Proceedings of NASA

NASA/INDUSTRY PERT COMPUTER CONFERENCE

JULY 22 AND 23 1964

MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

TABLE OF CONTENTS

AGENDA

NASA/Industry PERT Computer Conference

July 22

8:00 Transportation departs Downtown Area

8:30 Transportation departs Airport Location

9:00 Transportation arrives Manned Spacecraft Center

9:00 - 9:30 Registration: Each attendee is requested to register with the NASA personnel in the Auditorium Lobby. A correct mailing address is essential to receipt of the "Proceedings".

9:30 - 9:45 Welcome to the Manned Spacecraft Center

Maxime A. Faget, Assistant Director for Engineering and Development
Eugene H. Brock, Chief, Computation and Analysis Division

9:45 - 10:00 "NASA PERT, Past, Present and Future" - A brief description of the current and future use of NASA PERT by the Director, Management Information Systems - Walter Haase

10:00 - 10:15 Coffee

10:15 - 12:00 "NASA PERT "C" Program" - Personnel from the Manned Spacecraft Center will describe the 7090/94 machine language program they have developed for agency-wide use. The variable input formats, the standard and optional output reports, as well as the techniques utilized in topological ordering and merge processing will be described in some detail.

Larry Stevens
John Leonard

12:00 - 12:30 "Report of NASA PERT "C" Pilot Test" - Two representatives of industry who have participated in the testing of NASA PERT "C" will describe their experience with the program.

McDonnell Automation Center - Harry Parsh
The Boeing Company - Homer Smith

12:30 - 1:30 Lunch

July 22

1:30 - 2:45 "The NASA PERT Fortran IV Program" - Personnel from the Lewis Research Center and Goddard Space Flight Center will describe the Fortran IV program they have jointly developed and released as well as some additional developments now undergoing test.

Ross Bainbridge
Hans Bremer
Elizabeth Ryan

2:45 - 3:00 Coffee

3:00 - 3:15 "The NASA PERT 1410 System" - Personnel from the John F. Kennedy Space Center will describe the 1410 system that they have developed for production of the standard NASA PERT reports.

Al Sheffield

3:15 - 3:45 "Use of the SC-4020 for NASA PERT Graphic Output" - The Marshall Space Flight Center has developed a program which presents the NASA PERT information in the more traditional master milestone schedule format. Through this technique NASA management personnel are supplied key milestone information in graphic display directly from the computer.

Jerry Huffman

3:45 - 4:15 "Plans for the Mechanization of the Contractor Financial Management Report" - A representative of the Headquarters Financial Management Division will discuss NASA's approach and plans in the area of cost information.

Ken Mulcahy

4:15 - 5:00 Questions and Answers - All persons making presentations will be available during this time to answer inquiries from attendees.

July 23

8:00 Transportation departs Downtown Area

8:30 Transportation departs Airport Location

9:00 Transportation arrives Manned Spacecraft Center

July 23

9:00 - 12:00 Group Discussions - Three informal discussion groups oriented towards the major agenda items will meet separately to discuss in detail the programs presented on the previous day. The groups will be organized as round-table discussions with the emphasis on a mutual interchange of information in an informal environment. Participation by attendees in these groups is optional. The groups will be:

Group A - NASA PERT "C"
Group B - Fortran Program
Group C - Utility Programs

NASA PERT PAST, PRESENT, AND FUTURE

by

Walter W. Haase
Director, Management Information Systems Division
NASA Headquarters, Washington, D.C.

Thank you Mr. Brock. Ladies, Gentlemen - I appreciate the opportunity to be here today to discuss NASA PERT computer program developments. We have been utilizing PERT since early 1961 and have made continual improvements in our PERT computer programs as a result of operational experience with the system. As previously indicated, the purpose of this conference is to present these developments to you in the hope that some of the concepts and techniques developed by NASA may be of value to you. in your PERT operations. In addition, through the medium of tomorrows discussion groups we hope to learn about new developments elsewhere.

During the day you will be hearing detailed presentations on the various computer programs which have been developed. No one group within NASA has been solely responsible for these developments. It has been a joint venture with contributions from many NASA field centers. The detailed presentations on the various aspects of the program will be made by the organizations responsible for developing the program - and in many cases - by the individuals that did the programming.

Before proceeding with the detailed presentations, I would like to take a few minutes to review the history of PERT within NASA. The PERT system of planning and scheduling was introduced in NASA during the early months of 1961. From an initial pilot implementation on CENTAUR at Marshall Space Flight Center and a few facility construction projects at Langley Research Center, acceptance and utilization of PERT has grown to considerable proportions in the past three years. At the present time approximately 79 R&D contracts, 189 C of F contracts and 36 in-house project efforts consisting of over two hundred thousand contractor and in-house activities are monitored by the PERT system . The total contract value of efforts monitored is approximately six billion dollars.

The chronology of PERT computer program developments - from the early months of 1961 to todays introduction of the NASA PERT "C" and FORTRAN IV Programs - is illustrated on this vugraph (VG #1 - Tab 1). In 1961 we adopted existing programs developed by other government agencies and industries. The two programs used were the then popular, Lockheed and Navy (Dahlgren) programs.

By mid-1961 (VG #1 - Tabl 2), we initiated development of our NASA PERT "A" Computer Program. This program developed by the Langley Research Center, Hampton, Virginia, - was essentially a modification of the Lockheed program. The majority of the changes involved in-put and out-put format rather than the internal operations of the program. After pilot testing in NASA, the program was released to industry in March 1962 and used by our field centers throughout the major portion of calendar year 1962.

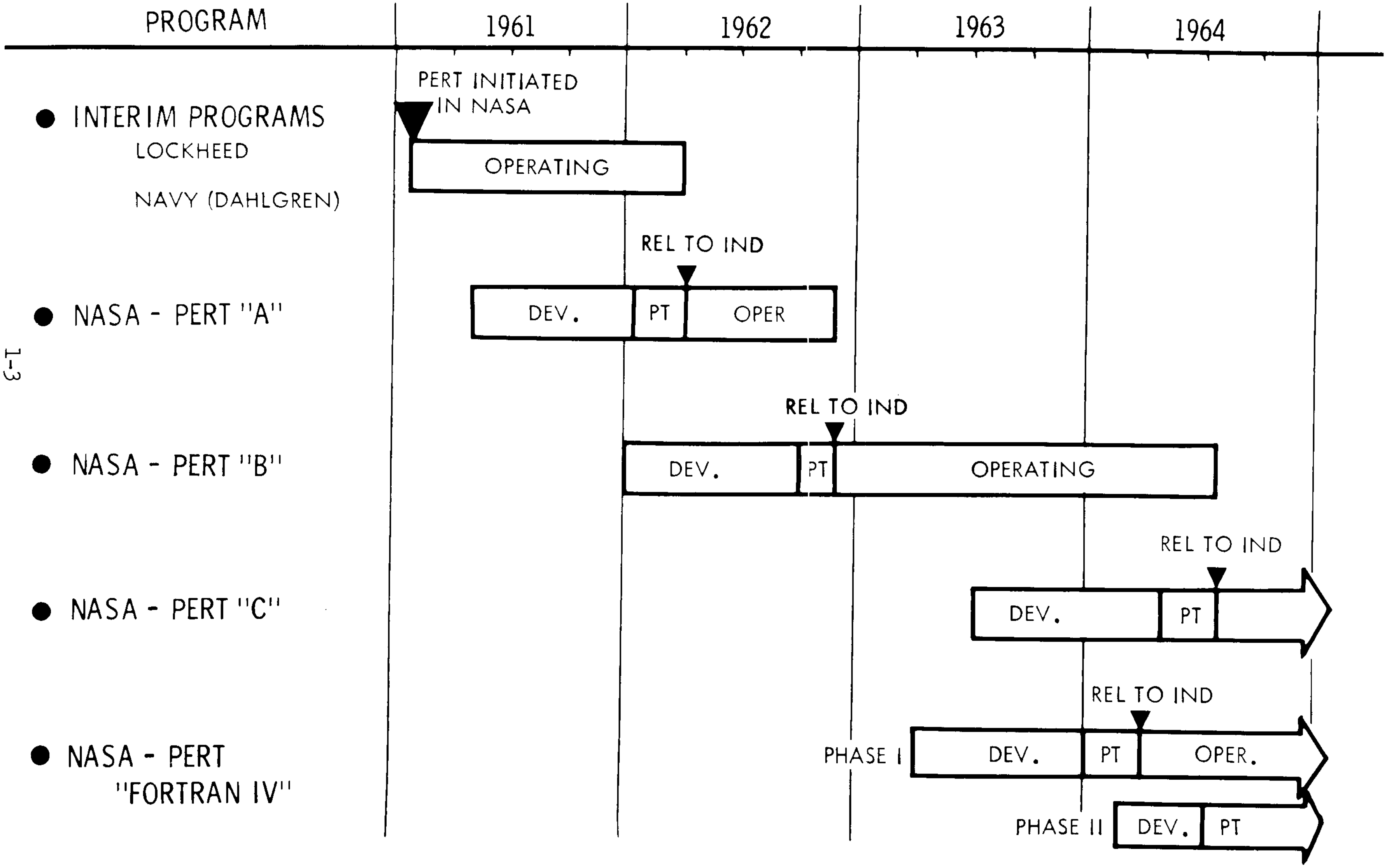
Late in calendar year 1961 (VG #1 - Tab 3), the development of NASA PERT "B" was undertaken by the Manned Spacecraft Center, Houston, Texas. This was a more extensive program development and incorporated many features not found on programs available at that time. These features are described in our NASA PERT "B" Computer Systems Manual. This program was pilot tested within NASA field centers and became the official NASA PERT program in November 1962 at which time it was released to industry.

A number of modifications to the NASA PERT "B" Program have been developed over the past year. To reduce the burden of up-dating documentation and the large number of computer programs which had been released to industry, we elected to accumulate these modifications and release them in a batch as the "C" version of the program (VG #1 - Tab 4). We have completed pilot testing of these modifications and are now ready to release the "C" version of the NASA PERT Computer Program. This program is one of the primary topics of today's discussion.

The NASA PERT "A" - "B" and "C" programs have all been machine language 7090/94 programs. To overcome the criticism that NASA was developing computer programs unique to one manufacturers equipment, and because of machine configuration changes at Lewis Research Center which precluded using the existing program without extensive changes, we undertook the development of a compiler language program in mid-1963 (VG #1 - Tab 5). This programming effort, in FORTRAN IV, was jointly undertaken by Lewis Research Center in Cleveland, Ohio and Goddard Space Flight Center in Greenbelt, Maryland, and was broken down into two phases.

NASA-PERT COMPUTER PROGRAMS

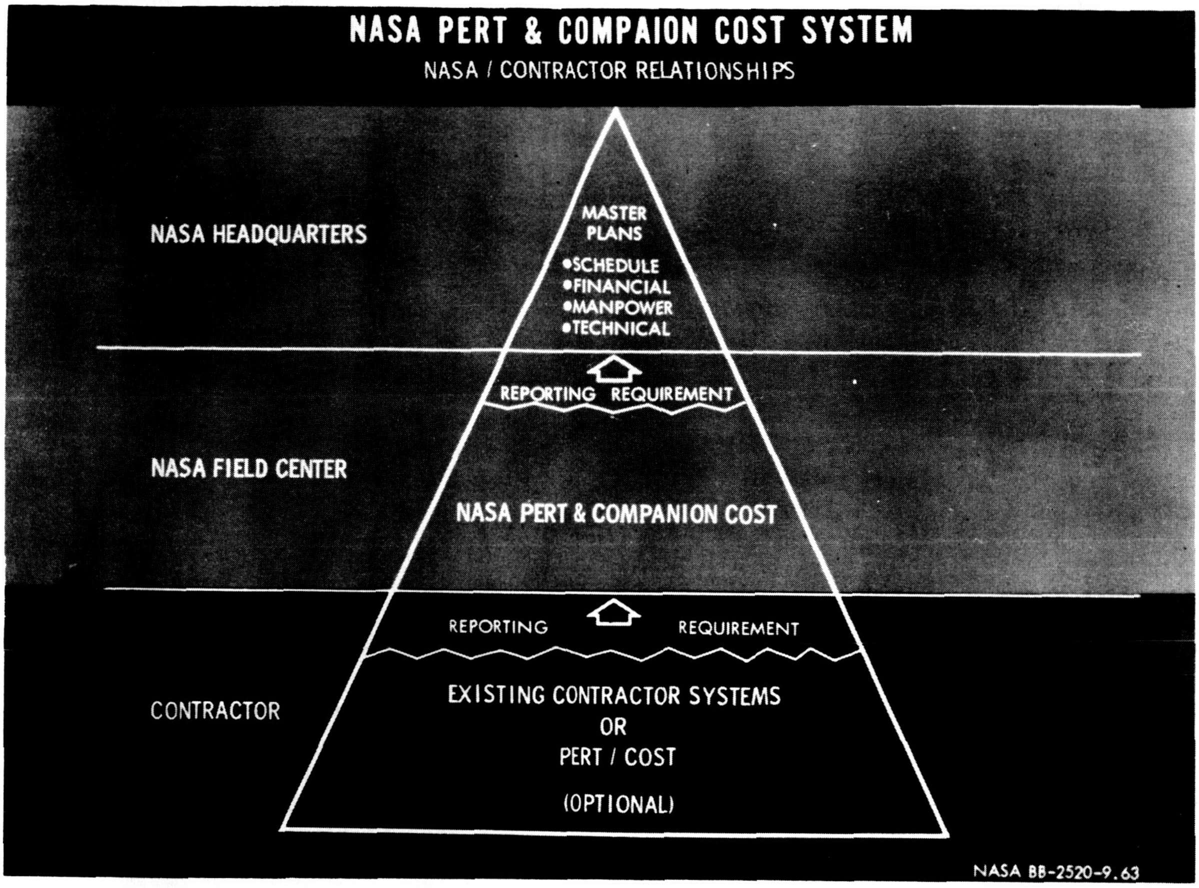
DEVELOPMENT CHRONOLOGY



Phase 1 - the development of a basic module capable of processing 3500 activities and phase 2 - the expansion of the program to permit processing larger networks by utilizing the phase 1 module in conjunction with skeletonizing or summarization techniques. Phase 1 of this programming effort was completed late in 1963, pilot tested early 1964, and released to industry through SHARE in March 1964. Phase 2 development is essentially complete but has not been pilot tested and is not ready for release. This FORTRAN IV program will also be described in detail in subsequent presentations.

In addition, NASA has done some work on a 1410 computer program which will be described today. This effort was undertaken to fulfill the processing needs of the John F. Kennedy Space Center, which does not have ready access to a medium or large scale computer at this time. Although the basic concept of the PERT system has remained unchanged since its introduction in 1958, there are many variations of the system in existence. Most of these variations stem from the manner in which the system is utilized by a particular organization. The purpose for which the system is used, the level of management the system serves, the level of detail, etc., all have influenced the detailed design and implementation of the system. Consequently, I believe it would be appropriate to review the manner in which the system is used in NASA prior to detailed discussion of the related computer programs.

The total NASA effort is divided into major program areas such as: the LUNAR AND PLANETARY EXPLORATION PROGRAM, GEOPHYSICS AND ASTRONOMY PROGRAM, APOLLO PROGRAM, GEMINI PROGRAM, etc. These programs are further subdivided into distinct projects such as, SATURN V, APOLLO SPACECRAFT, F-1 ENGINE DEVELOPMENT which are portions of the APOLLO PROGRAM. Program level management is the responsibility of NASA Headquarters whereas project level management is the responsibility of the various decentralized field centers such as, the Manned Spacecraft Center here at Houston, the Lewis Research Center in Cleveland, etc. The NASA PERT system is basically designed as a tool for the NASA project manager in the field center. It is not uncommon for a single project to involve a number of associate contractors as well as extensive in-house work by NASA field installations. The NASA PERT system is designed for the NASA project manager to integrate the various pieces of the project - various contractor efforts as well as in-house efforts - into an integrated project plan. The level of detail in these networks is at a summary level - the level which is needed by the NASA project manager to do his job of integrating the pieces of the program. This system must, of course, be backed-up by a more detailed system at the contractor level. However, our management has not elected to make the use of PERT or PERT/COST a mandatory requirement on the contractor. We require only that there be a logical system of some type at the contractor level to back-up the summary level activities on the NASA overall project level PERT network. This back-up can be in the form of Gantt charts, milestone charts, or in the form of more detailed PERT or PERT/COST, whatever the case maybe.



NASA PERT & COMPAION COST SYSTEM

NASA / CONTRACTOR RELATIONSHIPS

NASA HEADQUARTERS

MASTER PLANS
- SCHEDULE
- FINANCIAL
- MANPOWER
- TECHNICAL

REPORTING REQUIREMENT

NASA FIELD CENTER

NASA PERT & COMPANION COST

REPORTING REQUIREMENT

CONTRACTOR

EXISTING CONTRACTOR SYSTEMS
OR
PERT / COST
(OPTIONAL)

NASA BB-2520-9.63

Vugraph # 2

The next vugraph (VG #2) summarizes the overall NASA/Contractor relationships in the NASA PERT system. It illustrates the position of PERT in the management structure and shows how in-puts from the contractor's systems are integrated and summarized through PERT at the field center and presented as summary level master plans at the Headquarters level.

We are also working on the utilization of PERT at the program level of management in NASA Headquarters. I believe that you can visualize the desirability of integrating various project plans into an overall program network. The Office of Manned Space Flight at NASA Headquarters in Washington is now developing an overall Manned Space Flight PERT network to include GEMINI, APOLLO and post APOLLO efforts. Significant progress in this effort has been made, and some of the techniques under development for summarization of the project level PERT data will be touched on today. However, the system is not expected to be completely operational until later this fall.

With this introduction to NASA PERT, Past, Present, and Future -- let us now move on to the detail presentation on the PERT computer program developments.

NETWORK MAINTENANCE AND PROCESSING

TECHNIQUES USED IN CONJUNCTION WITH NASA PERT "C"

by

Larry Stevens and John Leonard
NASA Manned Spacecraft Center
Houston, Texas

INTRODUCTION - (Overall processing cycle)

The network maintenance phase of the PERT processing cycle is performed at MSC on a 1401, due to the relatively low cost and accessibility of this computer. However, a Fortran program is available for those who do not have access to a 1401.

This chart (Figure 1) depicts the PERT processing cycle used at MSC. Update data is received by various means from contractors located all over the United States. It is sequenced, edited and placed on tape, using our 1401 utility package. The tape then become the input medium for the NASA PERT "C" program. If no errors are detected during processing, the reports are produced and returned to the analyst. If data errors are detected, a diagnostic will be produced. These errors must be corrected and the processing cycle re-entered at the update point.

