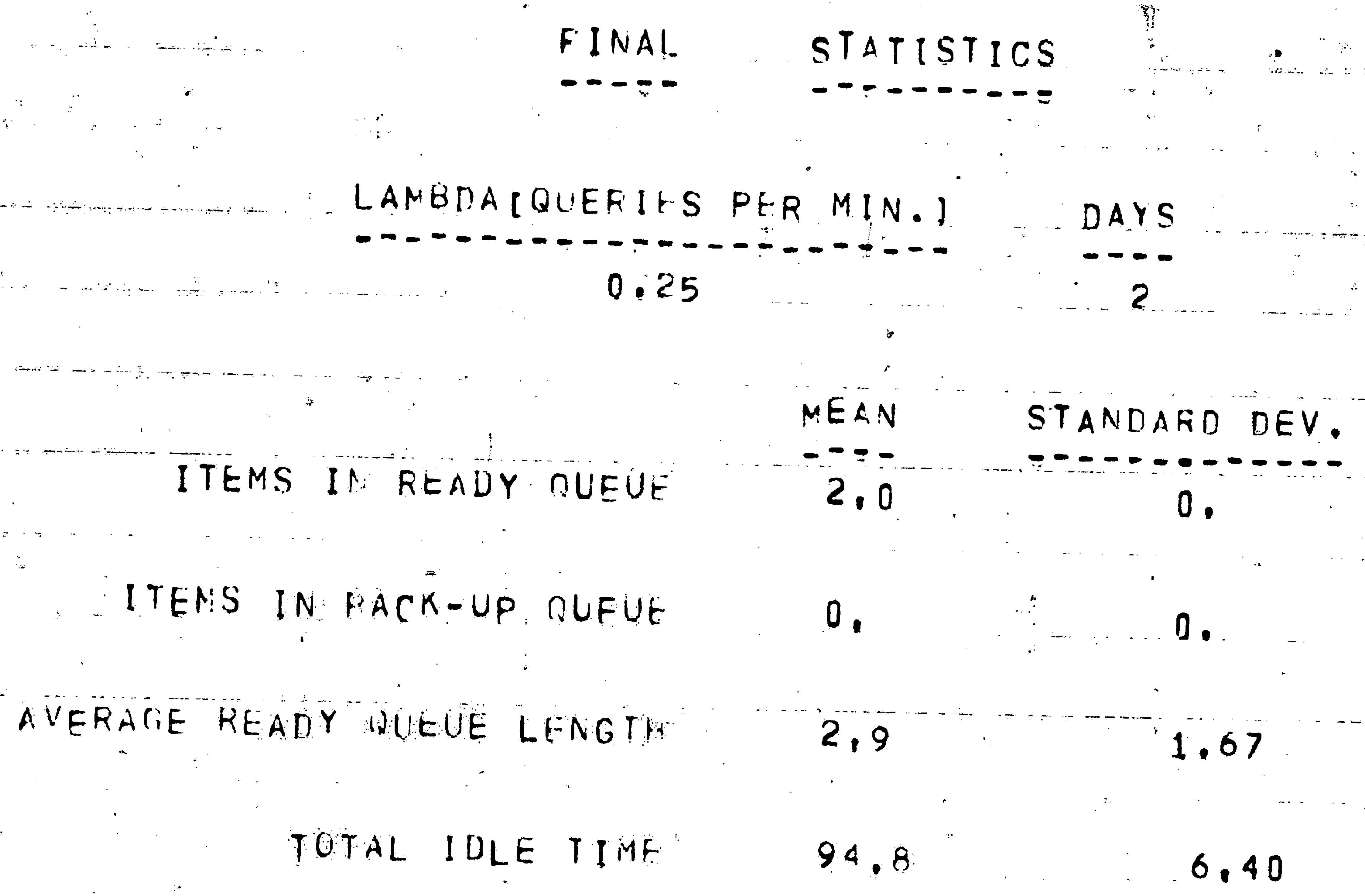


```
                        FOSSAL    OUTPUT
                        ------    ------

        INFORMATION     RETRIEVAL     SYSTEM     EVALUATION
        -----------     ---------     ------     ----------

                        DAILY     STATISTICS
                        -----     ----------

                            AVERAGE RETRIEVAL TIME                      AVERAGE
                                         OF                              READY
                    1-TERM     2-TERM     3-TERM     4-TERM     5-TERM    QUEUE
 DAY      HOUR      SEARCH     SEARCH     SEARCH     SEARCH     SEARCH   LENGTH
 ---      ----      ------     ------     ------     ------     ------   ------
   1         1       4.965      5.056      6.263      0.         0.       0.429
   1         2       4.738      7.054      8.247     15.345      0.       1.143
   1         3       5.155      9.497      8.663      0.         0.       1.571
   1         4       0.         8.044     10.450     14.606      0.       2.000
   1         5       7.701     11.660     14.374     16.671      7.593    2.143
   1         6       4.605      6.674     10.211      0.         0.       2.583
   1         7       0.        27.589     18.174     31.135      0.       7.300
   1         8      23.603     23.726     27.156     27.383      0.       4.579

            TOTAL PROCESS TIME OF ITEMS                           ITEMS          ITEMS
                IN              IN            IN      TOTAL         IN             IN
 CYCLE    BACK-UP QUEUE    READY QUEUE       CPU    IDLE TIME   READY QUEUE   BACK-UP QUEUE
 -----    -------------    -----------       ---    ---------   -----------   -------------
  123          0.             7.0081      2.1249     88.3921        2.00            0.
```

Figure 10. FOSSAL Output - Daily Statistics (λ= .25)

is obtained. Thus the difference between the 4.965 minutes, average response time, and the 3.0 minutes for search-process time, is the time that the query waits in the ready queue plus the time to transmit the retrieved documents to the user via the teletype unit. By observing the average response time for the other time cycles of the first day, it is apparent that the response time increases proportionally with the average queue length. In all cases during the first day, the average response time of a two-term search is greater than that of a one-term search. This is due to the fact that it takes longer to perform a two-term search, and does not represent the fact that a priority is given to a one-term search. The values for the average response times and the average ready queue lengths are reset at the beginning of each hour, so the values depicted in Figure 10 are not biased by any previous values. Such a precaution is necessary, because if there is no queue at the beginning of the day, and a large one at the end of the day, the average for that day would not indicate the extremes for such a large time period. To offset this problem, the average values were taken on an hourly basis.

At the bottom of Figure 10 seven variables are printed which depict the condition of the simulator at the end of the day. The first variable, cycle, gives the number of queries that arrived during the day. Thus, at the end of the first day, 123 queries have arrived, two items are in the ready queue, and no items are in the back-up tape queue. The total time needed to process those two queries in the ready queue is listed as the third variable at the bottom of Figure 10. The fourth variable indicates the amount



```
                              FINAL    STATISTICS
                              -----    ----------

                    LAMBDA[QUERIES PER MIN.]    DAYS
                    ------------------------    ----
                              0.25                2

                                    MEAN     STANDARD DEV.
                                    ----     -------------
        ITEMS IN READY QUEUE         2.0          0.

      ITEMS IN BACK-UP QUEUE         0.           0.

  AVERAGE READY QUEUE LENGTH         2.9          1.67

             TOTAL IDLE TIME         94.8         6.40
```

Figure 11. FOSSAL Output - Final Statistics ($\lambda = .25$)

of time needed to process the query that is in the CPU. However, one must remember that the time listed under "Total Process Time of Items in CPU" (2.1249) is the time needed to complete the processing of a particular item, and is not the time that is needed from start to finish, since part of the query may have been processed during the previous cycle. The total idle time listed in Figure 10 is the cumulative time since the beginning of the day that the CPU is idle.