Because of the size and nature of the networks processed at MSC, all networks are maintained on tape. To more fully utilize the capacity of a magnetic tape, the networks pertaining to a specific project are placed on the same master tape, setting each up as a separate file (Figure 1a), or as fragnets of one large network (Figure 2). All networks are blocked, 16 activities to a block.

To provide the capability of using multi-file input, such as that in figure 2, a special input card was developed for the PERT "C" program and called an "UPPERT" card. By using an UPPERT card, it is possible to change the input tape at execution time to either of 2 tape drives not normally used and to select files in random order to be processed. Figure 3 indicates the format of the UPPERT card.

UPDATE PACKAGE

A. Label Checking

In an attempt to insure against updating the incorrect master file or destroying a current master file, tape and network labels are placed on all network files. The first record on each tape is a Tape Label. This label contains Tape ID, a Tape Cycle Number (either 1, 2, 3 or 4), and an Update Sequence Number.

Each tape file of networks is maintained on a group of 4 reels. As each update is made, label checks are performed to insure against updating a wrong reel (Figure 4). The first card of the update deck must contain the Tape ID and the Tape Cycle Number (1, 2, 3, or 4). Both Tape ID and Cycle Number on the card and Old Master Tape must match before an update can be performed. Also, the Tape ID's of both old and new master tapes must compare, and the Cycle Numbers of the two tapes be sequential from old to new, i.e. (1-2, 2-3, 3-4, and 4-1).

The 7094 Fortran and the 1401 update packages both perform the same label checking procedure.

B. Editing

By following the philosophy of doing as much processing as is practical on the lower cost 1401, a great deal of editing is performed using the utility package:

1. Sequence Check - All activities within a network must be in numeric sequence by predecessor and successor event number. A list of activities out of sequence is printed.

2. Duplicates - All activities must be uniquely identified. When duplicates are found in the update deck, both are printed but only the first is used to update the network and the second is placed as a duplicate.

3. Predecessor-Successor Equal - The predecessor and successor event numbers cannot be the same. A check is made for this condition and when found, the activity is printed.

4. Illegal Code in Column (1) - Characters other than 1 through 5 are not legal transaction codes. Illegally coded activities are ignored during the update.

5. Check Date - All completed activities must contain a date.

C. Update Networks

A transaction list indicating all changes to the network as well as all invalid data is printed during an update.

An update can be made from either cards or tape. On large updates (several thousand cards), it is very likely that there will be some invalid cards. Rather than go through the relatively slow card update and edit process, a faster "card-to-tape and edit" routine is used to detect invalid data prior to the updating process. The resulting tape is used for a tape update. The combination of two routines, card-to-tape and edit, and tape update, actually requires less 1401 time than the straight card update and edit process. On large updates some contractors use the card-to-tape and edit routine, then transmit tape-to-tape to a NASA Center, using Data Phone transmission. With no additional preparation the resulting tape can be used to update the network.

In the process of maintaining networks on tape, we found that it was frequently necessary to change just the A, B or C control cards of a network prior to execution. Therefore, the capability of changing these header cards at network processing time was added to the PERT "C" system. Using a feature of the UPPERT card, any combination of the header cards may be changed.

To provide more flexibility, the feature to read the UPPERT card and control cards either from the A-2 tape or from the on-line card reader has been incorporated into the system.

D. Demerge

Each network maintained at MSC has a unique set of event numbers. Therefore, any combination of fragnets can be merged and processed as one network, using the 1401 utility package. One example of how this capability might effectively be utilized is as follows:

Suppose within a space project a series of networks is developed for each subsystem such as shown in Figure (5). If the networks are structured properly, a person can extract all fragnets relating to one particular vehicle from each of the subsystem networks, merge them and examine the complete status of that particular vehicle. We are trying to illustrate that the utility package will select any number of specified fragnets from separate networks and merge them for processing as one network with no network continuity errors.

Another example of how the demerge technique is used is as follows: At MSC our largest network is in connection with the Apollo Project. The project is broken into 28,000 activities which are maintained as one large network. This network is processed two ways; first as a merged network, and second as a set of 28 demerged subsystem networks, each processed independently of the others.

Because of the use of the code 4 interface activities, once the merged network has run with no input diagnostics, 28 subsystem networks will also run, using no additional input cards.

Each of the 28 subsystems are extracted from the merged master network and written as separate stacked jobs on another tape. Because of a look-ahead feature in PERT "C", control remains within the PERT "C" system until all 28 subsystem networks are processed. Since PERT "C" is not an integral part of our general in-house monitor system, this ability to stack PERT jobs saves a considerable amount of machine time because it is not necessary to return to the monitor system between PERT jobs.

OTHER UTILITY ROUTINES

In Addition to providing update, edit, label checking, and demerge capability, the 1401 utility package can also perform the following operations in the area of file maintenance:

A. Print any network or fragnet of a network. (Figure 6)

B. Punch any network or fragnet of a network. (Figure 7)

C. Delete selected network from a master tape.

D. Delete selected fragnets from a master network.

E. Copy a master network or a fragnet onto a new tape.

F. Add a network to a master tape. (Figure 10)

G. Card-to-tape and edit routine. (Figure 11)

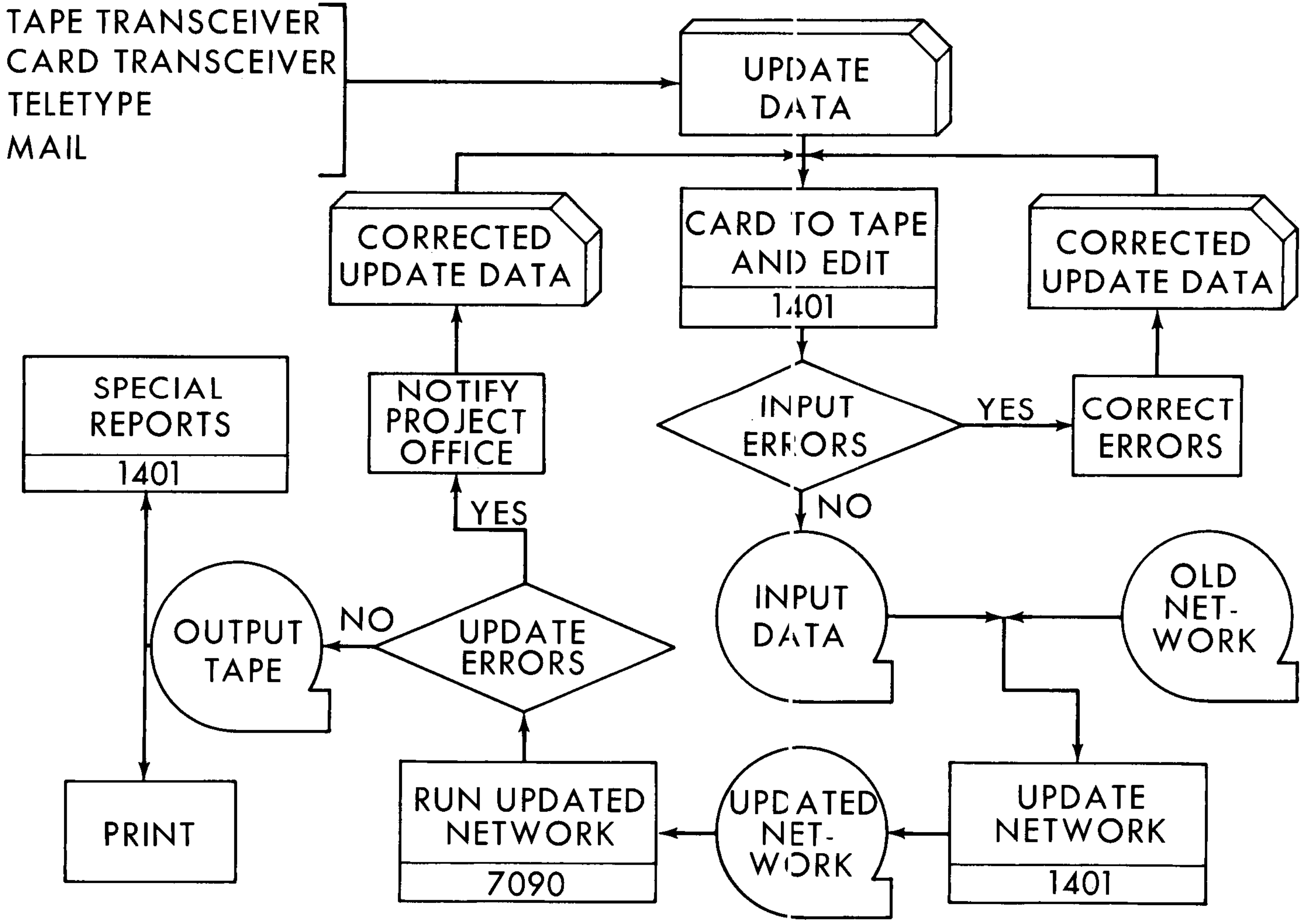
H. Delete all completed activities from a network. Then, provide a start date for each path and tie the paths together. (Figure 12)

Other 1401 routines which have been developed to assist PERT analysis in working large networks are:

A. Print all interface activities in a network. (Network interfaces are frequently a source of trouble).

B. Print all predecessor or successor events to a given event. This is very helpful in correcting a "constrained but complete" condition in a large network.

C. Another 1401 routine will produce a graphic display of the Master Schedule Report.



NASA-S-64-4749

PERT PROCESSING CYCLE

Figure 1

NASA-S-64-4653

TYPE 1 TAPE FORMAT

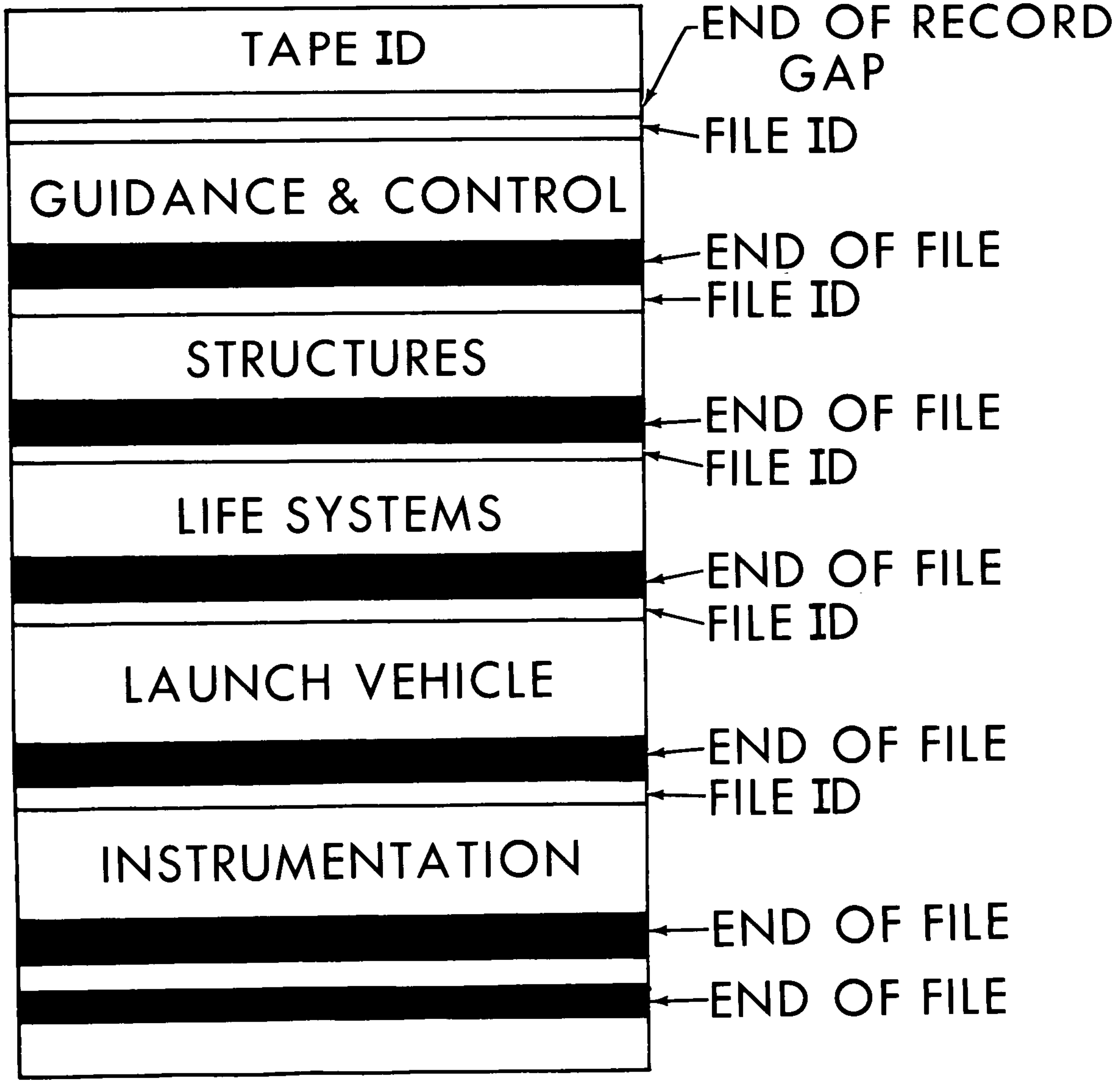


Figure 1a

NASA-S-64-4654

TYPE 2 TAPE FORMAT

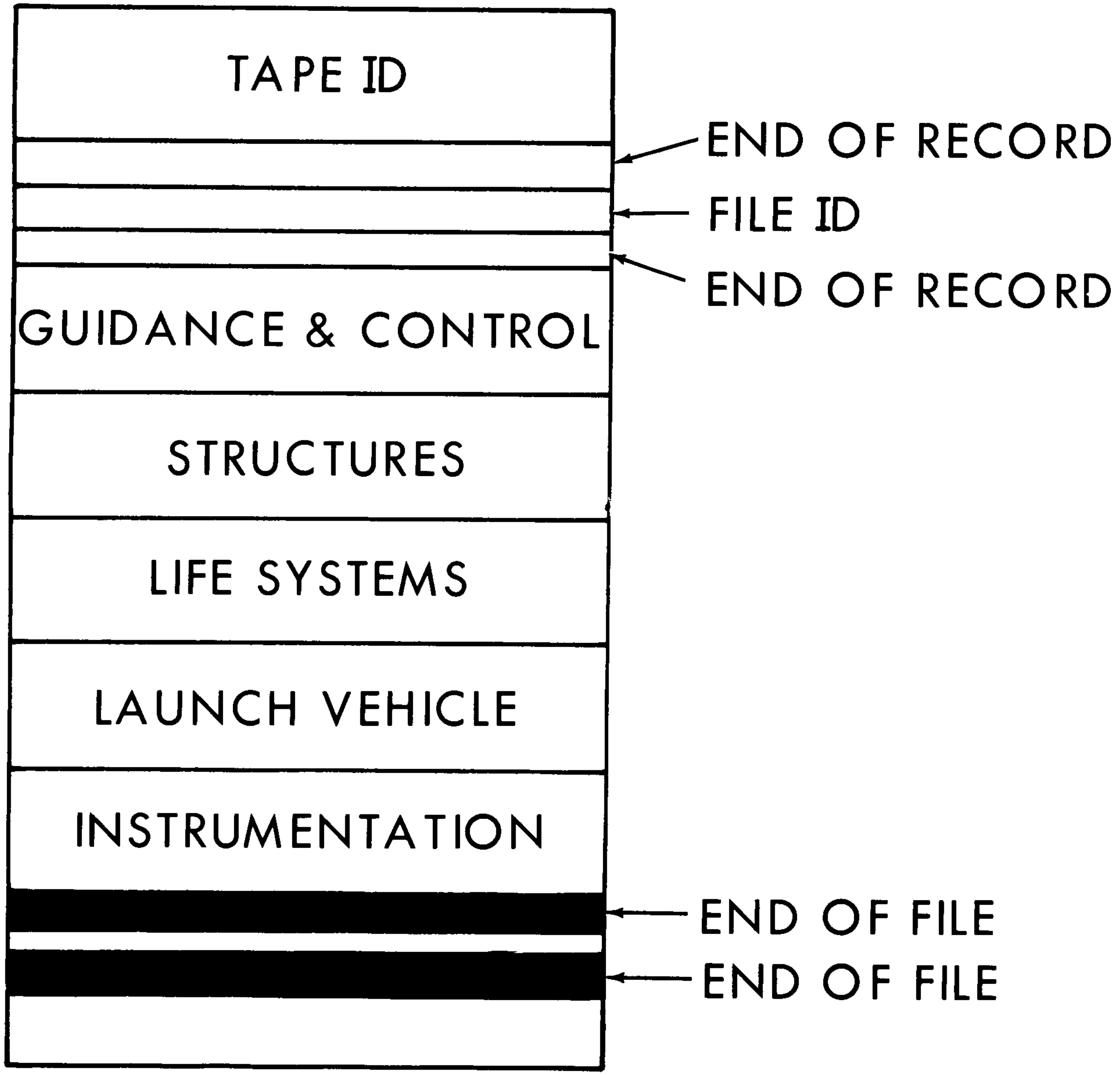


Figure 2

NASA-S-64-4752

UPPERT CARD FORMAT

```
COLUMN 1      8        16   18   21             32
       *      UPPERT   X,   Y Y,  Z . . . . . . . Z
```

X=1	INPUT TAPE MOUNTED ON A2
X=2	INPUT TAPE MOUNTED ON A7
X=3	INPUT TAPE MOUNTED ON A8
YY=1 TO 99	A TWO DIGIT NUMBER SPECIFYING THE FILE TO BE EXECUTED
Z . . . . . Z	A TWELVE DIGIT NETWORK IDENTIFICATION

Figure 3

NASA-S-64-4657

DECK SETUP FOR AN UPDATE

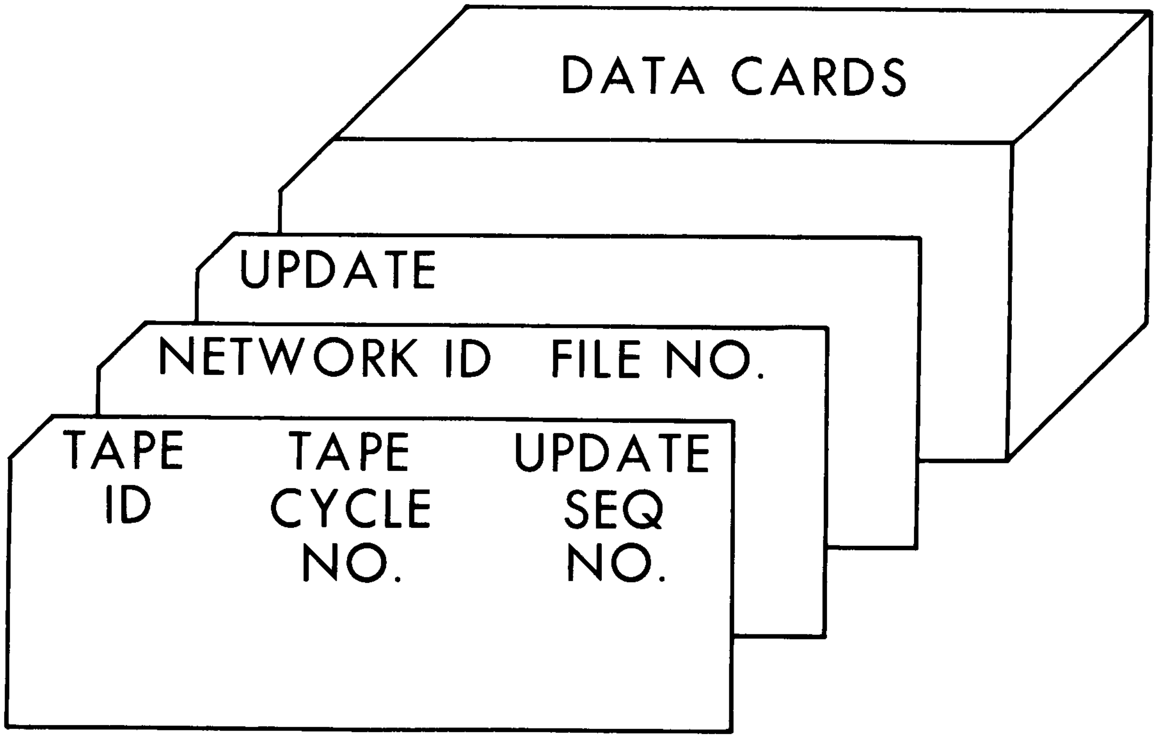


Figure 4

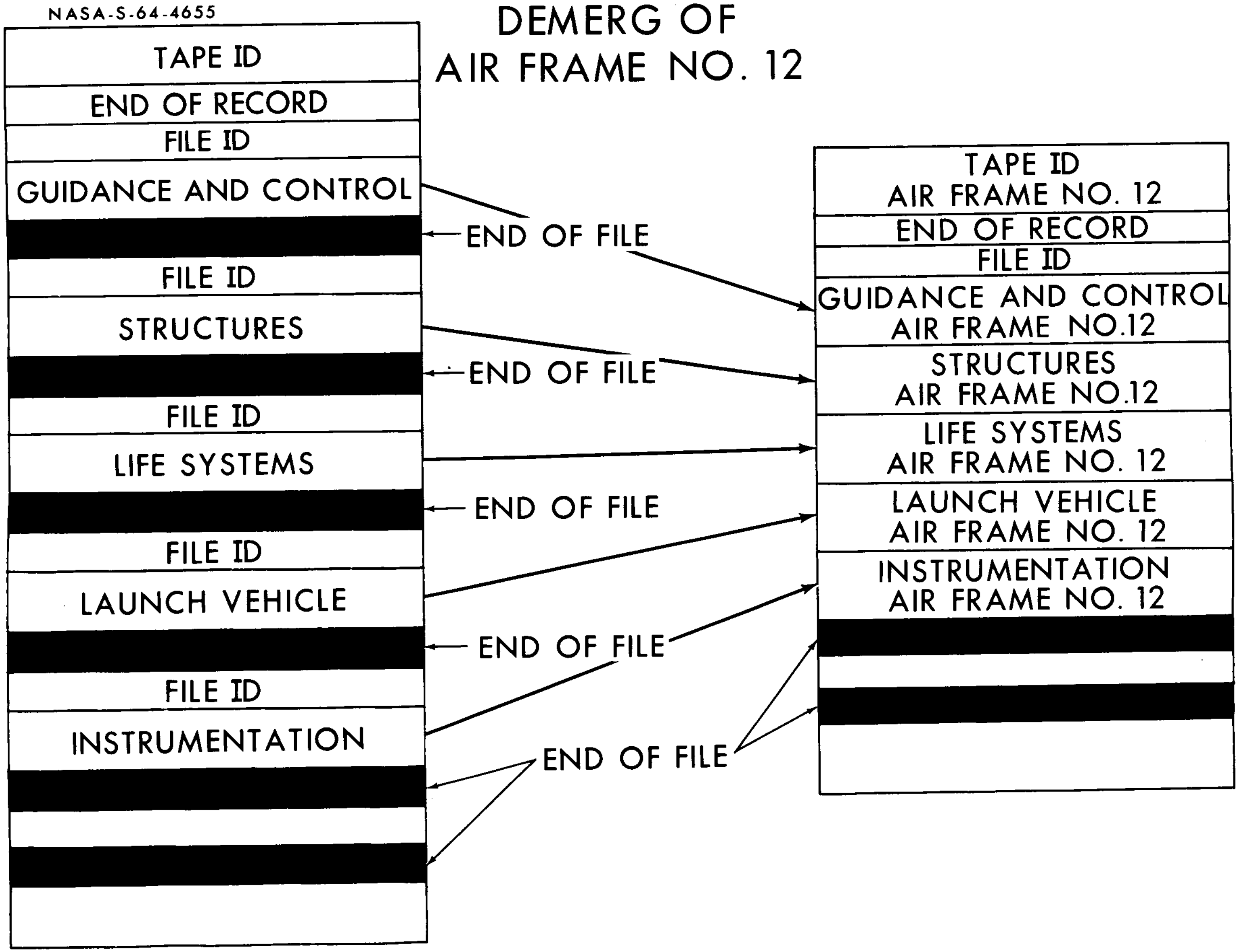

NASA-S-64-4655
DEMERG OF
AIR FRAME NO. 12
TAPE ID
END OF RECORD
FILE ID
GUIDANCE AND CONTROL
FILE ID
STRUCTURES
FILE ID
LIFE SYSTEMS
FILE ID
LAUNCH VEHICLE
FILE ID
INSTRUMENTATION
END OF FILE
END OF FILE
END OF FILE
END OF FILE
END OF FILE
TAPE ID
AIR FRAME NO. 12
END OF RECORD
FILE ID
GUIDANCE AND CONTROL
AIR FRAME NO.12
STRUCTURES
AIR FRAME NO.12
LIFE SYSTEMS
AIR FRAME NO. 12
LAUNCH VEHICLE
AIR FRAME NO. 12
INSTRUMENTATION
AIR FRAME NO. 12