The second type of output is shown in Figure 11, and is always obtained at the end of a simulation run. It is designed as a summary statement to give a minimum of information so that print-out time can be kept very low on long simulation runs. A simulation that is run for 30 days takes approximately 15 minutes of computer time on the G.E. 225. The second line of Figure 11 explains that the arrival rate, λ, is equal to 0.25 queries per minute, and that the simulation statistics are compiled for a period of 2 days. For the two day period, the final number of items in the ready queue has a mean of 2.0 and standard deviation of 0.0. The average ready queue length has a mean of 2.9 with $\sigma = 1.67$. This value is the number of queries that could be expected in the queue during any hour of the day. Since the final length of the queue is less than the average queue length, it is obvious that the maximum queue appears at some time other than at the end of the day. When reference is made to Figure 10, one notices approximately when the ready queue reaches its maximum by reading the average ready queue length for each hour of the day that the simulation is run.

It has been mentioned that one of the aims of this research

$\lambda = 0.20$				$\lambda = 0.25$				
Final Ready Queue Length, RQ		Average Ready Queue Length, AQL		Final Ready Queue Length, RQ		Average Ready Queue Length, AQL		
MEAN	Standard Dev.	MEAN	Standard Dev.	MEAN	Standard Dev.	MEAN	Standard Dev.	Days
2.0	2.0	2.5	1.63	2.0	0	2.9	1.67	2
1.4	1.74	1.9	1.23	3.4	3.07	3.3	1.95	5
0.9	1.62	1.5	1.6	2.9	3.02	3.3	2.4	20
0.8	1.34	1.4	1.35	3.6	3.42	3.9	2.94	30

Figure 12. Effect of Time on Queue Length

was to find out how many cycles of the simulation must be run to get a true picture. The results of this experiment are shown in Figure 12. The two arrival rates used, λ= 0.20 and λ= 0.25, seemed to be quite adaptable to the parameters of the system discussed earlier. When λ=0.20, the queue length decreases as the simulation is run for a greater number of days, so in this case, the average queue length and the final queue length appear to be at a maximum when the simulation is run only for two days; however, when λ= 0.25, the queue length increases as the simulation time increases. Thus, for the system parameters used, an arrival rate of 0.25 queries per minute seems to put too much of a load on the system if a zero queue is desired at the end of the day.

In Figure 13 the effect of the arrival rate, λ, on the queues is seen. It is quite obvious that as the arrival rate increases the system has a harder time of handling the queries, so the queue lengths increase. From such a curve a designer can tell management what the expected queue is at the end of the day, and management can then decide whether overtime is necessary to work off the backlog, or whether it is more desirable to restrict the rate of arrivals to the system.

Figure 14 shows the idle time on a daily basis for the same arrival rates shown in Figure 13. Thus, a designer may want to make a trade-off between idle time and final queue length in an attempt to obtain the best possible system. All the data used in Figures 13 and 14 was obtained from 30 daily cycles of the simulator. Although the idle time values in Figure 14 appear to be quite large, one must remember that in many instances the idle

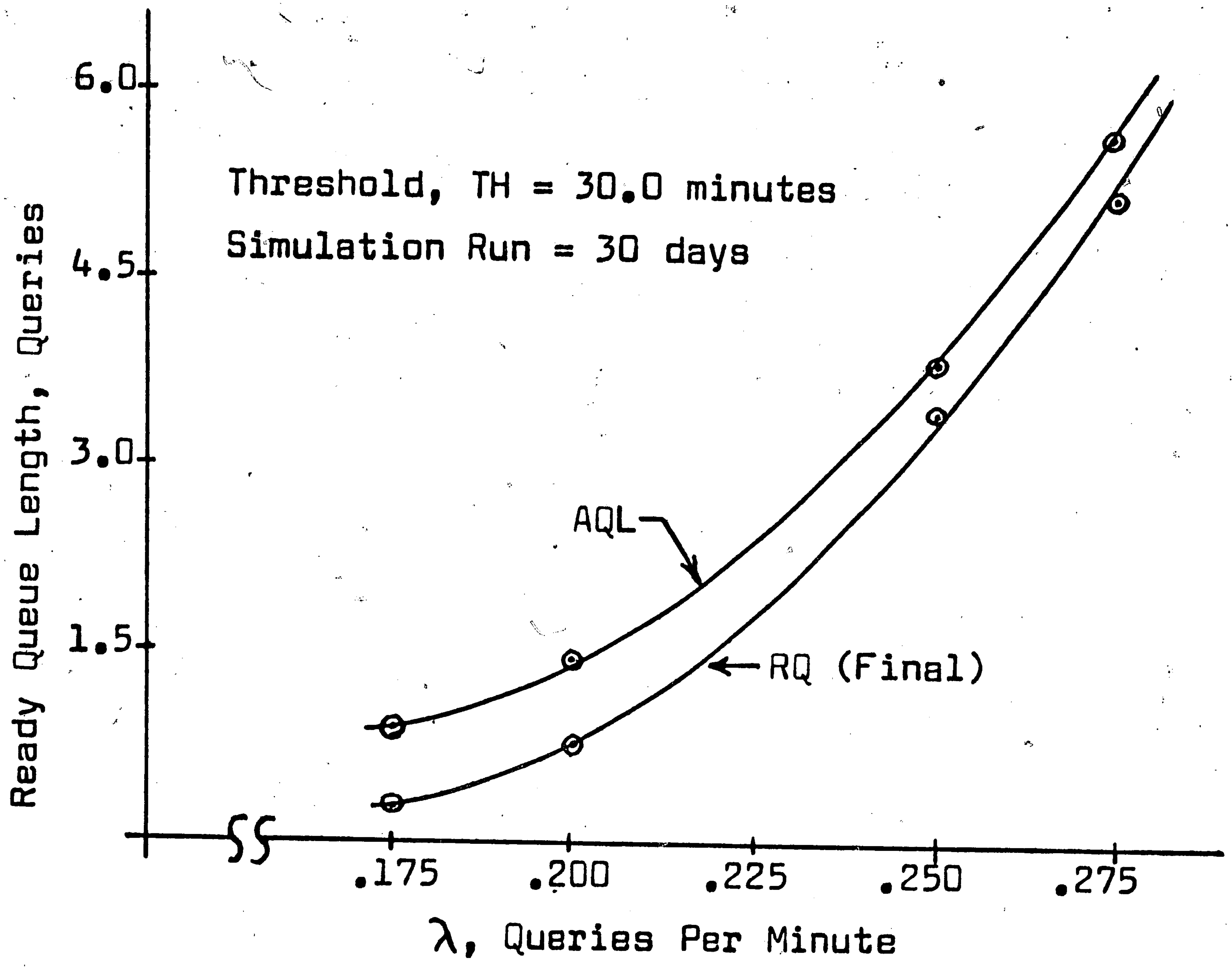


Figure 13. Effect of Arrival Rate on Queue Length

λ (Queries Per Min.)	Total Idle Time	
	MEAN	Standard Deviation
0.175	208.5 min./day	39.26 min./day
0.200	162.0 min./day	36.57 min./day
0.250	73.5 min./day	35.19 min./day
0.275	45.8 min./day	31.12 min./day

Figure 14. Effect of Arrival Rate on Total Idle Time

time accumulates in the beginning of the day when the queues are empty.