Figure 5

NASA-S-64-4691

PRINT

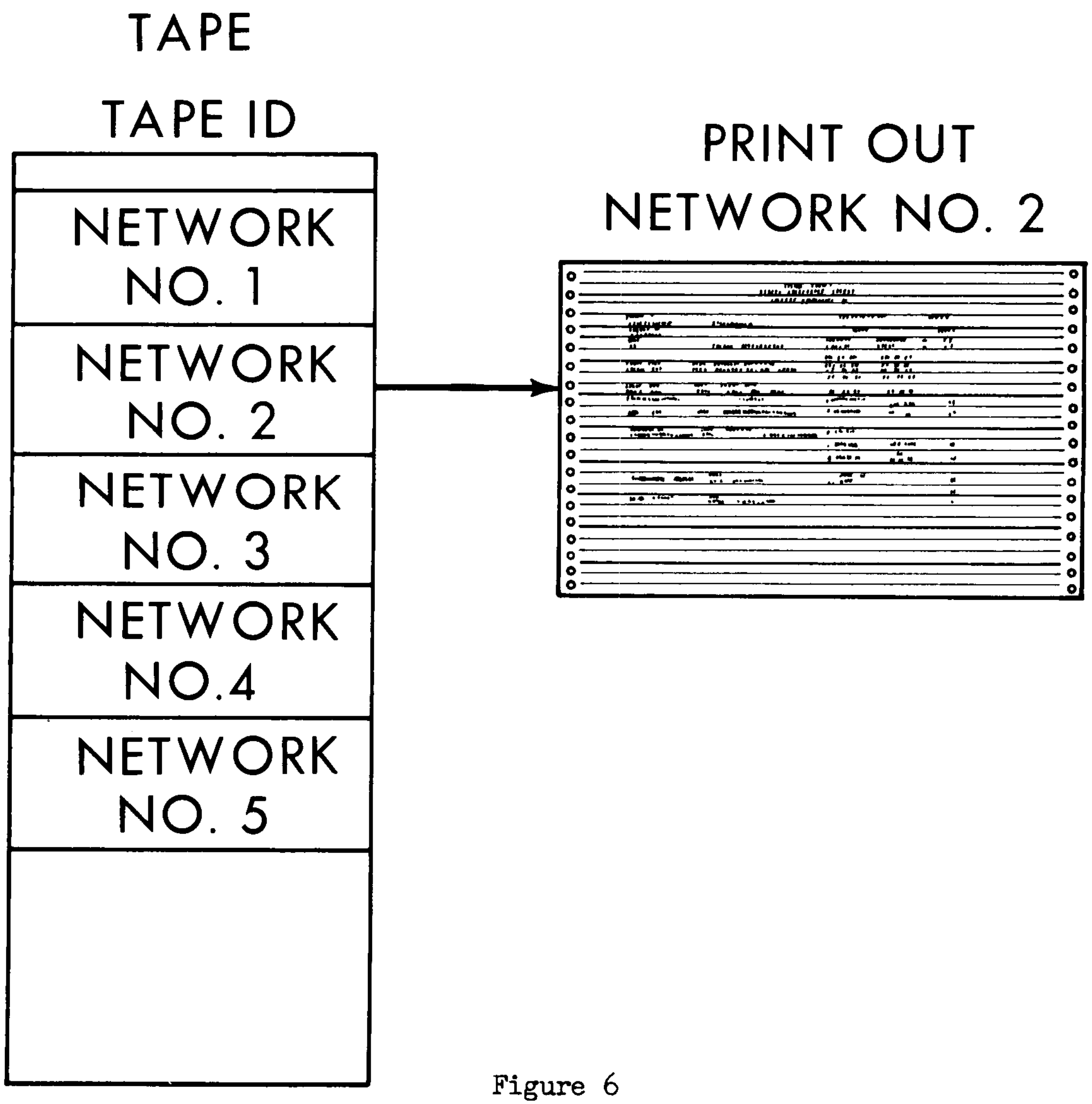


Figure 6

NASA-S-64-4698

PUNCH

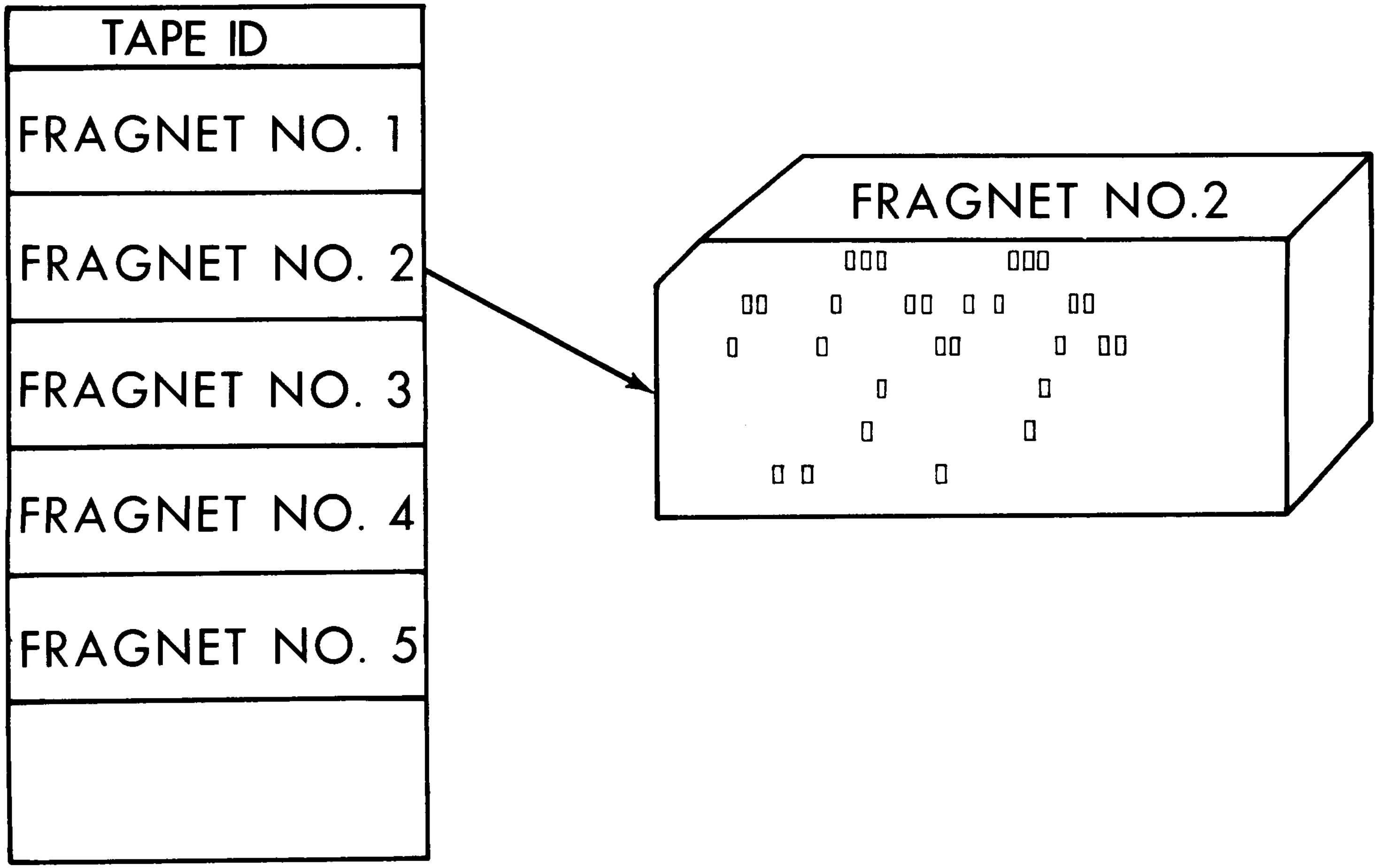


Figure 7

NASA-S-64-4699

DELETE

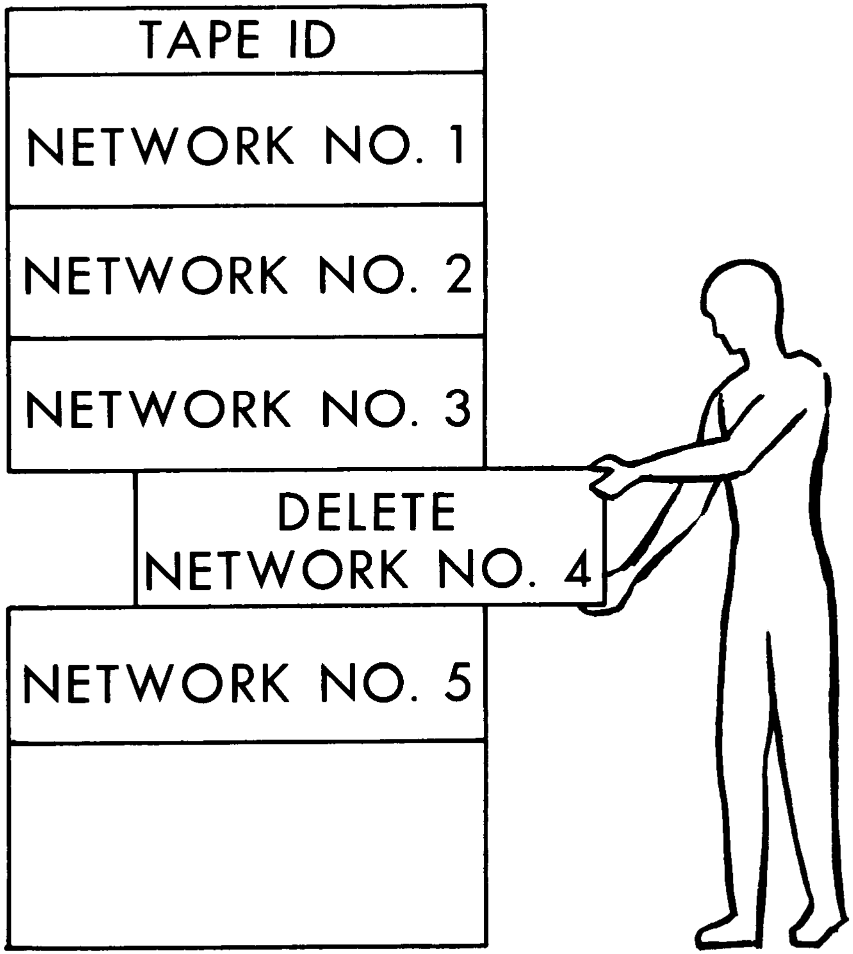


Figure 8

NASA-S-64-4700

COPY

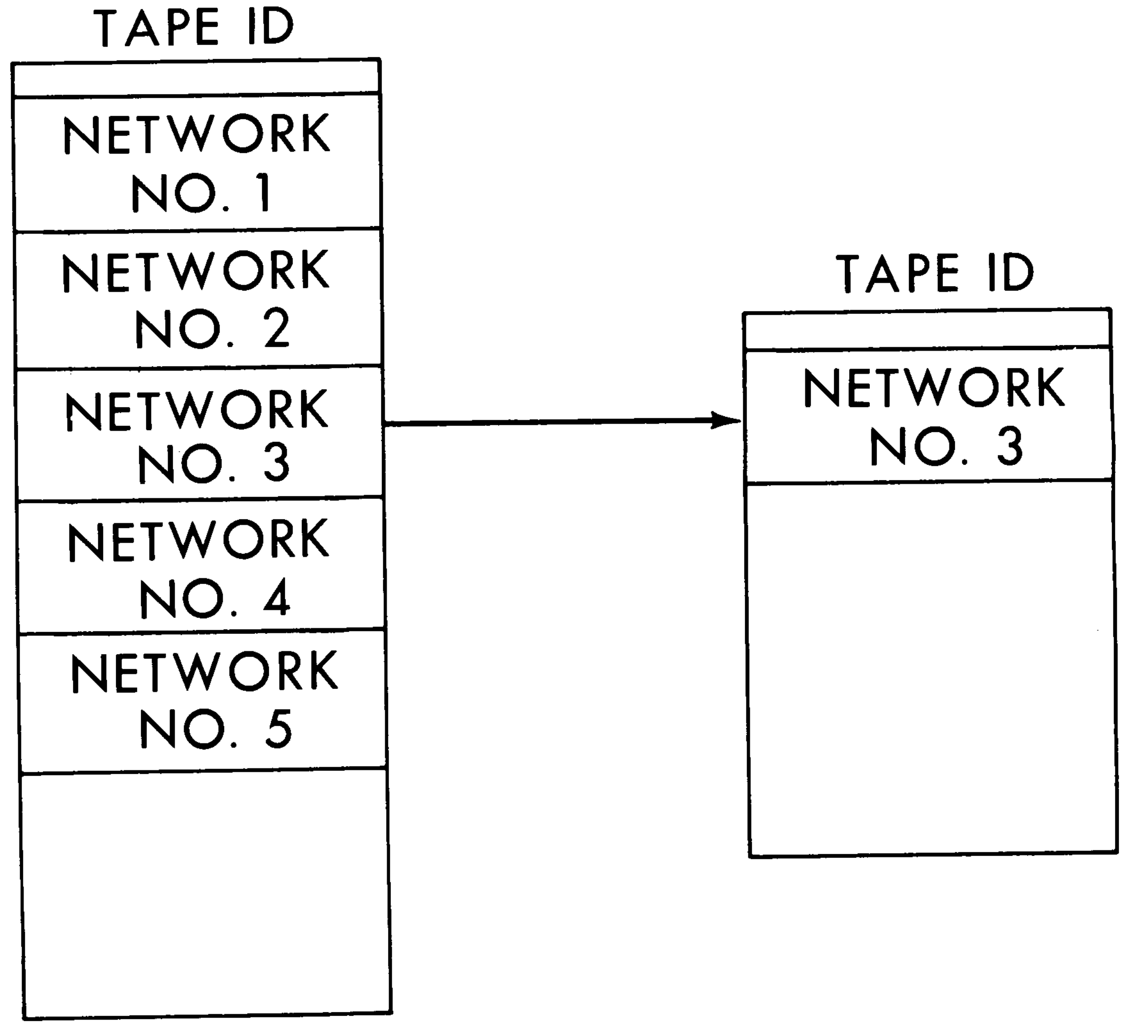


Figure 9

NASA-S-64-4701

ADD

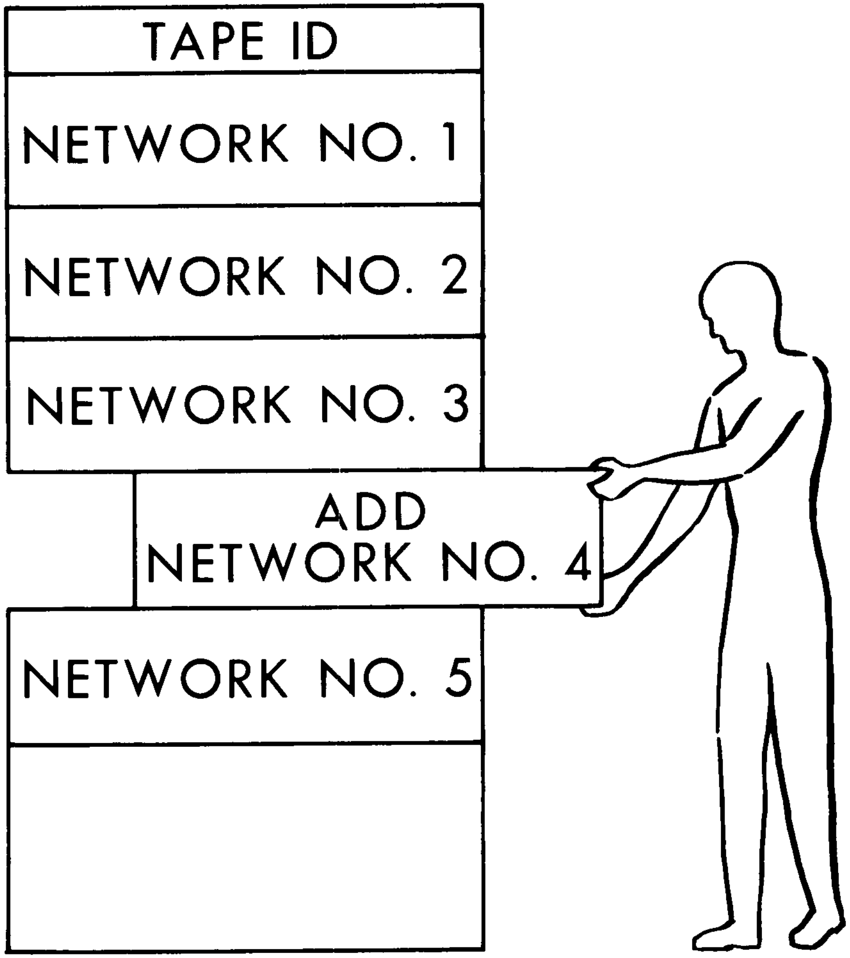


Figure 10

NASA-S-64-4702

CARD TO TAPE AND EDIT

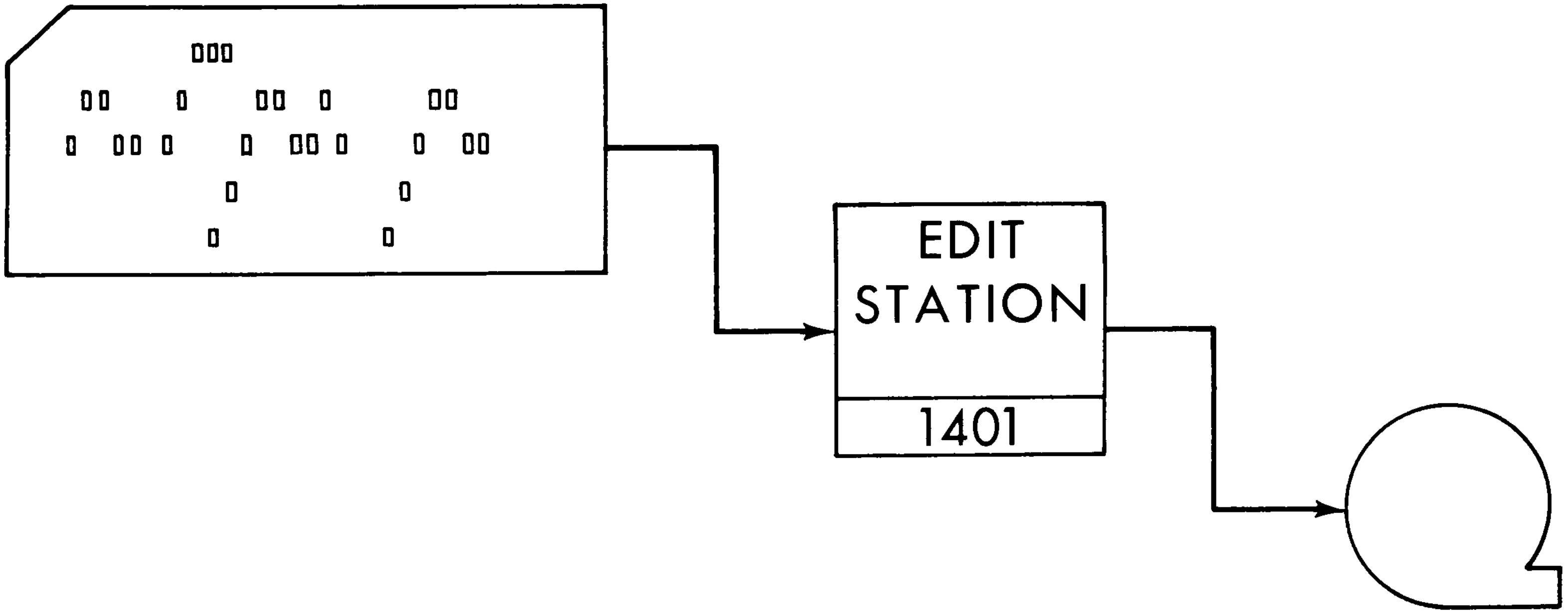
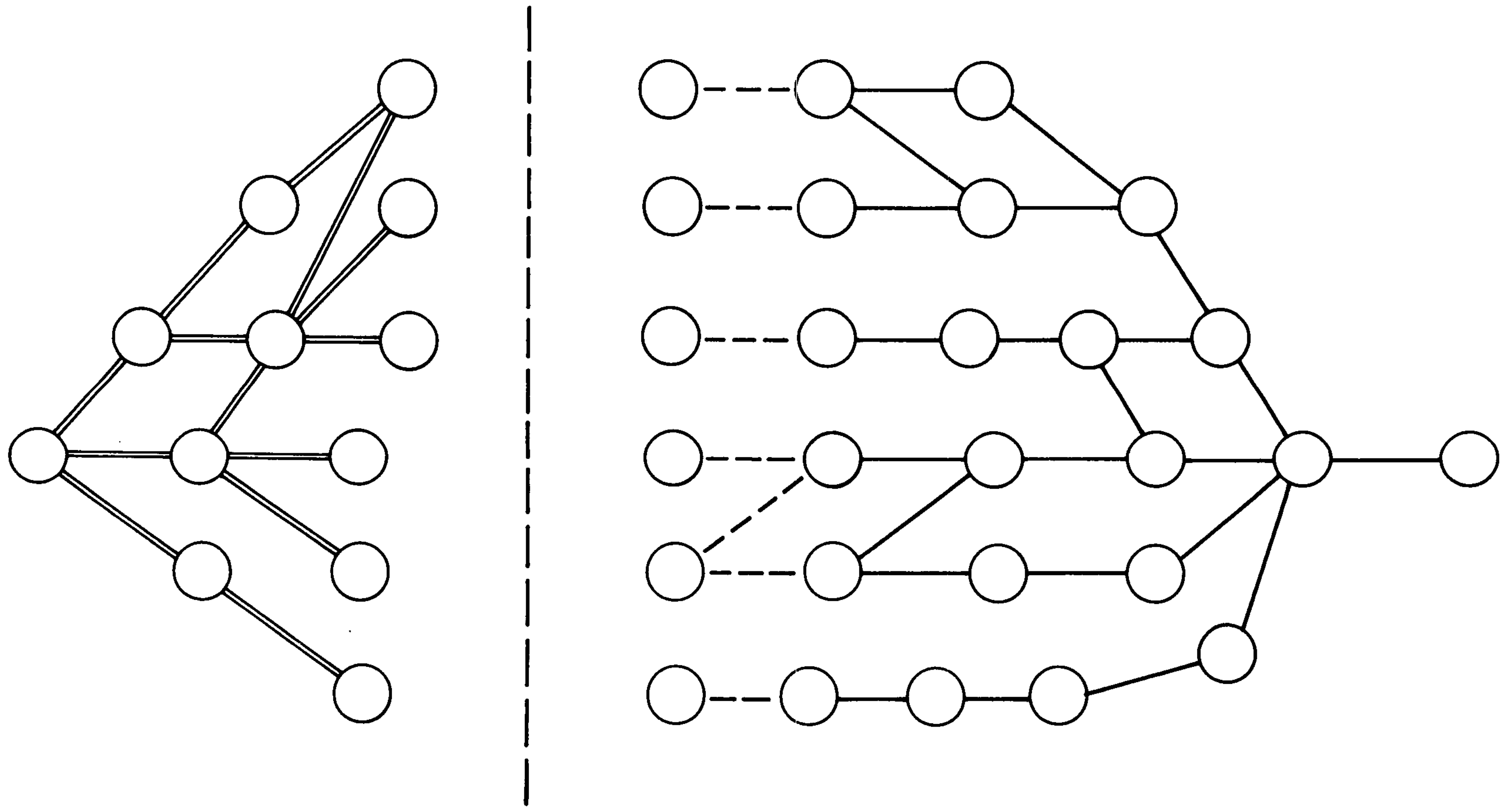


Figure 11

NASA-S-64-4703

DELETE COMPLETED ACTIVITIES



═══ COMPLETED ACTIVITIES
--- PSUEDO START ACTIVITIES
—— OPEN OR INCOMPLETE ACTIVITIES

Figure 12

NASA PERT "C" SYSTEMS LOGIC AND REPORT GENERATION

by

Larry Stevens and John Leonard
NASA Manned Spacecraft Center
Houston, Texas

SYSTEM CONFIGURATION

A. Functions of Each Pass

The NASA PERT "C" Program is organized as 14 passes plus 2 packages of common subroutines. Each of these is a self-loading record on the system tape. Each of the passes overlays the preceding pass and uses the entire core memory.

(Figure 1) illustrates the logical flow of control through the system. (Figure 2 through 17) depict the functions of each pass and its position in the system.

One of the handouts you have received is a timing chart showing the time required in each pass for various size networks. As each pass is discussed, it might be interesting to note the amount of time it requires.

The first record of the tape contains a communications region and several subroutines. This lower core package is referenced by all succeeding passes and remains in core at all times. A list of tapes required is located in this package and all tape references are obtained from this list. The control information for all options is also contained here along with the tape check and on-line printer subroutines. This lower core package does not execute but simply loads pass 1.

The primary function of pass 1 is to read and edit the input tape. This tape is read until an "A" header record is found. The work count of the "A" header record is calculated and the block size for all subsequent read commands is established. Thus PERT "C" will accept input blocked from 1 to 64 activities per record. The ending events of the "A" card are converted to binary and stored. The "B" and "C" cards are then read and the "C" card options are converted to report controls. If the run is a restart of a prior run, control is transferred directly to the appropriate pass; otherwise, the data is read and checked for the following errors:

(1) Sequence Error

(2) Duplicate Error

(3) Predecessor and Successor Event Equal

(4) Zero Event Number

(5) Invalid Code in Column One

(6) Invalid Schedule Date

(7) Invalid Completion Date

If an activity produces any of the first five errors as listed above, the activity is deleted from the run. If an invalid scheduled date occurs, the data is zeroed, and in the case of invalid completion dates, the date of the report is used.

Each data error will produce a diagnostic on the A3 output tape and will turn on the pass 1 error indicator. All error diagnostics may also be printed on-line if desired.

As the data is checked, it is converted to binary, and two output tapes are written. One tape (B-3) contains the activity event numbers and a sequence number, and the other tape (A-4) contains all activity information.

At the completion of pass 1, even though data errors may have been encountered, control is transferred to pass 2. The pass 1 error flag is checked at the completion of pass 4 and if on, the run is terminated at that time.

In pass 2 the first sort takes place. The B-3 tape, containing activity event numbers, is read and sorted into successor-predecessor sequence. These event numbers are sorted 4096 at a time and written on the B-3 output tape, and the next pass is loaded.

Pass 2A now calculates the criteria necessary for assigning topological numbers to the events of the network. A detailed description of the functions of passes 2A and 2B will be included in the discussion on topological ordering.

Pass 2B assigns topological sequence numbers to each event of the network, assembles a four-word record for each activity, and writes it on tape.

A loop diagnostic routine in this pass will locate events in a loop and list the activities off-line. All loops in the network are detected and printed out. If a loop exists, the processing of the network is terminated at this point.

Pass 3 - The purpose of pass 3 is to arrange the activity records into sequence by forward topological order number. Tape A-4 contains activity records in sequence by predecessor and successor events, and B-3 contains abbreviated activity records of event numbers and topological order numbers. Tapes A-4 and B-3 are read concurrently and the topological order numbers are merged with the activity records. These records are then sorted and written on B-3. Control is then transferred to pass 4.

Pass 4 - This pass calculates the expected date of each activity. The activity records are encountered in forward topological order. As an activity record is encountered, its expected date is calculated and posted to a table where the latest date for its successor event is developed. When all successor activities to an event are encountered, the event and its date are vacated from the table, and the vacated cells are used for other events. The program, therefore, continually restricts the table to as small a portion of memory as possible. The events are stored in random order and are retrieved by a scan of the occupied portion of memory, which is kept acceptably small. The activity records, complete with expected date, are written on A-4. As the expected date is being calculated, several options may be exercised. By updating past-due activities or using the time-now option, the expected date calculation may be changed. Also, all completed activities may be deleted at this time.

Pass 5 - This pass rearranges the activities into sequence by backward topological order number, and writes the activities on Tape B-3.

Pass 6 - This pass calculates the latest allowable date of each activity by reversing the technique of pass 4. It also calculates slack for each activity. Pass 6 is performed up to 28 consecutive times, once for each run required according the the "A" header cards. Each execution produces a file of data on tape A-4. However, if no prime reports (these are reports which include all activities of the network) were requested, the file is not produced. Control is then transferred to the second lower core package.

Lower Core Package 2 - This second lower core package is a group of formatting and output routines that remain in core during the report generation phase. As soon as these routines are loaded, control is transferred to Pass 7.

Pass 7 - Pass 7 is the first of the report generation phase. This pass produces a complete report sorted by successor-predecessor event number. Upon entering Pass 7 the output controls are checked; if the report was requested, it is executed; if not, the next pass is loaded. To produce the report, a file of the Pass 6 output tape is read and sorted. At the completion of the sort, the activity information is converted to BCD and formatted for processing on the 1401 and/or 4020. The A-4 tape is positioned for the next pass and control is transferred to either Pass 8 or back to Pass 7 according to the output request.

Pass 8 - Pass 8 first checks the output controls for the status of the report. If no Critical Path Report was requested, control is transferred to Pass 9. To produce the report the activity information is read from A-4 and rearranged in a suitable sequence to be sorted by slack and expected date. At the completion of the sort the activity information is converted to BCD and formatted for processing on the 1401 and/or 4020. As this process takes place the critical path cutoff option may be exercised and the report terminated. Again the A-4 tape is positioned for the next pass and control transferred to the appropriate pass.

Pass 9 - Pass 9 differs from 8 only in that it rearranges the data so that it can be sorted by expected date and successor event to produce the Expected Date Report. If the expected date cutoff option is exercised, the report will not contain all activities of the network. Control is then transferred to either Pass 10 or back to Pass 7, according to the output request.

Pass 10 - Pass 10 differs from 8 only in that it rearranges the data so it can be sorted by scheduled or latest allowable date and successor event number to produce Scheduled or Latest Allowable Date Report. If the latest allowable date cutoff option is exercised, this report will also be terminated before the complete report is produced. Control is then transferred to either Pass 11 or Pass 7.