VII. - Conclusions

The use of the simulation model, FOSSAL, has proven to be useful in evaluating certain parameters of an on-line real-time information retrieval system which uses a coordinate index. Specifically, in this research the effect that the parameters of a system had on the final and average ready queue length was studied. Although only the arrival rate and number of days were varied, there are many more combinations of variables that can be studied with FOSSAL. Since there are over 30 variable input parameters, this presents a great number of combinations by which an IR system can be evaluated.

With regard to the model itself, the point must be made that any one who uses FOSSAL must be sure that the distributions of the retrieved documents are the same as those in the simulation model. In Appendix 6 it is stated that the distributions were all Poisson distributions, and that the means of their distributions are variable input parameters. If the user of FOSSAL does not feel that the distributions match the ones of the system being evaluated, then slight changes would have to be made in the program.

Even though this simulation model did not make use of time-sharing techniques, the model can lend aid in evaluating less sophisticated systems.

Appendix 1 - Derivation of Disk Retrieval Equation

The disk access time has been stated by Scheer (6) to be the sum of the arm positioning time and rotational delay time. Also, G.E. (7) lists the disk retrieval time to be that of the sum of total access time, latency time, and read/write time. It appears that a close relationship exists between the different disk systems, and that a generalized equation can be derived to calculate the retrieval time for records from a serial and inverted file.

Let: DAT = disk access time in milliseconds,
DDT = disk delay time after positioning in millisec.,
DRW = disk read/write speed in microseconds per word,
WPR = words per disk record.

The read/write time per record is:

$$\text{TRWR} = (\text{DRW})(\text{WPR}) \text{ microseconds/record} \qquad \text{Eq.A1.1}$$

The time needed for disk command available is:

$$\text{DCA} = (\text{DAT} + \text{DDT})10^3 \text{ microseconds} \qquad \text{Eq.A1.2}$$

The average number of records that each user has for a characteristic on the inverted file is:

$$\text{UR} = \frac{\text{DAD}}{\text{DDR}} \text{ records} \qquad \text{Eq.A1.3}$$

where DDR is the number of documents held by one record on the disk, and DAD is the average number of documents for each term of the inverted file on the disk. This equation includes the use of chaining of records on the inverted file, since if DAD is large compared with the physical record of the disk, the value of UR will be larger than 1, indicating that more than one seek is necessary to retrieve all the documents indexed under that characteristic. However, UR can never be smaller than one; so if DAD DDR, a value of one is assigned to UR.

Combining equations A1.1 and A1.2, the total time to retrieve the characteristics and their related items from the inverted file is obtained:

$$D1 = (PR)(TOS)(TRWR + DCA) \text{ microseconds} \qquad \text{Eq.A1.4}$$

where the physical record, $PR = \begin{cases} 1 \text{ if } UR < 1 \\ UR \text{ if } UR \geq 1 \end{cases}$

and TOS is the number of terms in the Boolean "AND" request.

Once an index term and it's document numbers are retrieved from the inverted file, a comparison is made to locate all relevant documents. The computer time required to perform the needed comparisons on each record retrieved from the inverted file varies for each situation, thus the following parameters are introduced:

DNI1 = the number of machine instructions necessary to complete a one-term Boolean search after the term is retrieved from the inverted file on the disk.

DNI2 = instructions for a two-term Boolean "AND" search, etc.

DNI3 = instructions for a three-term Boolean "AND" search, etc.

DNI4 = instructions for a four-term Boolean "AND" search, etc.

DNI5 = instructions for a five-term Boolean "AND" search, etc.

CMIT = computer time needed to perform 1 machine instruction.

The total time needed to perform the comparisons and the related programming steps is:

$$D2 = (CMIT)(DNI(1,..,5))(DAD)(TOS) \text{ microseconds} \qquad \text{Eq.A1.5}$$

Once the relevant document numbers are produced, the bibliographic document information must be retrieved from the serial file, also stored on the disk. Let W equal the number of relevant documents; then the time to search and retrieve the information is obtained by combining Eqs. A1.1 and A1.2 and multiplying by W.

Thus,

$$D3 = (W)(TRWR + DCA) \text{ microseconds} \qquad \text{Eq.A1.6}$$

The total time needed for the retrieval of all the relevant document information is given by:

$$PRT = (D1 + D2 + D3)(10^{-6}/60) \text{ minutes} \qquad \text{Eq.A1.7}$$

The value of PRT includes all the time needed to retrieve the document information up to the point that the information is to be transferred to the output queue. Any further time needed for manipulation or code conversion is handled in the output queue.

Appendix 2 - Derivation of Tape Retrieval Equation

Before attempting to derive an equation for the retrieval time, the scheme of the retrieval mechanism must be made clear. The search that is performed on the tape system is executed on one serial file; the file contains the item numbers and the index terms that characterize it, plus all the necessary bibliographic data. When the Boolean request is read into the CPU, the retrieval phase starts by reading each item from the serial file until a match is found. The matching is executed in the central processing unit. When a match is found, the pertinent bibliographic data is transferred from the central processing unit to the output queue and on to the user.

Let: TCM = character mode of tape (i.e., 3 = binary coded decimal; 4 = binary),
TREC = number of records on tape,
WPRT = words per record,
THS = tape transfer rate in kilocycles,
TSST = tape start-stop time in seconds.

These terms can be combined to obtain the time needed to read a record (8):

$$RT = \frac{(TCM)(TREC)(WPRT)}{THS} + (TREC - 1)(TSST) \quad \text{sec.} \qquad \text{Eq.A2.1}$$

However, equation A2.1 does not take into consideration the use of a blocking factor. To apply the equation to a system with blocked records, the start-stop time (TSST) is the only part of the equation that is affected. Thus the time to read the entire serial file stored on tape is:

$$RSFT = \frac{(TCM)(TREC)(WPRT)}{THS} + \frac{(TREC - 1)(TSST)}{BF} \quad \text{sec.} \qquad \text{Eq.A2.2}$$

When applying this equation to a specific problem, one must keep in mind whether double words are used in the tape record.

If double words are used, then the value of WPRT is twice as large as a single word.

Once the record is placed in the processor, the search program is ready to check for a match with the request. The time to perform the checks and comparisons on each record retrieved from the serial file varies for different Boolean searches and also varies due to the fact that no two programmers think alike when writing the coding for a program. To permit the simulation to have universal usage, the time to perform these comparisons are broken down into the sum of the machine instructions of the individual program. When each document is read from the serial file, there are a certain number of terms characterizing the document which are contained in this same record. The number of terms per document is not a variable in this simulation, but is an average value. The number of terms per document is needed so that the process time in the CPU can be calculated.

Let: TNI1 = the number of machine instructions necessary to complete a one-term Boolean search after the term is retrieved from the serial file on the tape.

TNI2 = instructions for a two-term Boolean "AND" search.

TNI3 = instructions for a three-term Boolean "AND" search.

TNI4 = instructions for a four-term Boolean "AND" search.

TNI5 = instructions for a five-term Boolean "AND" search.

CMIT = computer time needed to perform one machine instruction.

ATTR = average number of terms in a tape record of the serial file divided by 2.

The total time needed to perform the comparisons and the related programming steps is:

$$CT = (CMIT)(TNI(1,..,5))(ATTR)(TREC)\ 10^{-6}\ \text{sec.} \qquad \text{Eq.A2.3}$$

Since the bibliographic information for each document is read from the tape in the same record as the document characteristics, no additional read time is needed when a document is found that is relevant to the search request.

The total time needed to retrieve and process the documents from a serial file on tape is obtained by combining equations A2.2 and A2.3; thus:

$$PRT = \frac{(CT + RSFT)}{60}\ \text{minutes} \qquad \text{Eq.A2.4}$$

The assumption is made that if the document collection is stored on two tapes, the transition from one tape handler to another is executed without any lapse in time other than the inter-record gap start-stop time. The value of PRT includes all the time needed to retrieve the document information up to the point that the information is to be transferred to the output queue. Any further time needed for manipulation or code conversion is handled in the output queue.

Appendix 3 - Calculation of New Arrival

The purpose of this particular subroutine is to determine which Boolean combination is used in the new query entering the system, and to find out how many documents will satisfy the query. This information is combined and used to determine the response time whether it be by the disk or tape subsystem.

When the subroutine NEWARR is called by the main program, the initial step of NEWARR is to determine which Boolean search is used. A random number is taken from a uniform distribution to determine the Boolean request (the percentages are discussed in Section III. B). Once the type of Boolean request, TOS, is determined, the mean number of retrieved documents for that particular search, S(1,..,5)M, is used to determine the exact number of documents by taking a sample from the Poisson distribution for the documents (see Appendix 6). A special subroutine, POISSN, is called which takes a uniformly distributed random number and transforms it to a random variable with a Poisson distribution. The program for this subroutine (9) is listed in Appendix 5.

Now that the type of Boolean search, TOS, and the number of retrieved documents, X, is calculated, the only remaining function of the subroutine is to combine them in the following manner:

$$QA = TOS + \frac{X}{1000} \qquad \text{Eq.A3.1}$$

The decimal point of X is moved three places to make it's value less than 1. The value of QA is returned to the main program for use by the tape or disk subroutine.

Appendix 4 - Holding Time and Transmission Equations

There are three functions in the system that comprise the total response time of any user who enters the system, and the subroutine, OUTBUR, is called to combine these functions into one value for a user's total response time.

The first functional component of the total response time is the query waiting time, QWT. This is the time that a query waits in the back-up and ready queues, plus the time to retrieve the documents from storage and process them in the CPU. The value of QWT is calculated in either the tape or disk subroutine under the name, PRT; and is transferred eventually to the OUTBUR subroutine for further calculations.

The functional component, output queue holding time, is the second part of the total response time. This is the time that the retrieved documents wait in the output queue before the complete transfer to the output buffer. As discussed in Section III.D, the holding time is exponentially distributed. The value for the holding time, HT, is obtained by sampling from the random number generator, Y, and applying to equation A4.1, which produces a random number exponentially transformed with a mean of HDM.

$$HT = \frac{1}{HDM}(LOGF(Y)) \text{ minutes} \qquad \text{Eq.A4.1}$$

The mean, HDM, is a variable which is adjusted to meet the needs of a particular environment.

The final functional component of the total response time is the transfer and transmission time. Figure 15 shows the areas covered by transfer and transmission time by an asterisk (*). This is the time to transmit all retrieved document information to the user, plus the time involved in transferring the information from the CPU to the buffer.

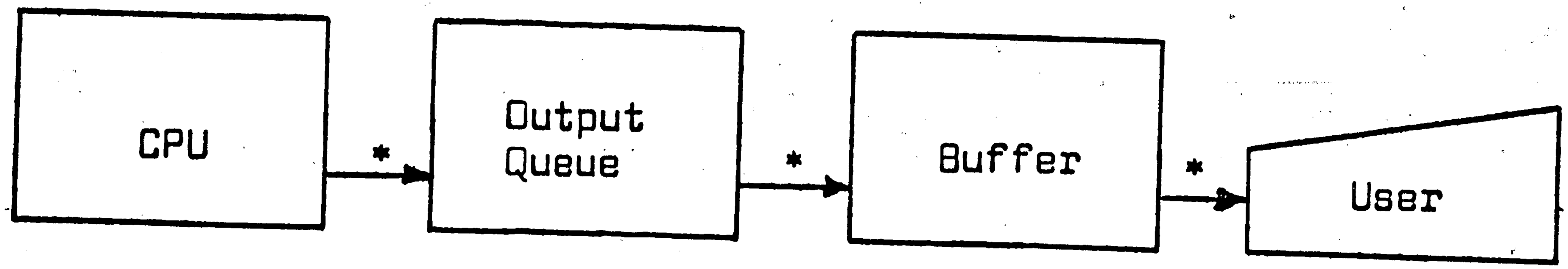


Figure 15. Flow of Data from CPU

The transfer time per document, TRTPD, is calculated (Eq.A4.2) by combining the number of characters per retrieved document, BCPD, and the transfer time per character, CTT, for the system under evaluation. The factor of 2 enters the equation since the data is transferred into and out of the output queue.

$$\text{TRTPD} = (\text{BCPD})(\text{CTT})(2)10^{-3} \text{ seconds/document} \qquad \text{Eq.A4.2}$$

The transmission time per document, TTPD, is obtained by using equation A4.3, where,

BPC = the number of bits per character
TR = transmission rate

$$\text{TTPD} = \frac{(\text{BCPD})(\text{BPC})}{\text{TR}} \text{ seconds/document} \qquad \text{Eq.A4.3}$$

Equations A4.2 and A4.3 are combined and increased by the number of retrieved documents, as shown in Eq.A4.4.

$$\text{UTT} = \frac{(\text{TTPD} + \text{TRTPD})}{60} (\text{W}) \text{ minutes} \qquad \text{Eq.A4.4}$$

W = number of retrieved documents
UTT = transfer-transmission time

The three functional components are combined to obtain QRT, the total time that a user must wait from the time of entry to receipt of all document information. Thus,

$$\text{QRT} = (\text{QWT} + \text{HT} + \text{UTT}) \text{ minutes} \qquad \text{Eq.A4.5}$$