Pass 11 - Pass 11 differs from 8 only in that it rearranges the data so it can be sorted by department and successor event number to produce the Department report. The Department cutoff option may reduce the volume of information contained in this report. At the end of the report, control is transferred to either Pass 12 or Pass 7.

Pass 12 - Pass 12 uses a different method from the preceding passes to read the data into core. A load channel loop is used that discards all those activities that do not contain a master schedule code. This normally produces great savings in time. For example, on an in-house network of 28,000 activities only 1,000 are coded with master schedule flags. Therefore instead of performing a long sort and multiple merge, only 1 sort must take place. The amount of time saved by this procedure is approximately 12 minutes on the prime run. At the conclusion of the report control is transferred to Pass 7. It is only natural to inquire about the flexibility of a common system such as this.

B. Incorporation of Additional Report Generators

For some reason, either by necessity or just plain human nature, almost every user of a common system will want to develop his own unique innovations. Probably one of the most important and necessary characteristics of a common system is its flexibility and the ease with which modifications can be made. NASA PERT "C" is flexible in many respects. Tape and channel assignments can be changed very easily. The tape addresses are stored in a table in lower core and a change in this table is reflected in all passes of the program. However, it would be an overly optimistic statement if it were said that everyone here could easily add his own report generator to the PERT "C" system. In NASA PERT "A", this was a simple task. However, due to the many combinations of optional reports that have been incorporated since PERT "A" was distributed 2 years ago, the addition or replacement of a report generator is now more difficult. For this reason, special detailed documentation has been developed to assist users in the modification, addition, or replacement of report generators. Included is a detailed description of how to establish the system control areas necessary to maintain continuity within the system. For example, to add an additional report sorted by Department, Latest Allowable Date, and Successor Event Number, an update of 125 cards was required.

These changes would produce a new pass under "C" card option control. This completes the discussion on systems configuration. Next, I would like to discuss operational procedures using PERT "C". There are several methods of operation that will best utilize the capabilities of this program.

OPERATIONAL PROCEDURES

A. Standard or Total Run

A normal production run will consist of the A, B, and C header cards and the activity data. The A and C control cards should contain all necessary data to request calculation options, reports, report cutoffs, and end runs. The A card contains the date of the report which appears on every page of output and is used as the base date for Time-Now calculations. It also contains the event numbers to be used for End Runs. The C card controls the production of all reports. If all input data is correct, the program will perform the requested operations and then return control to the monitor system mounted on A-1. It is recommended that 2 intermediate tapes be saved from all long runs. These tapes will contain all network information necessary to restart the job for additional output.

B. Restart

The PERT "C" restart feature provides the capability of restarting a job at any one of 4 different points. The input requirements for a restart are: an intermediate tape from a previous run, and new A, B, and C header cards. There are many reasons for using the restart feature:

(1) It provides a method of correcting certain types of data errors.

(2) Since it is sometimes difficult to obtain time for a long run, by using the restart feature, a run can be broken into several smaller runs.

(3) Additional reports other than those requested on the original run may be obtained.

(4) Additional end runs other than those requested on the original run may be obtained (see Figure 18).

To correct certain data errors, the program can be restarted with pass 5, using tape A-4, the output tape from pass 4.

A job can be stopped at the completion of any pass from 3 through 12 and restarted at a later date. The output tape from the last pass executed should be saved for the restart. This will allow a job to be broken into many smaller jobs. The total time for the job would be only seconds longer than if the job had been processed completely at one time.

If additional end runs are required other than those requested on the original run, the pass 5 output tape should be used to restart pass 6. The event numbers on the new A control cards will be used for the end runs. If possible, all prime run reports should be eliminated to take advantage of the pass 6 capability of only producing data when requested.

By restarting at pass 7, using the pass 6 output tape, additional copies of selected reports can be produced very inexpensively. The data for the report has already been produced and only a report generator phase

NOTES

paths with the same slack. When the count of the slack changes has reached the number specified on the "C" control card, the report generation is stopped, and control is immediately transferred to the appropriate pass.

Figure 22 shows the paths that would be retained if a cutoff of 3 paths was used.

Portions of the Expected Date Report may be deleted during report generation. Any two-digit number, in a specified field of the "C" card, is added to the "date of the report". This produces a cutoff date, the given number of weeks past the date of the report. When the expected date of an activity exceeds the cutoff date, the report is terminated and control is transferred to the appropriate pass.

Figure 24 shows those activities that would be deleted.

The Latest Allowable Date Report has a cutoff similar to that of the Expected Date Report. The only difference is the use of another field on the "C" card and the Latest Allowable Date for the cutoff criterion.

Those activities deleted, using the latest allowable date cutoff are shown in Figure 26.

The Department Report produces a new page each time the department code changes.

The report can be limited to only those activities that contain a department code. The activities with a blank field will, on option, be deleted from the report (see Figures 27 and 28).

The Master Schedule Report is produced in a unique format. A portion of the information is deleted to produce an 8 x 11 form. Only those activities that contain a Master Schedule Flag will appear on the output.

By using the "C" card controls, the information may be sorted three ways. This could place the same activity on three pages. The standard report sorts on columns 2 and 3.

Figures 29 and 30 show Master Schedule 02 and 12 each with 2 activities. This report may be changed so that only column 2 or only column 3 is used as the sort field. This has the effect of producing a multiple level Master Schedule Report. If either column is not used, an asterisk will appear in its relative position in the heading.

is necessary to create a report. When processing large networks, it is recommended that only one or two reports be processed initially. The reports should be checked for input validity and, if correct, a pass 7 restart executed to provide the remaining requested reports. A pass 6 restart would then be executed to provide any needed end runs. This method, would hold to a minimum the number of times pass 6 would be executed. On long runs this is an important factor, since pass 6 may consume between 10 and 25 percent of the time for a total run.

OUTPUT REPORTS

So far this paper has been confined to the system used to generate the PERT reports. We would now like to describe the various reports and calculation options available. By using the C control card, the user has complete flexibility in the selection of reports. All reports produced by PERT "C" are available on an optional basis. Each may be produced on a BCD tape suitable for obtaining tabular output and/or a binary tape for obtaining microfilm output. There are several reasons for using microfilm output:

(1) Microfilm processing is less expensive.

(2) It is much easier to handle and store microfilm.

(3) Using a microfilm viewer, the analyst has access to the same information as he would from hard copy.

(4) Output which takes 3 to 4 hours to print on a 1401 can be processed on microfilm in about 30 minutes.

(5) Eight and a half by eleven hard copy can be obtained from microfilm quickly and inexpensively.

Figure 19 depicts the network used to produce the output reports shown in Figures 20 through 31.

The Successor-Predecessor Report will always contain all activities of the network except when the option to delete completed activities has been exercised. The format is the standard form for the first 5 reports.

The Paths of Criticality Report is sorted by Slack, Expected Date, and Successor Event. The volume of output on the Paths of Criticality Report may be reduced by stopping the report as soon as a specified number of paths have been produced. The count is not of true slack paths but of changes in slack. It is possible to have several

Figure 31 shows the report produced by sorting only on column 3. The report contains those activities of the 02 report and the 12 report.

B. Report Options

Two report options are available which will affect the calculation of the expected date:

(1) The Updating of Past Due Activities

The first updating of past due activities will insure that all dates produced are future ones. If the expected date of an activity is prior to the date of the report, the expected date is set equal to the date of the report. These activities will appear on a report titled "List of Past Due Activities". (Figure 31A)

Note in Figure 32 the date of the report is 7/1/64. By standard PERT calculations, the expected date of activity 2 to 3 would be 6/22/64 which is prior to the date of the report. By activating the update expected date option this date would be changed to 7/1/64. In either case the expected date of activity 3 to 4 would be calculated by adding the time estimate to the expected date of event 3.

(2) "Time Now" Method

A "Time Now" method may also be used in calculating the Expected Date of all activities in progress. If all predecessors to an activity are complete, then the Expected Date of the activity is calculated by adding the time estimate to the date of the report instead of the latest completion date of the preceding activities. The time estimate is then, by definition, the amount of time remaining for the activity instead of the total time of the activity.

An example of the "time now" calculation is shown in Figure 33. By standard PERT calculation, activity 3 to 4 has an Expected Date of 7/20/64. If the same date was run on "time now", the 4-week-time estimate of activity 3 to 4 would be added to 7/6/64, the date of the report, to produce a date of 8/3/64. Actually the time estimate should be updated to be run on "time now", as shown with the time estimate of 2 weeks producing a date of 7/20/64.

C. Deletion of Completed Activities

The computation and printing time for generating a report can be reduced by deleting the actual activities from the network. This can be accomplished several ways. As a network is being processed on the 7094, all completed activities may be deleted in pass 4. These activities are simply not placed on the output tape at the completion of computation.

If the network shown was processed without deleting completed activities, the first print out would be produced. The second is produced by deleting completed activities. If the actuals are deleted, the first incomplete activity of each path is flagged with the letter "B" before the Expected Date on all output reports. This provides the analysts a beginning actitivy for each path. (See Figures 36 and 37).

A portion of the completed activities may also be deleted from the input prior to a 7090 run. A tape containing all activities for a network is required as the input to the program.

The first step is to delete all completed activities from the tape and produce a tape containing only the completed activities (See Figure 38).

A tape containing all events that have an incomplete successor activity and a completed predecessor activity is generated. This Event Tape is than read to form a table and the successor event of every completed activity is matched with each event in the table. As shown in Figure 39, those activities that match are added to the new network tape.

If there are less than 750 of these added activities, the network is ready to be processed. (The 750 figure is due to the maximum number of starting events that may be accepted in the PERT "C" program). If more than 750 activities are added to the network, they are tied back together with dummy activities (see Figure 40 and 41).

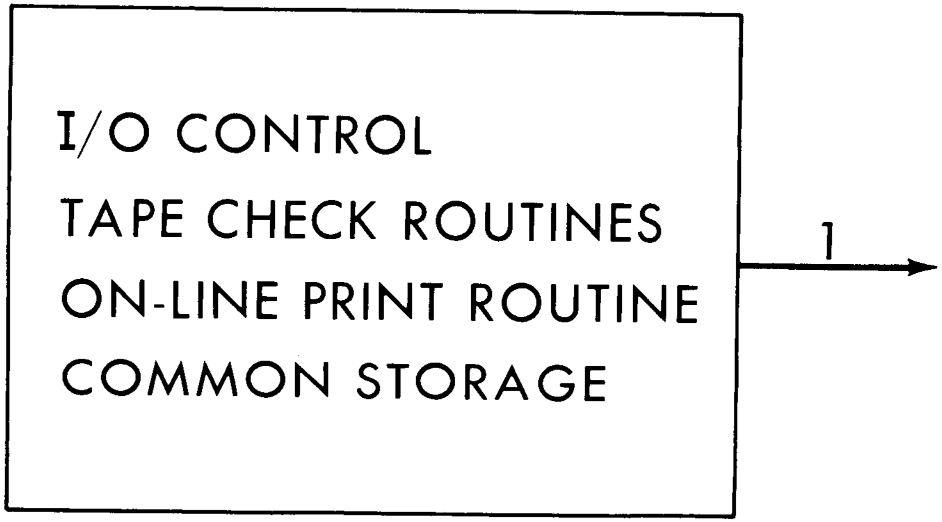
This is done by producing dummy start events whose successor events are the predecessor events of the added completed activities. A single dummy predecessor event is set up for each subnet, therefore, reducing the number of start events to the number of subnets. This method for deleting completed activities would not be used each time the network is processed -- only when a significant percentage of the network was completed.

NETWORKING TECHNIQUE

In addition to the optional reports, calculation options and condensing methods contained in PERT "C", a new networking technique is available. A new transaction code, a "Code 4", has been developed.

NASA-S-64-4663

LOWER CORE PACKAGE 1


I/O CONTROL

TAPE CHECK ROUTINES

ON-LINE PRINT ROUTINE

COMMON STORAGE

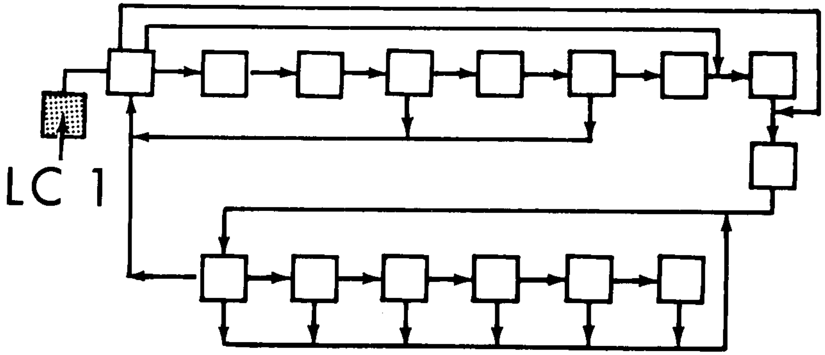
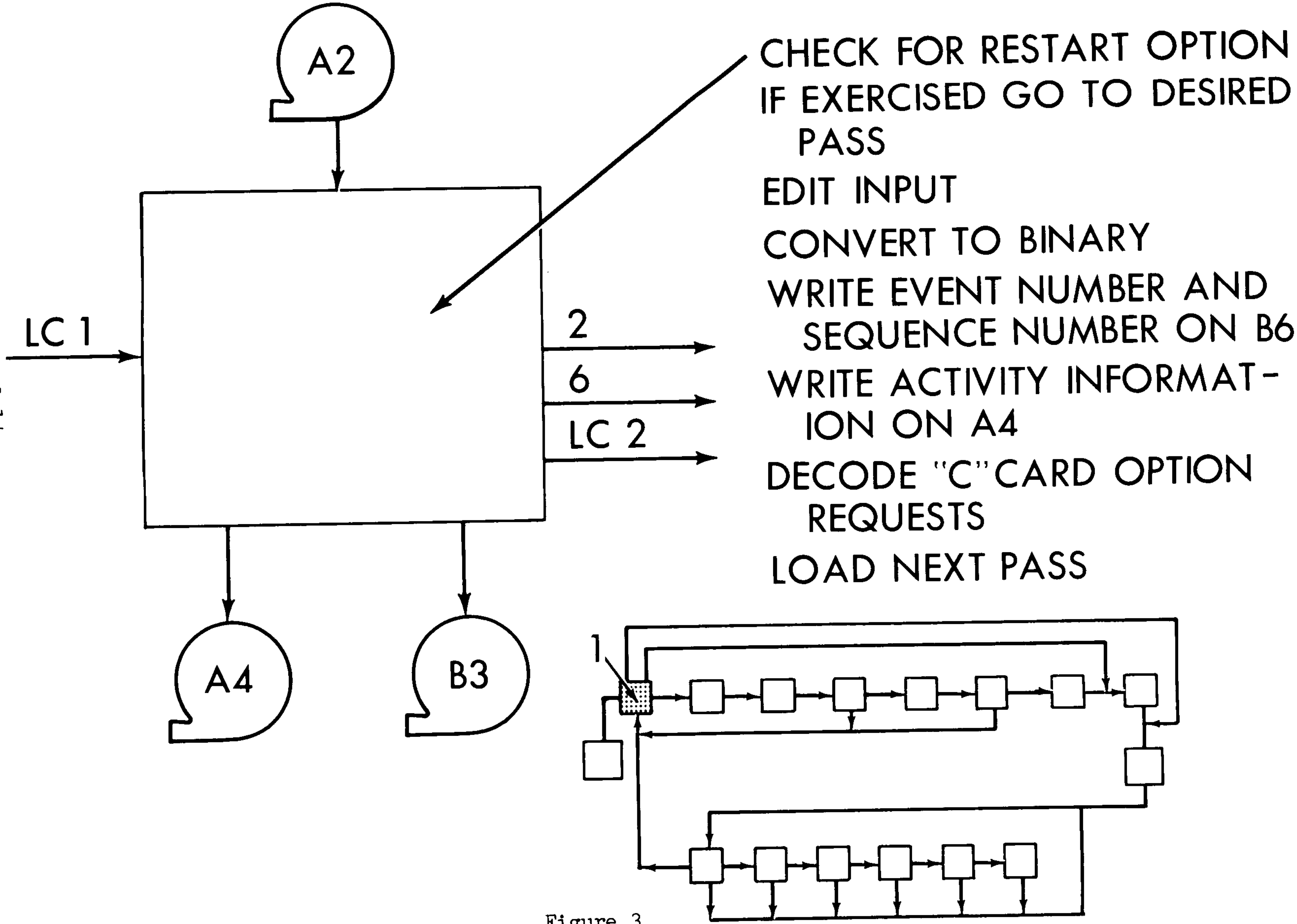


Figure 2

NASA-S-64-4638

PASS 1



CHECK FOR RESTART OPTION

IF EXERCISED GO TO DESIRED PASS

EDIT INPUT

CONVERT TO BINARY

WRITE EVENT NUMBER AND SEQUENCE NUMBER ON B6

WRITE ACTIVITY INFORMATION ON A4

DECODE "C" CARD OPTION REQUESTS

LOAD NEXT PASS

Figure 3

A code 4 activity is entirely different from either a 1, 2, or 3 coded activity in that its status depends on its use within the network. There are several uses of the Code 4 activity. When all interface activities are Code 4, it is very easy to break a network into its respective fragnets to be processed individually. The Code 4 activities in the merge network would have predecessors and would process as incomplete activities. In the demerge, the interface activities should appear in both the sending and the receiving subnet. In the sending subnet, the interface will be an ending activity and the date will be used as a Schedule Date. In the receiving subnet, there will be no predecessor activity and the date will be used as a Scheduled Start Date. This will allow the network to run without producing an "incomplete activity has no predecessor" diagnostic.

Code 4 activities may be used to start a path at some time in the future. In this capacity it is sometimes referred to as a Directed Start Date. In all cases where a Code 4 activity has no predecessors, the date will be used as a Scheduled Start Date, and the letter "S" will be placed before the date on the reports.