Appendix 5 - Fortran II Simulation Program

```
      COMMON W,PRT,RQ[100],OET[100],TPE[100],TEI[50],
     1TPET[50],PQ[50]
C     READ VARIABLE PARAMETER CARDS
 1010 READ 1000,ISTART,AMIN,KDAYS
 1000 FORMAT[I5,5X,F6.1,4X,I2]
      IF[ISTART]998,1009,1009
 1009 READ 1001,EXM
 1001 FORMAT[F6.4]
      READ 1002,HDM,CTT,TR,BPC
 1002 FORMAT[F6.4,4X,F6.2,4X,2[F5.0,5X]]
      READ 1003,MSM,TH,BCPD
 1003 FORMAT[I1,9X,F5.1,5X,F5.1]
      READ 1004,S1M,S2M,S3M,S4M,S5M
 1004 FORMAT[5[F6.3,4X]]
      READ 1005,DRW,DAT,DDT,DDR,DAD,WPR
 1005 FORMAT[6[F6.1,4X]]
      READ 1006,THS,TCM,TREC,WPRT,ISST,ATTR,BF
 1006 FORMAT[4[F6.1,4X],F6.3,4X,2[F6.1,4X]]
      READ 1007,TNI1,TNI2,TNI3,TNI4,TNI5
 1007 FORMAT[5[F6.1,4X]]
      READ 1008,DNI1,DNI2,DNI3,DNI4,DNI5
 1008 FORMAT[5[F6.1,4X]]
C   INITIALIZATION OF PARAMETERS
      KD=0
      SUMCM=0.0
      SUMCL=0.0
      SUMAQL=0.0
      SUMTIT=0.0
      SSQCM=0.0
      SSQCL=0.0
      SSQAQL=0.0
      SSQTIT=0.0
  999 CONTINUE
      DO 997 I=1,452
  997 PQ[I]=0
      KD=KD+1
      IPC=0
      ATT=0.0
      ART=0.0
      RT=0.0
      ST=0.0
      SWT=0.0
      QRT=0.0
      QWT=0.0
      TIT=0.0
      STP=0.0
      T1S=0.0
      T2S=0.0
      T3S=0.0
      T4S=0.0
      T5S=0.0
      T1RT=0.0
      T2RT=0.0
      T3RT=0.0
      T4RT=0.0
      T5RT=0.0
      K=1
      L=1
      M=1
      CAT=0.0
      H=0.0
      TCL=0.0
      TPEP=0.0
      PT=0.0
      QA=0.0
      W=0.0
      TOS=0.0
      PRT=0.0
      LP=1
      NP=1
      KH=0
      PRINT 417
C   GENERATE INTERARRIVAL TIME
    1 Y=FLAT[ISTART]
      KEY=0
      U=[[1.0/EXM]*LOGF[Y]]*[-1.0]
      ATT=ATT+U
      ART=U
      SWT=RT+ST
C    CHECK SYSTEM
      IF[ART-SWT]2,2,3
    3 TIT=[ART-SWT]+TIT
    2 IF[ST]300,300,4
    4 IF[ART-ST]5,6,6
C   UPDATE CENTRAL PROCESSOR
    5 ST=ST-ART
      ART=0.0
      GO TO 200
    6 ART=ART-ST
      ST=0.0
  100 CONTINUE
C   UPDATE OUTPUT QUEUE AND BUFFER
      CALL OUTBUR[HDM,Y,TPEP,CTT,TR,QWT,QRT,STP,BPC,BCPD]
      H=H+1.0
      IF[STP-1.0]8,8,9
    8 T1S=T1S+1.0
      T1RT=T1RT+QRT
      GO TO 16
    9 IF[STP-2.0]10,10,11
   10 T2S=T2S+1.0
```

```
      T2RT=T2RT+QRT
      GO TO 16
   11 IF[STP-3.0]12,12,13
   12 T3S=T3S+1.0
      T3RT=T3RT+QRT
      GO TO 16
   13 IF[STP-4.0]14,14,15
   14 T4S=T4S+1.0
      T4RT=T4RT+QRT
      GO TO 16
   15 T5S=T5S+1.0
      T5RT=T5RT+QRT
   16 IF[KEY]17,17,25
   17 IF [ART]200,200,18
C     UPDATE READY QUEUE
   18 IF[RQ[K]]300,300,19
   19 IF[ART-RQ[K]]20,21,21
   20 RQ[K]=RQ[K]-ART
      RT=RT-ART
      ART=0.0
      GO TO 200
   21 RT=RT-RQ[K]
      ART=ART-RQ[K]
      QWT=ATT-QET[K]-ART
      TPEP=TPE[K]
      KEY=1
      GO TO 100
   25 J=2
      DO 50 I=1,99
      RQ[I]=RQ[J]
      QET[I]=QET[J]
      TPE[I]=TPE[J]
   50 J=J+1
      L=L-1
      GO TO 17
C     UPDATE BACK-UP QUEUE
  200 IF[RT-TH]201,300,300
  201 IF[PT]300,300,202
  202 RQ[L]=PQ[1]
      QET[L]=TET[1]
      TPE[L]=TPET[1]
      RT=RT+RQ[L]
      PT=PT-PQ[1]
      L=L+1
      J=2
      DO 205 I=1,49
      PQ[I]=PQ[J]
      TET[I]=TET[J]
      TPET[I]=TPET[J]
  205 J=J+1
```

```
      M=M-1
      GO TO 200
  300 CONTINUE
C     GENERATION OF A NEW ARRIVAL
      CALL NEWARR[QA,Y,S1M,S2M,S3M,S4M,S5M,TOS]
C     RETRIEVAL AND PROCESSING TIME
      IF[MSM-1]500,500,501
  500 CONTINUE
      CALL TAPEST[TCM,TREC,WPRT,THS,TSST,BF,TNI1,TNI2,
     1TNI3,TNI4,TNI5,CTT,TOS,ATTR]
      GO TO 400
  501 CONTINUE
      CALL DISKST[DRW,WPR,DAT,DDT,DAD,DDR,TOS,DNI1,
     1DNI2,DNI3,DNI4,DNI5,CTT]
C     PLACEMENT OF NEW ARRIVAL
  400 IF[L-101]420,420,403
  420 IF[RT-TH]401,402,402
  402 IF[QA-3.0]401,403,403
  403 PQ[M]=PRT
      TET[M]=ATT
      TPET[M]=QA
      M=M+1
      PT=PT+PRT
      GO TO 404
  401 RQ[L]=PRT
      QET[L]=ATT
      TPE[L]=QA
      L=L+1
      RT=RT+PRT
  404 IF[ST]405,405,406
  405 IF[RQ[K]]406,406,407
C     TRANSFER OF QUERY TO CPU
  407 ST=RQ[K]
      QWT=[ATT-QET[K]+ST]
      TPEP=TPE[K]
      RT=RT-RQ[K]
      J=2
      DO 408 I=1,99
      RQ[I]=RQ[J]
      QET[I]=QET[J]
      TPE[I]=TPE[J]
  408 J=J+1
      L=L-1
  406 CL=L-1
      CM=M-1
      HAT=ATT-CAT
      TCL=TCL+CL
      IF[IPC]480,480,481
  480 CONTINUE
      PRINT 1050
      PRINT 1051
      PRINT 1052
      PRINT 1053
      PRINT 1070
      PRINT 1071
      PRINT 1060
      PRINT 1061
      PRINT 1062
      PRINT 1063
      PRINT 1064
  481 CONTINUE
      IPC=IPC+1
      IF[HAT-60.0]1,412,412
  412 P1RT=T1S/H
      P2RT=T2S/H
      P3RT=T3S/H
      P4RT=T4S/H
      P5RT=T5S/H
      AQL=TCL/H
      A1RT=T1RT/T1S
      A2RT=T2RT/T2S
      A3RT=T3RT/T3S
      A4RT=T4RT/T4S
      A5RT=T5RT/T5S
      KH=KH+1
      CAT=HAT+CAT
      H=0.0
      PRINT 413,KD,KH,A1RT,A2RT,A3RT,A4RT,A5RT,AQL
      TCL=0.0
      T1RT=0.0
      T2RT=0.0
      T3RT=0.0
      T4RT=0.0
      T5RT=0.0
      T1S=0.0
      T2S=0.0
      T3S=0.0
      T4S=0.0
      T5S=0.0
      IF[ATT-AMIN]1,422,422
  422 CONTINUE
      PRINT 1054
      PRINT 1055
      PRINT 1056
      PRINT 1057
      PRINT 410,IPC,PT,RT,ST,TIT,CL,CM
      SUMCM=SUMCM+CM
      SUMCL=SUMCL+CL
      SUMAQL=SUMAQL+AQL
      SUMTIT=SUMTIT+TIT
      SSQCM=SSQCM+[CM**2.0]
      SSQCL=SSQCL+[CL**2.0]
      SSQAQL=SSQAQL+[AQL**2.0]
      SSQTIT=SSQTIT+[TIT**2.0]
      IF[KDAYS-KD]430,430,999
  430 CONTINUE
      D=KD
      CLM=SUMCL/D
      CMM=SUMCM/D
      AQLM=SUMAQL/D
      TITM=SUMTIT/D
      SIGCM=SQRTF[[SSQCM/D]-[CMM**2.0]]
      SIGCL=SQRTF[[SSQCL/D]-[CLM**2.0]]
      SIGAQL=SQRTF[[SSQAQL/D]-[AQLM**2.0]]
      SIGTIT=SQRTF[[SSQTIT/D]-[TITM**2.0]]
      PRINT 417
      PRINT 1075
      PRINT 1076
      PRINT 1077
      PRINT 1078
      PRINT 1079,EXM,KDAYS
      PRINT 1080
      PRINT 1081
      PRINT 1082,CLM,SIGCL
      PRINT 1083,CMM,SIGCM
      PRINT 1084,AQLM,SIGAQL
      PRINT 1085,TITM,SIGTIT
      PRINT 417
      GO TO 1010
  417 FORMAT[1H1]
 1050 FORMAT[50X,6HFOSSAL,5X,6HOUTPUT,53X]
 1051 FORMAT[50X,6H------,5X,6H------,53X,//]
 1052 FORMAT[33X,11HINFORMATION,5X,9HRETRIEVAL,5X,6HSYSTEM,
     15X,10HEVALUATION,36X]
 1053 FORMAT[33X,11H-----------,5X,9H---------,5X,6H------,5X,
     110H----------,36X,///]
 1070 FORMAT[50X,5HDAILY,5X,10HSTATISTICS]
 1071 FORMAT[50X,5H-----,5X,10H----------,///]
 1060 FORMAT[48X,22HAVERAGE RETRIEVAL TIME,18X,7HAVERAGE]
 1061 FORMAT[56X,2HOF,31X,5HREADY]
 1062 FORMAT[33X,6H1-TERM,5X,6H2-TERM,5X,6H3-TERM,5X,6H4-TERM,
     15X,6H5-TERM,6X,5HQUEUE]
 1063 FORMAT[16X,3HDAY,5X,4HHOUR,5X,6HSEARCH,5X,6HSEARCH,5X,
     16HSEARCH,5X,6HSEARCH,5X,6HSEARCH,5X,6HLENGTH]
 1064 FORMAT[16X,3H---,5X,4H----,5X,6H------,5X,6H------,5X,
     16H------,5X,6H------,5X,6H------,5X,6H------]
  413 FORMAT[17X,I2,6X,I2,6X,5[F6.3,5X],F6.3]
 1054 FORMAT[///,29X,27HTOTAL PROCESS TIME OF ITEMS,27X,5HITEMS,12X,
     15HITEMS]
 1055 FORMAT[30X,2HIN,15X,2HIN,10X,2HIN,8X,5HTOTAL,
     111X,2HIN,15X,2HIN]
1056 FORMAT[14X,5HCYCLE,5X,13HBACK-UP QUEUE,5X,11HREADY QUEUE,
     15X,3HCPU,5X,9HIDLE TIME,5X,11HREADY QUEUE,
     25X,13HBACK-UP QUEUE]
1057 FORMAT[14X,5H-----,5X,13H-------------,5X,11H-----------,
     15X,3H---,5X,9H---------,5X,11H-----------,
     25X,13H-------------]
 410 FORMAT[15X,I3,8X,F9.4,8X,F9.4,4X,F6.4,3X,F9.4,8X,F7.2,10X,F7.2]
1075 FORMAT[///,50X,5HFINAL,5X,10HSTATISTICS]
1076 FORMAT[50X,5H-----,5X,10H----------,///]
1077 FORMAT[42X,24HLAMBDA[QUERIES PER MIN.],5X,4HDAYS]
1078 FORMAT[42X,24H------------------------,5X,4H----]
1079 FORMAT[52X,F5.3,16X,I2,///]
1080 FORMAT[60X,4HMEAN,5X,13HSTANDARD DEV.]
1081 FORMAT[60X,4H----,5X,13H-------------]
1082 FORMAT[35X,20HITEMS IN READY QUEUE,5X,F4.1,10X,F5.2,///]
1083 FORMAT[33X,22HITEMS IN BACK-UP QUEUE,5X,F4.1,10X,F5.2,///]
1084 FORMAT[29X,26HAVERAGE READY QUEUE LENGTH,5X,F4.1,10X,F5.2,///]
1085 FORMAT[40X,15HTOTAL IDLE TIME,5X,F5.1,10X,F5.2]
 998 CALL EXIT
     END
```

```
      SUBROUTINE NEWARR [QA,Y,S1M,S2M,S3M,S4M,S5M,TOS]
      COMMON W,PRT
      Y=FLAT[ISTART]
      IF[Y-.1210]301,301,302
 301  TOS=1.0
      SM=S1M
      GO TO 350
 302  IF[Y-.5931]303,303,304
 303  TOS=2.0
      SM=S2M
      GO TO 350
 304  IF[Y-.8962]305,305,306
 305  TOS=3.0
      SM=S3M
      GO TO 350
 306  IF[Y-.9813]307,307,308
 307  TOS=4.0
      SM=S4M
      GO TO 350
 308  TOS=5.0
      SM=S5M
 350  CONTINUE
      CALL POISSN[SM,X,Y]
      UNOS=X/1000.0
      W=X
      QA=TOS+UNOS
      RETURN
      END
```

```
      SUBROUTINE POISSN[P,X,Y]
 900  X=0.0
 901  B=EXPF[-P]
 902  TR=1.0
 903  Y=FLAT[ISTART]
 904  TR=TR*Y
 905  IF[TR-B]908,906,906
 906  X=X+1.0
 907  GO TO 903
 908  CONTINUE
      RETURN
      END
      SUBROUTINE TAPEST[TCM,TREC,WPRT,THS,TSST,BF,TNI1,TNI2,
     1TNI3,TNI4,TNI5,CMIT,TOS,ATTR]
      COMMON W,PRT
      IF[TOS-1.0]600,600,601
  600 TNI=TNI1
      GO TO 610
  601 IF[TOS-2.0]602,602,603
  602 TNI=TNI2
      GO TO 610
  603 IF[TOS-3.0]604,604,605
  604 TNI=TNI3
      GO TO 610
  605 IF[TOS-4.0]606,606,607
  606 TNI=TNI4
      GO TO 610
  607 TNI=TNI5
  610 CT=CMIT*TNI*TREC*ATTR*0.000001
      RSFT=[TCM*TREC*WPRT*0.001/THS]+[[TREC-1.0]*[TSST/BF]]
      PRT=[CT+RSFT]/60.0
      RETURN
      END
```

```
      SUBROUTINE OUTRUR[HDM,Y,TPEP,CTT,TR,QWT,QRT,STP,BPC,BCPD]
      TRTPD=[BCPD*2.0*CTT]/1000.0
      BPD=BCPD*BPC
      TTPD=BPD/TR
      Y=FLAT[ISTART]
      HT=[[1.0/HDM]*LOGF[Y]]*[-1.0]
      N=1
      OUTSTP=TPLP
  101 OUTSTP=OUTSTP-1.0
      IF[OUTSTP-1.0]100,104,102
  102 N=N+1
      GO TO 101
  104 W=0.0
      STP=N+1
      GO TO 105
  100 STP=N
      W=OUTSTP*1000.0
  105 UTT=[W*[TTPD+TRTPD]]/60.0
      QRT=QWT+HT+UTT
      RETURN
      END
      SUBROUTINE DISKST[DRW,WPR,DAT,DDT,DAD,DDR,TOS,DNI1,
     1DNI2,DNI3,DNI4,DNI5,CMIT]
      COMMON W,PRT
      TRWR=DRW*WPR
      DCA=[DAT+DDT]*1000,
      UR=DAD/DDR
      IF[UR-1.0]701,701,702
  701 PR=1.0
      GO TO 710
  702 CONTINUE
      N=2
      DR=UR
  705 DR=UR-1.0
      IF[DR-1.0]703,703,704
  704 N=N*1
      GO TO 705
  703 PR=N
  710 D1=[PR*TOS]*[TRWR+DCA]
      IF[TOS-1.0]711,711,712
  711 DNI=DNI1
      GO TO 720
  712 IF[TOS-2.0]713,713,714
  713 DNI=DNI2
      GO TO 720
  714 IF[TOS-3.0]715,715,716
  715 DNI=DNI3
      GO TO 720
  716 IF[TOS-4.0]717,717,718
  717 DNI=DNI4
      GO TO 720
  718 DNI=DNI5
  720 D2=CMIT*DNI*DAD*TOS
      D3=W*[TRWR+DCA]
      PRT=[D1+D2+D3]*0.000001/60.0
      RETURN
      END
```