NASA-S-64-4637

PERT "C"

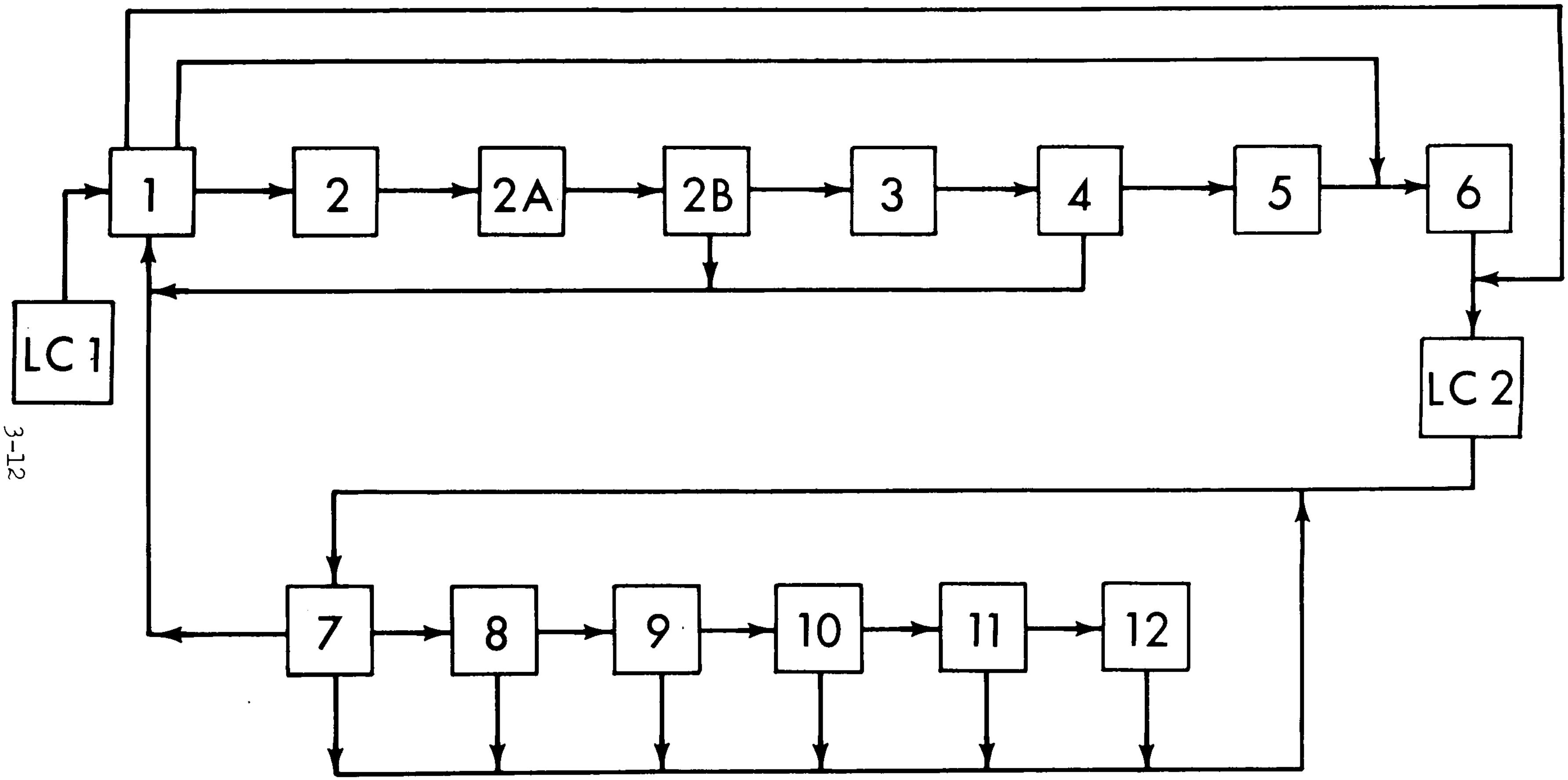
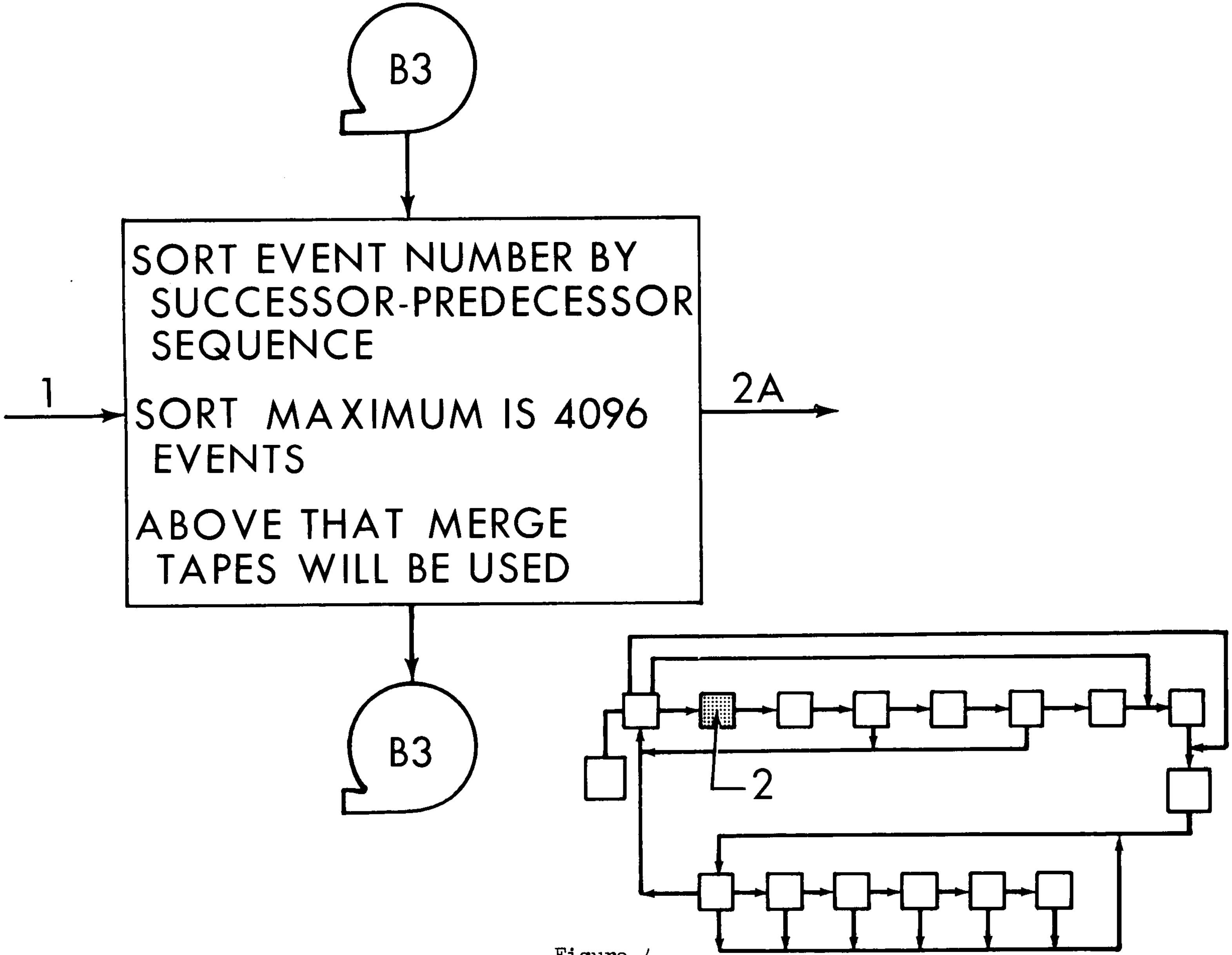