Appendix 6 - Document Distributions

Although it has been mentioned many times in this paper that the number of retrieved documents for a given Boolean search have a Poisson distribution, no justification has been given. It is the purpose then, of this section, to clarify such reasoning. All data that is used was obtained from the experimental C.I.S. system at Lehigh University. It should be recalled that both the Lehigh system and this simulation employ a coordinate indexing system.

The C.I.S. system uses over 400 index terms, and it is from these terms that the distribution for a one-term search was obtained. Figure 16 shows the graph of this one-term distribution, where the x-axis designates the number of documents indexed by a specific

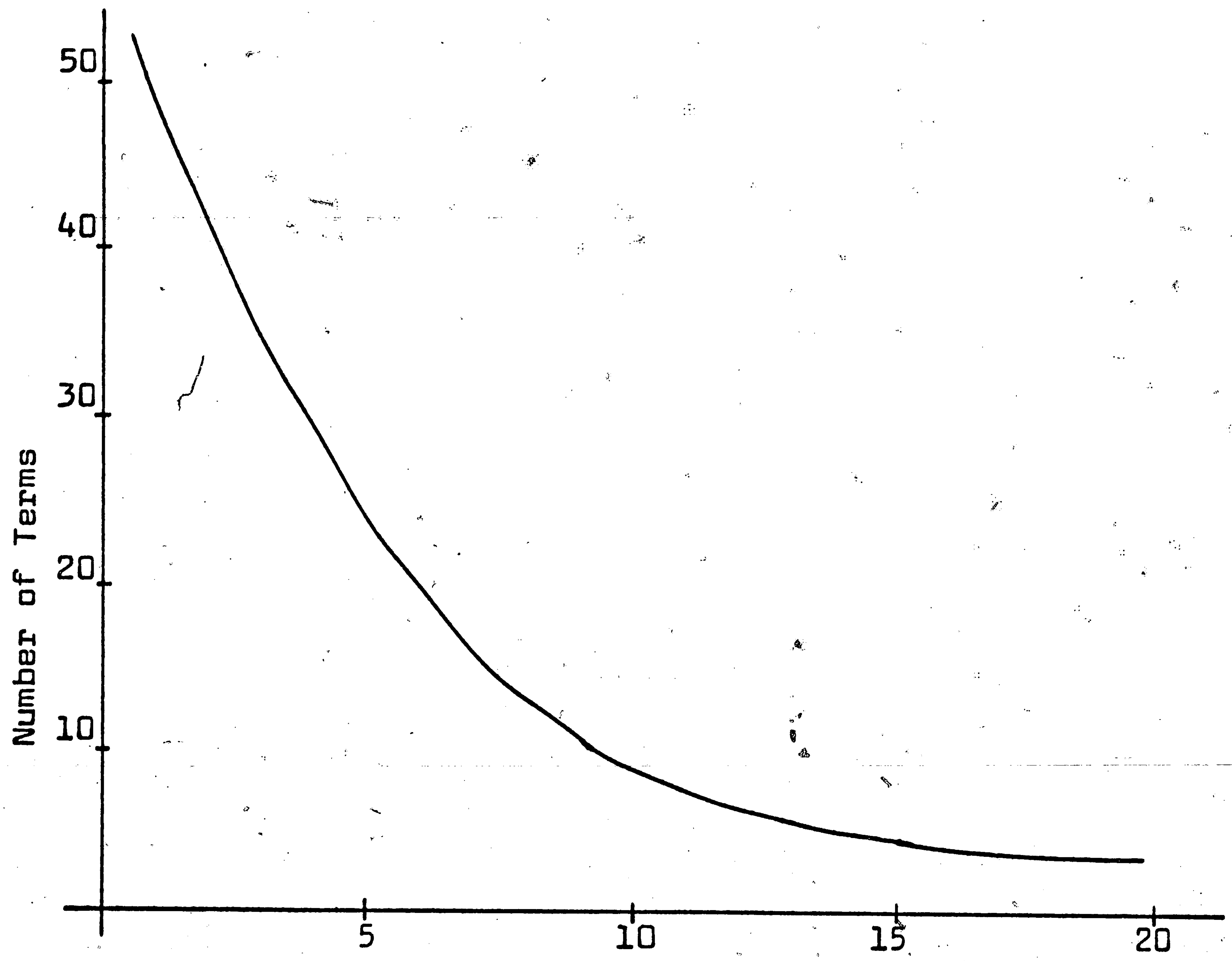


Figure 16. C.I.S. One-Term Document Distribution

term, and the y-axis is the number of times such a situation occurs.

Since one is interested in the number of retrieved documents, a discrete distribution must be used to approximate the actual data. The distribution chosen to represent the one-term document distribution in Figure 16, is the Poisson distribution. The assumption is made that all coordinate indexing systems approximate the C.I.S. system, so the Poisson distribution is adopted in the simulation with the mean for each system left as a variable input parameter. The mean for Figure 16 is 4.08 documents per term.

Figure 17 shows the values for the means of the other types of Boolean searches obtained from the C.I.S. collection. These searches are also assumed to approximate Poisson distributions.

Type of Search	Sample Size	MEAN
2 - Term	20	2.5
3 - Term	20	1.0
4 - Term	20	0.6

Figure 17. Document Distribution for Boolean "AND" Searches

REFERENCES

1. Kleinrock, Leonard, Communication Nets-Stochastic Message Flow and Delays, New York, McGraw-Hill Co., 1964, pp.15 and 136.

2. Herner, Saul, and Herner, Mary, "Determining Requirements for Atomic Energy Information from Reference Questions", International Conference on Scientific Information, Washington, D.C., November, 1958, p.175.

3. Herner, Saul, and Herner, Mary, p.175.

4. Burke, P. J., "The Output of a Queueing System", Operations Research, Vol. 4, 1956, p.700.

5. Center for the Information Sciences, "Systems Manual for the Experimental Literature Collection and Reference Retrieval System", Experimental Retrieval System Studies, Report No. 2, Lehigh University, Bethlehem, Pa., April, 1967.

6. Scheer, A. L., "An Analysis of Time-Shared Computer Systems", Project MAC, Massachusetts Institute of Technology, MAC-TR-18, 1965, p.113.

7. General Electric Corporation, GE Compatibles/200 Disc Storage Unit Subsystem, Phoenix, Arizona, 1964, p.54.

8. General Electric Corporation, GE 225 Programming Reference Manual (CPB-252), Phoenix, Arizona, 1963, p.VIII-18.

9. Naylor, T. H., Balintify, J. L., Burdick, D. S., and Chu, Kong, Computer Simulation Techniques, New York, John Wiley & Sons, Inc., 1966, p.114.

BIBLIOGRAPHY

1. Allard, J. L., A. R. Dobell, and T. E. Hull. "Mixed Congruential Random Number Generators". Journal of ACM. Vol. 10, 1963. pp.131-141.

2. Blunt, Charles R. "An Information Retrieval Systems Model." HRB-Singer Inc. AD-623590, Contract No.(3818(00)). October, 1965.

3. Bourne, Charles P., and Donald F. Ford. "Cost Analysis and Simulation Procedures for the Evaluation of Large Information Systems". American Documentation. April, 1964. pp.142-149.

4. Burke, P. J. "The Output of a Queueing System". Operations Research. Vol. 4, 1956. pp.699-704.

5. Center for the Information Sciences. "Systems Manual for the Experimental Literature Collection and Reference Retrieval System". Experimental Retrieval System Studies. Report No. 2. Lehigh University, Bethlehem, Pa. April, 1967.

6. Chambers, R. P. "Random Number Generation on Digital Computers". IEEE Spectrum. Vol. 4, No. 2, February, 1967. pp.48-56.

7. Chapman, Robert L. "The Case for Information System Simulation". Second Congress on the Information Sciences. Washington, D.C. Spartan Books, Inc. 1965. pp.477-484.

8. Cobham, A. "Priority Assignment in Waiting Line Problems". Operations Research. Vol. 2, 1954. pp.70-76.

9. Cobham, A. "Priority Assignment - A Correction". Operations Research. Vol. 3, 1955. p.547.

10. Dennis, J. B., and Edward L. Glaser. "The structure of On-Line Information Processing Systems". Second Congress on the Information Sciences. Washington, D.C. Spartan Books, Inc. 1965. pp.5-14.

11. Desmonde, William H. Real-Time Data Processing Systems, Introductory Concepts. Englewood Cliffs, N.J. Prentice-Hall Inc. 1964.

12. Feller, William. An Introduction to Probability Theory and Its Applications. Vol. 1, 2nd edition. New York. John Wiley and Sons. 1961.

13. General Electric Corporation. GE Compatibles/200 Disc Storage Unit Subsystem. Phoenix, Arizona. 1964.

14. General Electric Corporation. GE 225 Programming Reference Manual (CPB-252). Phoenix, Arizona. 1963.

15. Goode, Harry H., and Robert E. Machol. System Engineering, An Introduction to the Design of Large-Scale Systems. New York. McGraw-Hill Book Co. 1957.

16. Greenberger, Martin. "A New Methodology for Computer Simulation". Project MAC, Massachusetts Institute of Technology. MAC-TR-13. 1964.

17. Hammersley, J. M., and D. C. Handscomb. Monte Carlo Methods. New York. John Wiley and Sons, Inc. 1964.

18. Head, Robert V. Real Time Business Systems. New York. Holt, Rinehart and Winston, Inc. 1966.

19. Heilprin, L. B. "Information Storage and Retrieval as a Switching System". (Reprinted for Switching Theory in Space Technology, edited by Aiken and Main.) Stanford University Press. 1963. pp.298-332.

20. Herner, Saul, and Mary Herner. "Determining Requirements for Atomic Energy Information from Reference Questions". International Conference on Scientific Information. Washington, D.C. November, 1958. pp.171-177.

21. Holley, J. L. "Waiting Line Subject to Priorities". Operations Research. Vol. 2. 1954. pp.341-343.

22. Hull, T. E., and A. R. Dobell. "Mixed Congruential Random Number Generators". Journal of ACM. Vol. 11. 1964. pp.31-40.

23. Kleinrock, Leonard. Communication Nets-Stochastic Message Flow and Delay. New York. McGraw-Hill Book Co. 1964.

24. Levy, W. A. "Computer-Communications Interface". Computer Design. September, 1964. pp.20-22.

25. Martin, James. Programming Real-Time Computer Systems. New Jersey. Prentice-Hall, Inc. 1965.

26. McMillan, Claude, and Richard Gonzalez. Systems Analysis, A Computer Approach to Decision Models. Homewood, Ill. Richard D. Irwin, Inc. 1965.

27. Meyer, Paul L. Introductory Probability and Statistical Applications. Reading, Mass. Addison-Wesley Co. 1965.

28. Morse, P. M. "Stochastic Properties of Waiting Lines". Operations Research. Vol. 3. 1955. pp.255-261.

29. Naylor, T. H., J. L. Balintify, D. S. Burdick, and Kong Chu. Computer Simulation Techniques. New York. John Wiley & Sons, Inc. 1966.

30. Scherr, A. L. "An Analysis of Time-Shared Computer Systems". Project MAC, Massachusetts Institute of Technology. MAC-TR-18. June, 1965.

31. Wallace, Victor L. "Queueing Theory". Digital Computers in Real Time. The University of Michigan Engineering Summer Conference. 1964.

Vita

James Patrick Foster, son of John and Doris Foster, was born in Lebanon, Pennsylvania, on August 23, 1940. He was graduated from Lebanon Catholic High School in 1958, and after attending Holy Cross College, Worcester, Massachusetts, and St. Vincent College, Latrobe, Pennsylvania, he received a Bachelor of Science degree in Electrical Engineering from The Pennsylvania State University in March, 1963.

Mr. Foster was employed as an electrical engineer by the Radio Corporation of America prior to his appointment as a National Defense Education Act Fellow at Lehigh University. He is a member of Eta Kappa Nu and Sigma Tau honorary engineering fraternities.