Figure 1

NASA-S-64-4639

PASS 2



SORT EVENT NUMBER BY SUCCESSOR-PREDECESSOR SEQUENCE

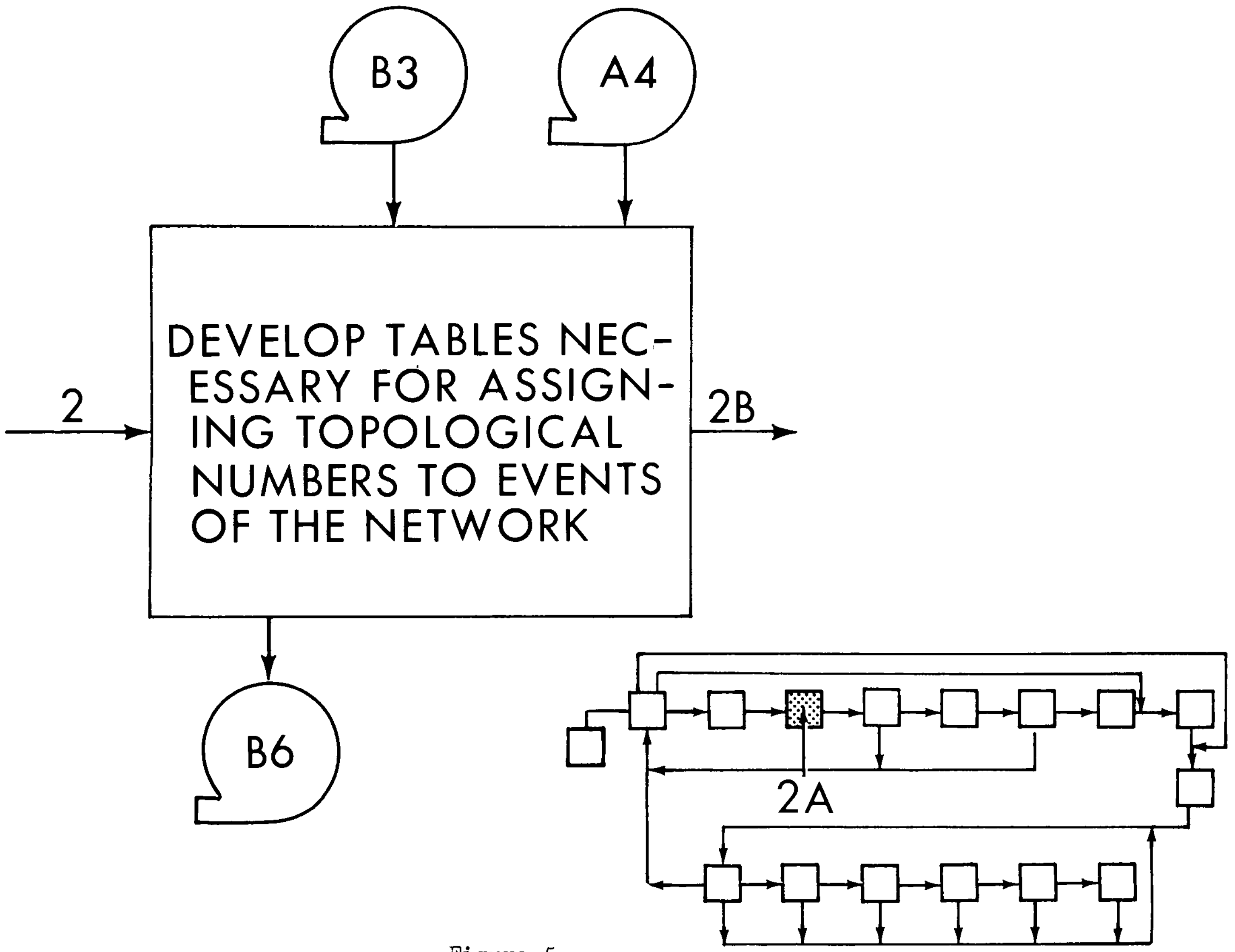
SORT MAXIMUM IS 4096 EVENTS

ABOVE THAT MERGE TAPES WILL BE USED

Figure 4

NASA-S-64-4640

PASS 2A

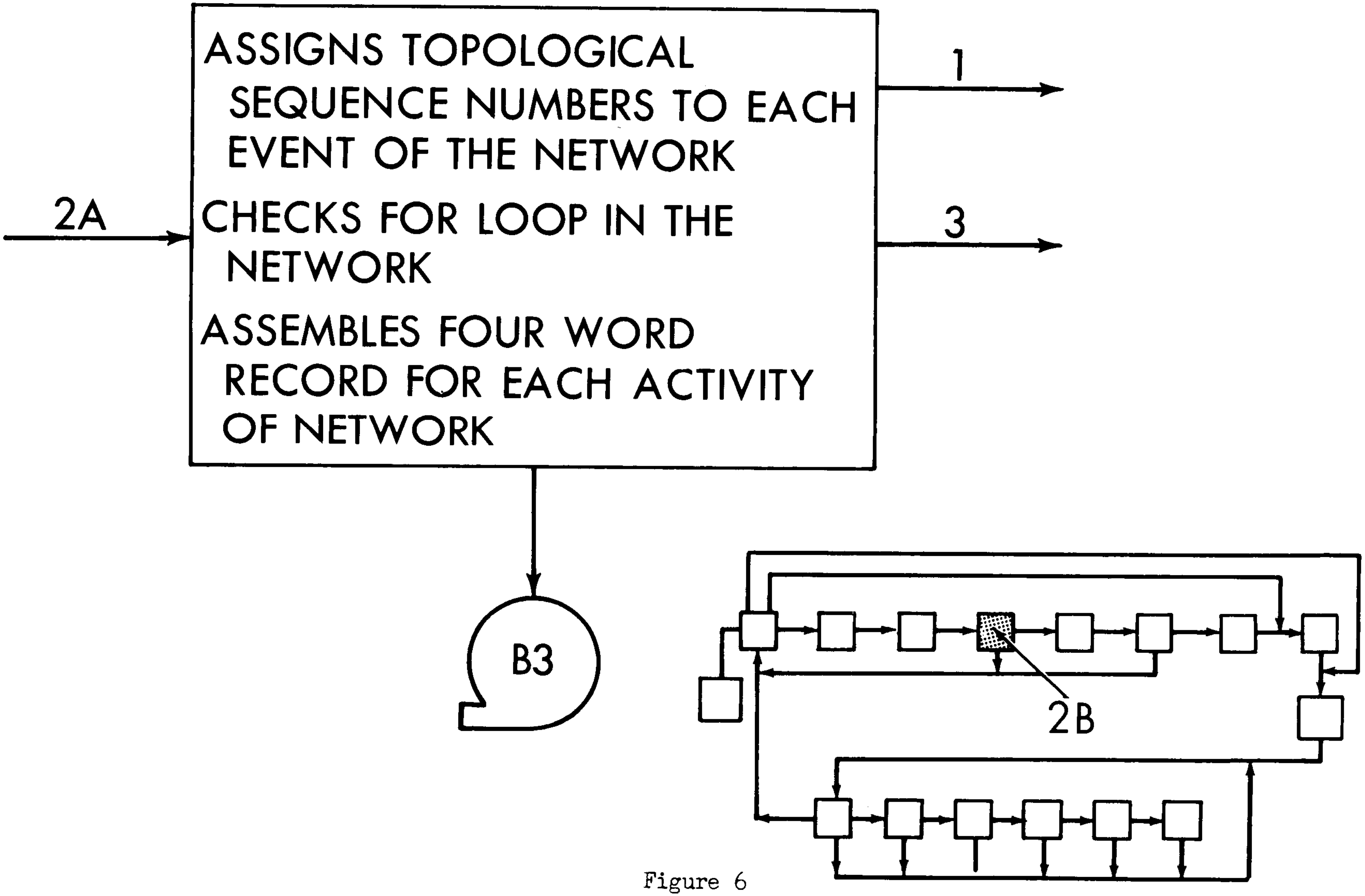


DEVELOP TABLES NECESSARY FOR ASSIGNING TOPOLOGICAL NUMBERS TO EVENTS OF THE NETWORK

Figure 5

NASA-S-64-4641

PASS 2B



ASSIGNS TOPOLOGICAL SEQUENCE NUMBERS TO EACH EVENT OF THE NETWORK

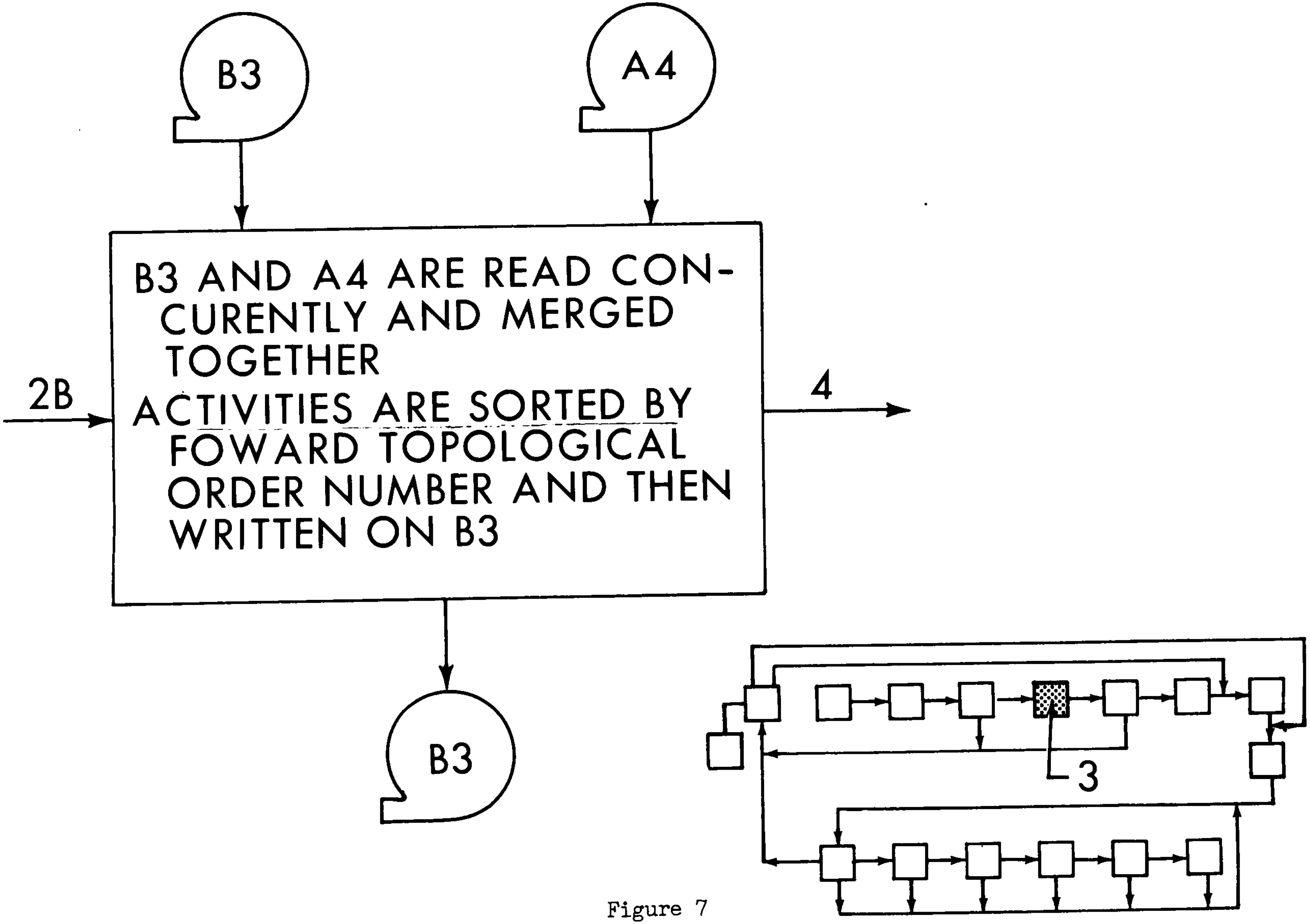
CHECKS FOR LOOP IN THE NETWORK

ASSEMBLES FOUR WORD RECORD FOR EACH ACTIVITY OF NETWORK

Figure 6

NASA-S-64-4642

PASS 3



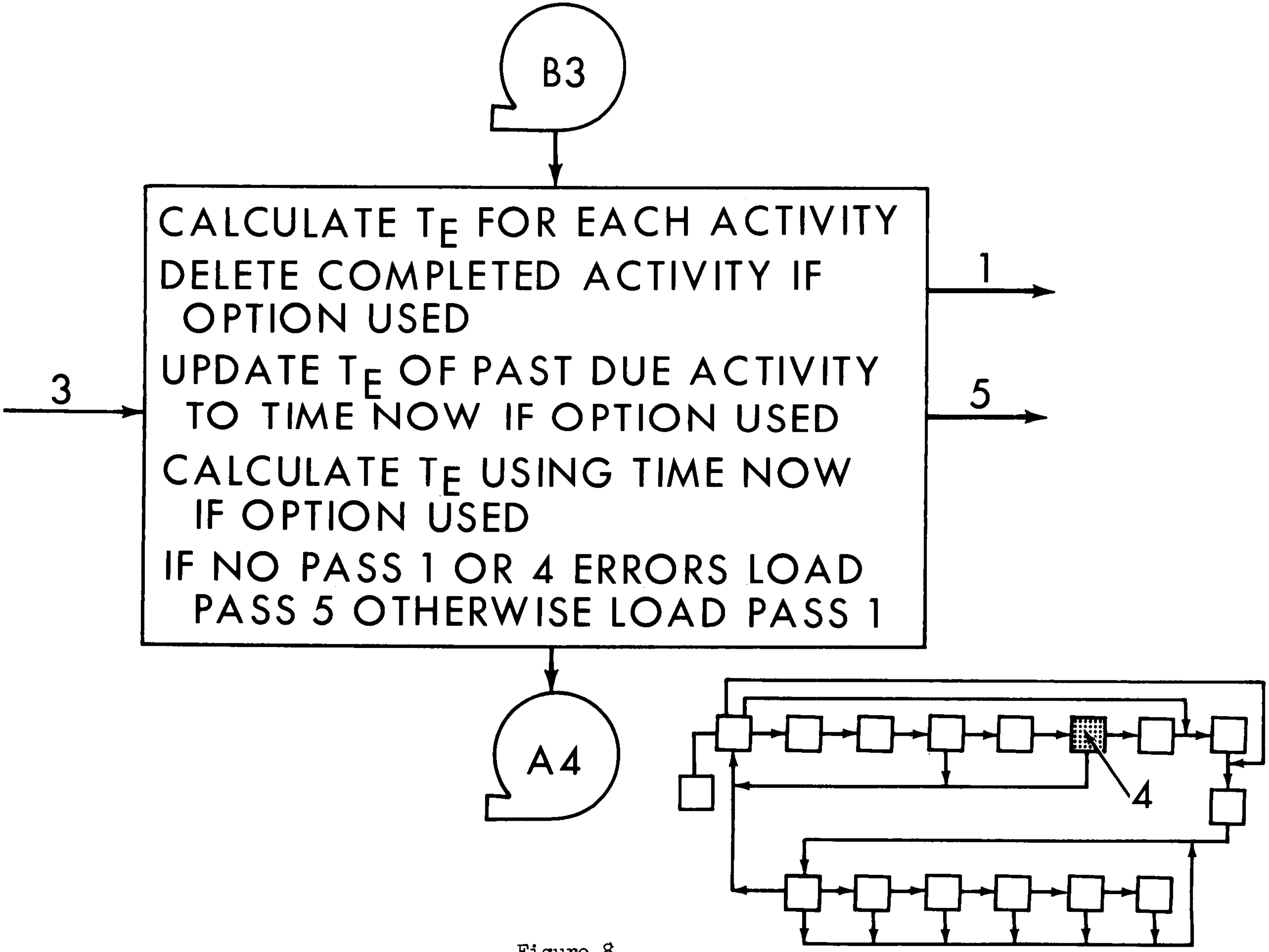
B3 AND A4 ARE READ CONCURENTLY AND MERGED TOGETHER

ACTIVITIES ARE SORTED BY FOWARD TOPOLOGICAL ORDER NUMBER AND THEN WRITTEN ON B3

Figure 7

NASA-S-64-4643

PASS 4



CALCULATE T_E FOR EACH ACTIVITY

DELETE COMPLETED ACTIVITY IF OPTION USED

UPDATE T_E OF PAST DUE ACTIVITY TO TIME NOW IF OPTION USED

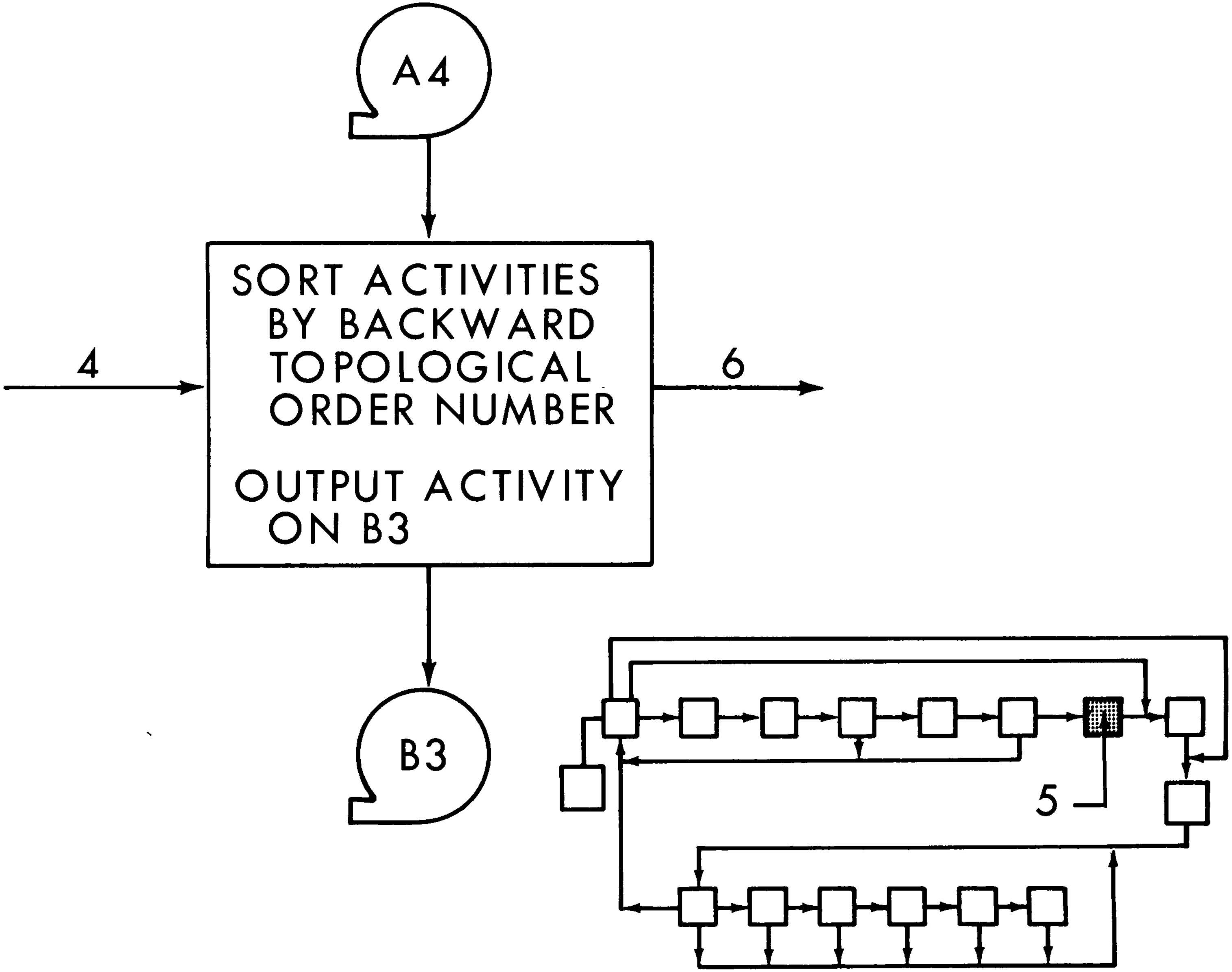
CALCULATE T_E USING TIME NOW IF OPTION USED

IF NO PASS 1 OR 4 ERRORS LOAD PASS 5 OTHERWISE LOAD PASS 1

Figure 8

NASA-S-64-4644

PASS 5



SORT ACTIVITIES BY BACKWARD TOPOLOGICAL ORDER NUMBER

OUTPUT ACTIVITY ON B3

Figure 9

NASA-S-64-4645

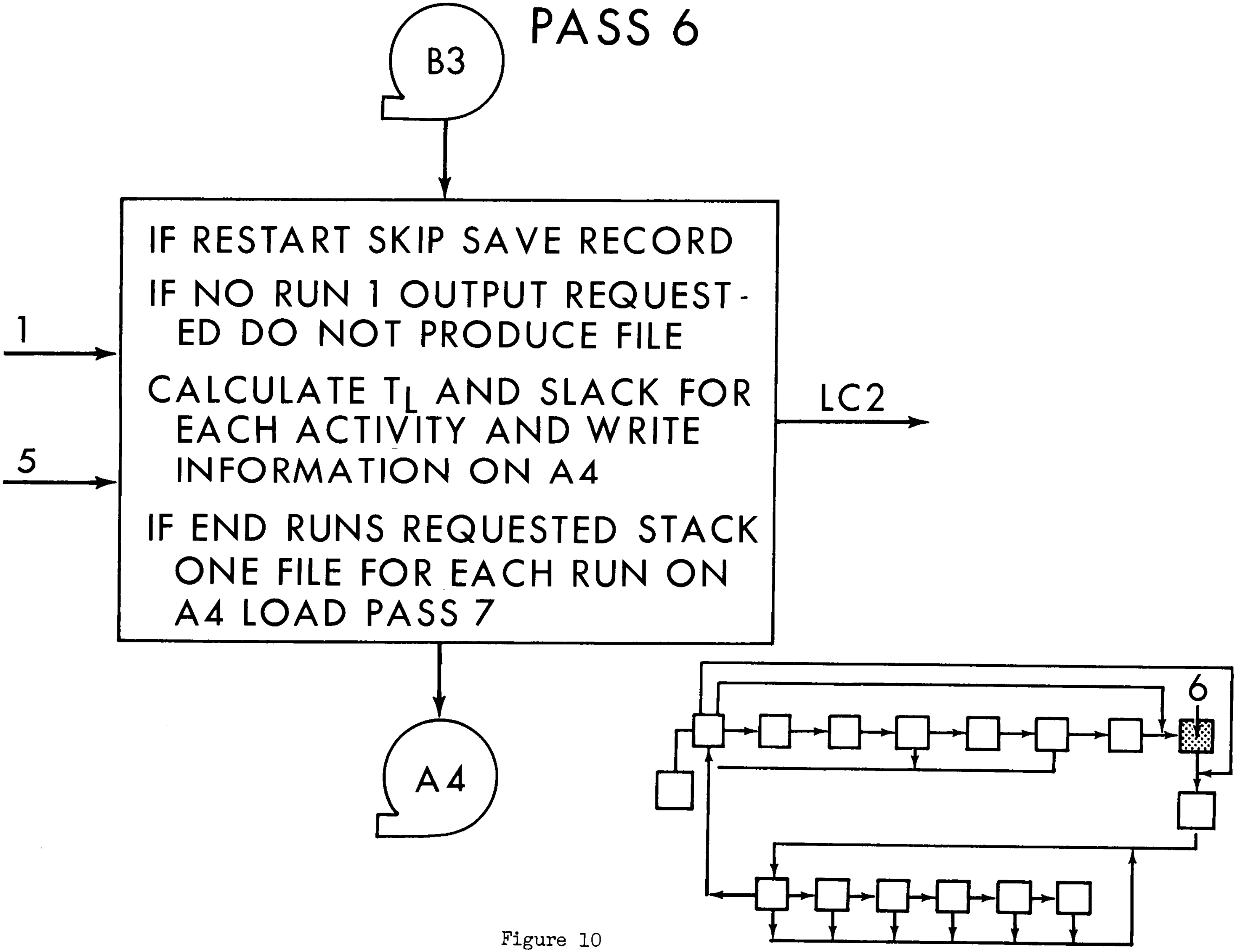
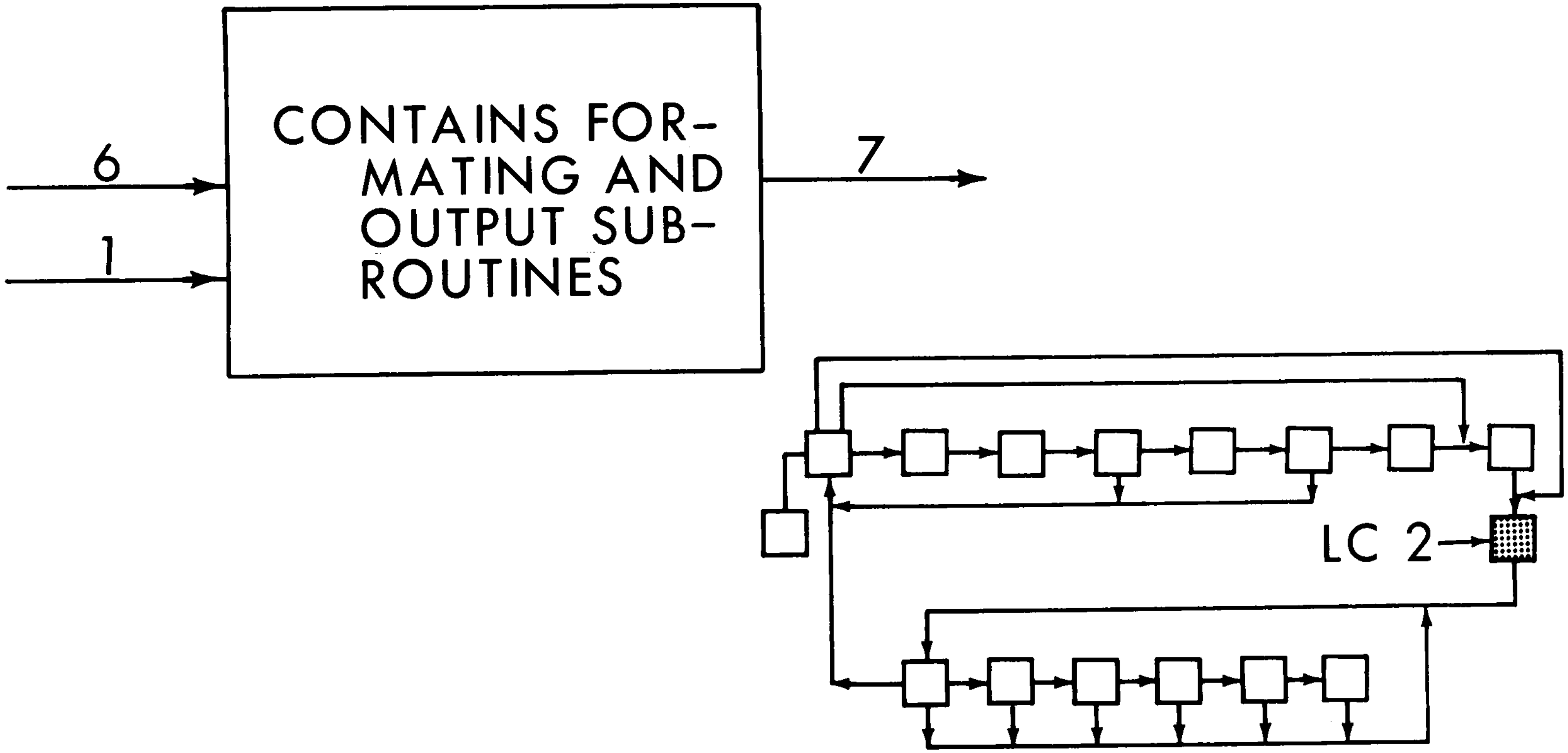

PASS 6
B3
1
5
IF RESTART SKIP SAVE RECORD
IF NO RUN 1 OUTPUT REQUESTED DO NOT PRODUCE FILE
CALCULATE T_L AND SLACK FOR EACH ACTIVITY AND WRITE INFORMATION ON A4
IF END RUNS REQUESTED STACK ONE FILE FOR EACH RUN ON A4 LOAD PASS 7
LC2
A4
6

Figure 10

NASA-S-64-4646

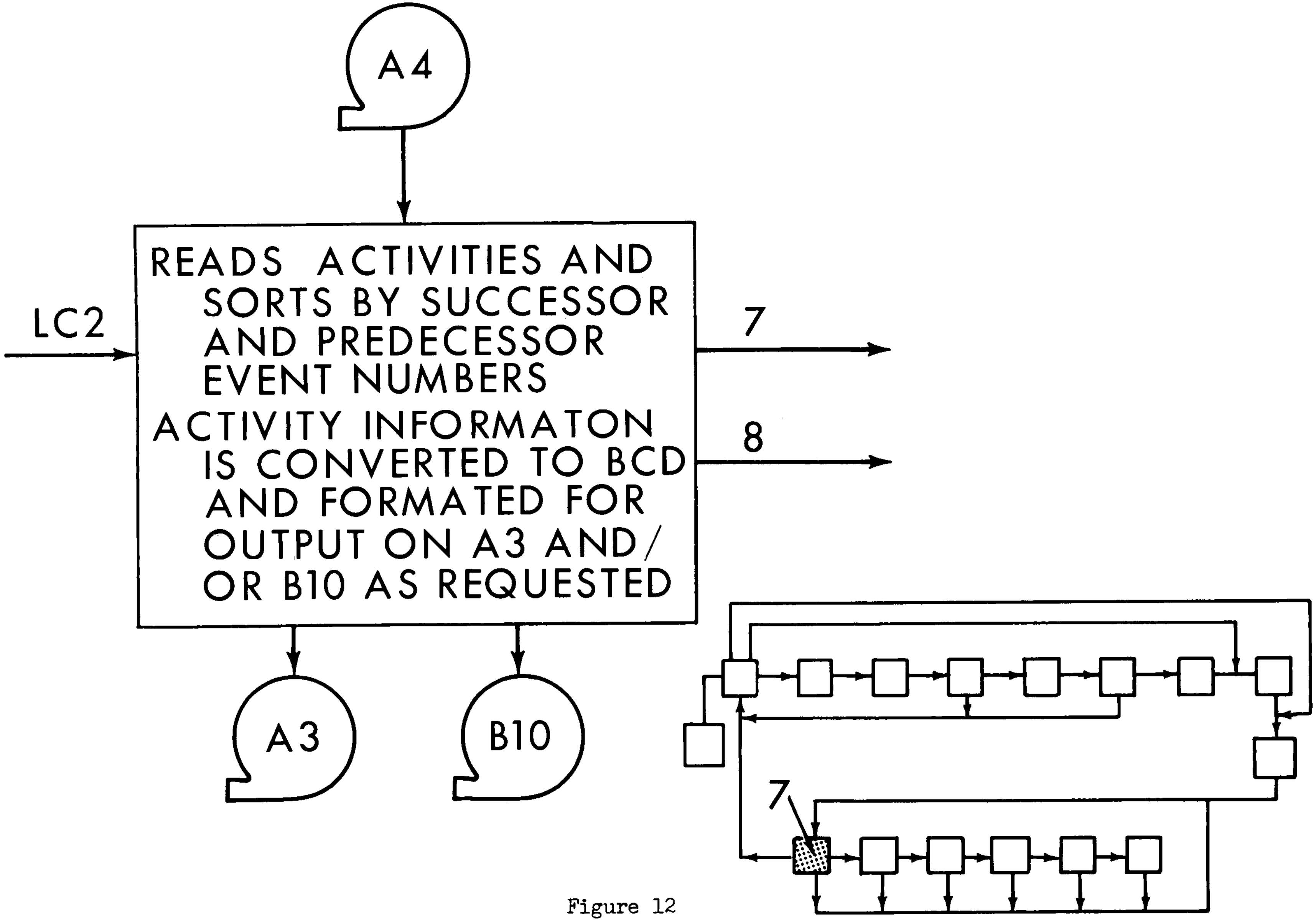
LOWER CORE PACKAGE 2



CONTAINS FORMATING AND OUTPUT SUBROUTINES

Figure 11

PASS 7



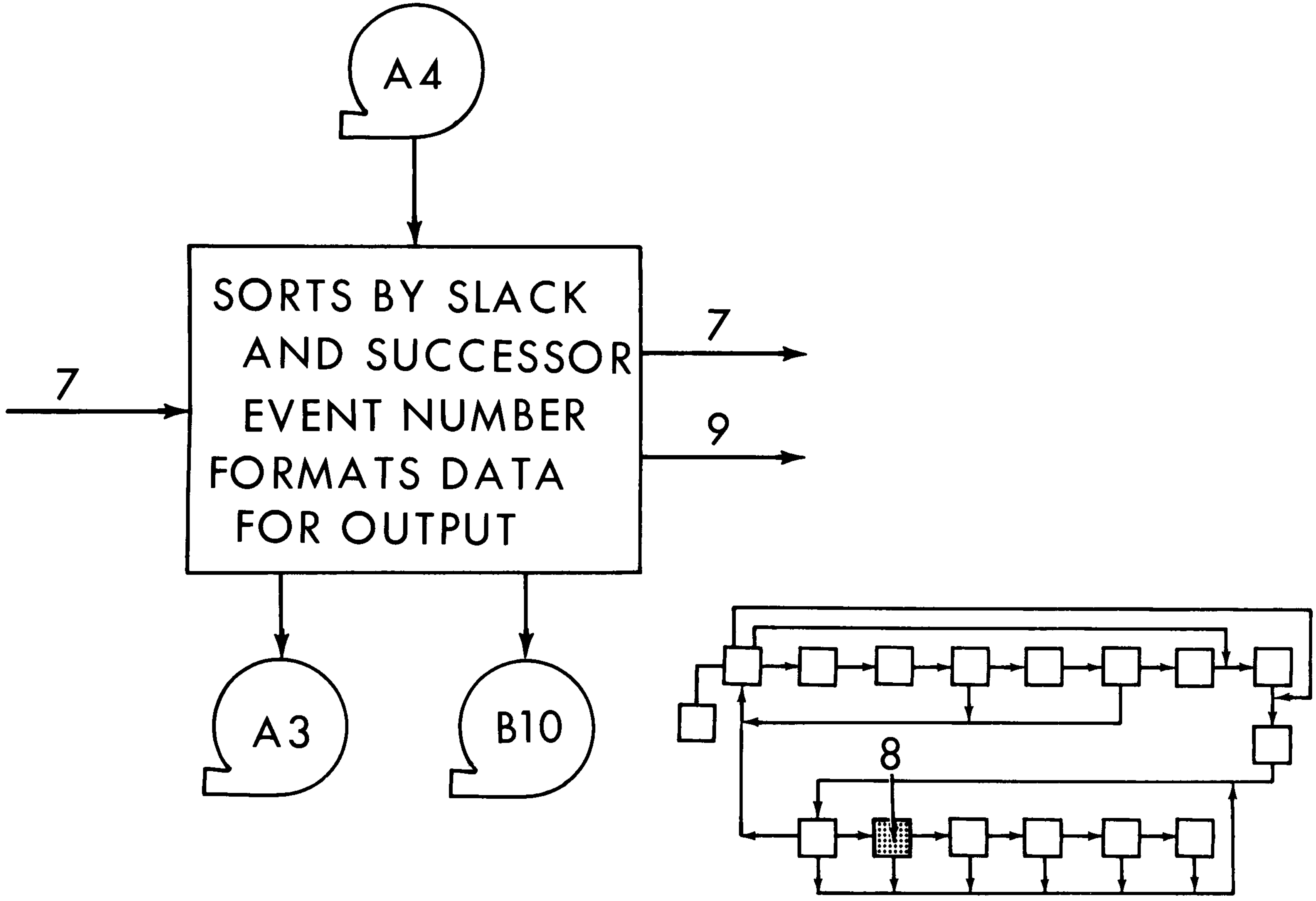
READS ACTIVITIES AND SORTS BY SUCCESSOR AND PREDECESSOR EVENT NUMBERS

ACTIVITY INFORMATON IS CONVERTED TO BCD AND FORMATED FOR OUTPUT ON A3 AND/ OR B10 AS REQUESTED

Figure 12

NASA-S-64-4648

PASS 8

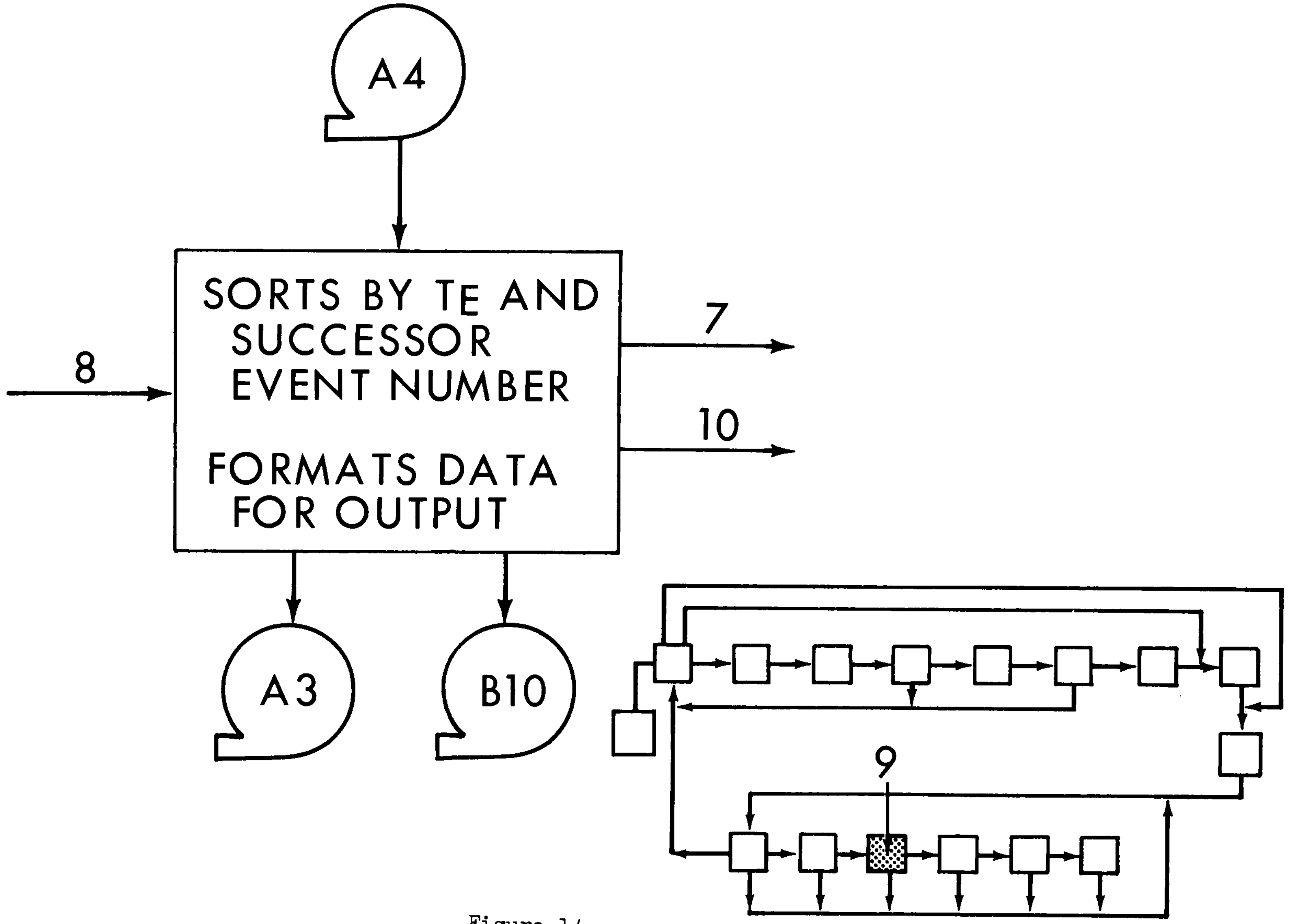


SORTS BY SLACK AND SUCCESSOR EVENT NUMBER FORMATS DATA FOR OUTPUT

Figure 13

NASA-S-64-4649

PASS 9



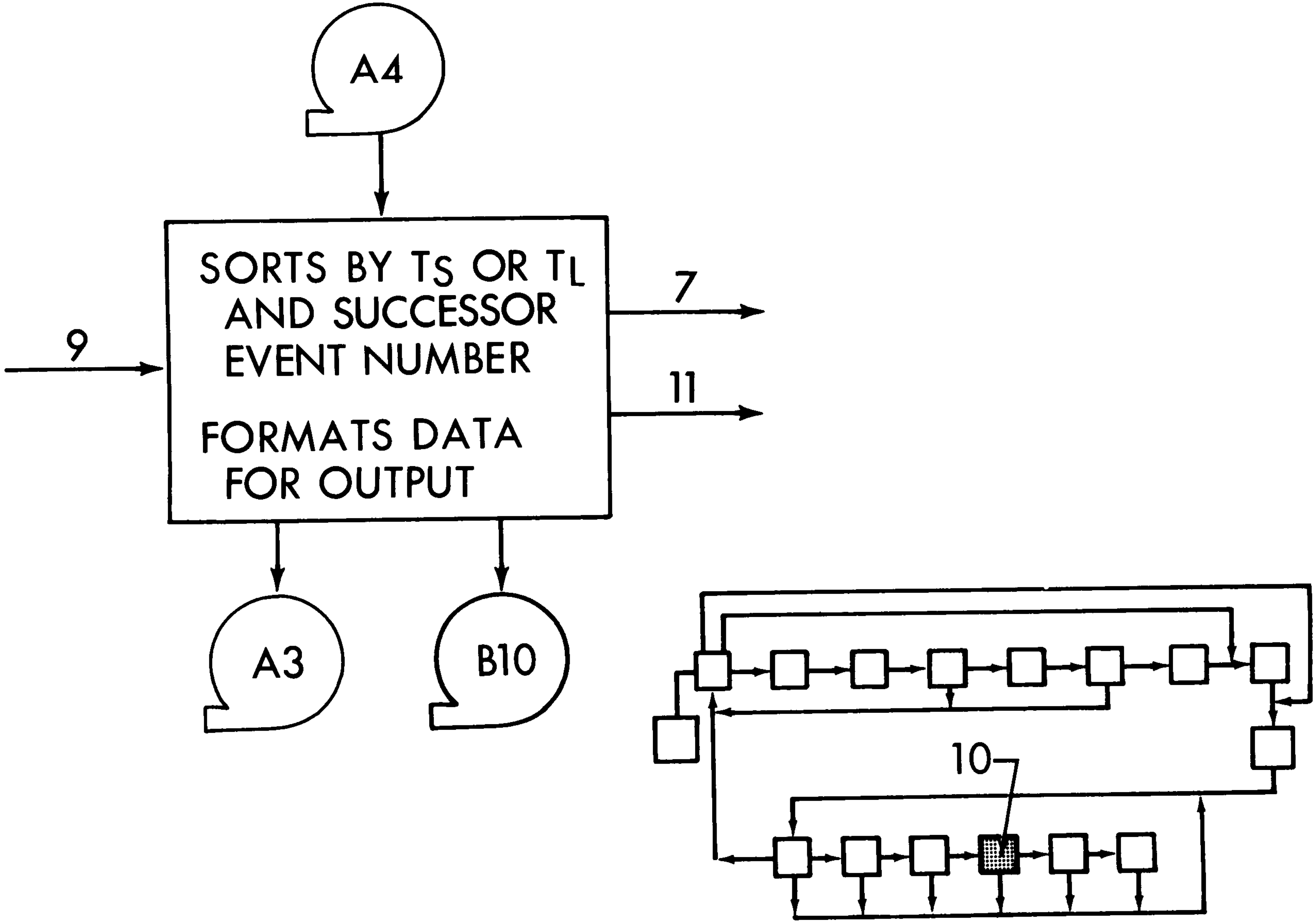
SORTS BY T_E AND SUCCESSOR EVENT NUMBER

FORMATS DATA FOR OUTPUT

Figure 14

NASA-S-64-4650

PASS 10



SORTS BY T_S OR T_L AND SUCCESSOR EVENT NUMBER

FORMATS DATA FOR OUTPUT

Figure 15

NASA-S-64-4651

PASS 11

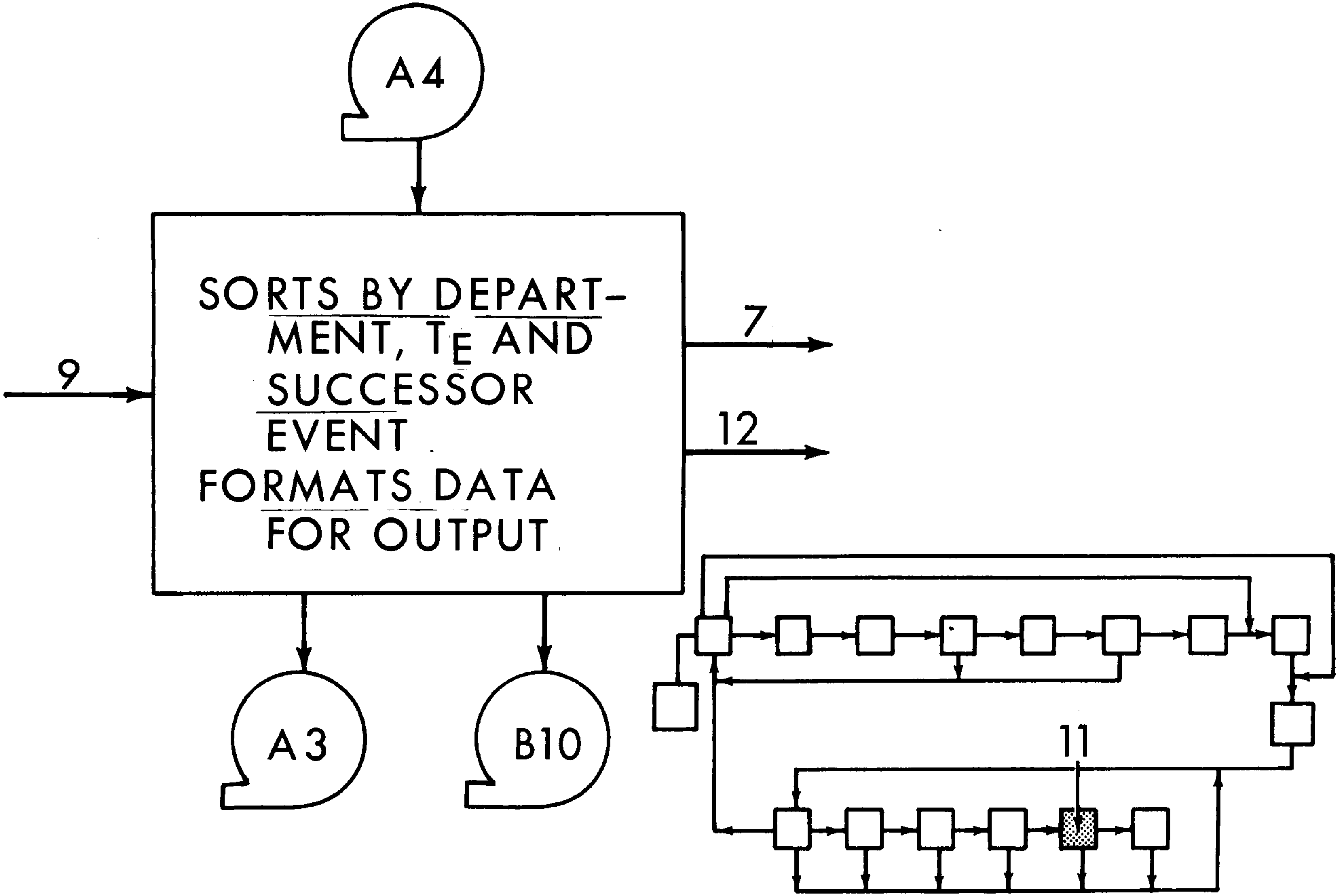


Figure 16

NASA-S-64-4652

PASS 12

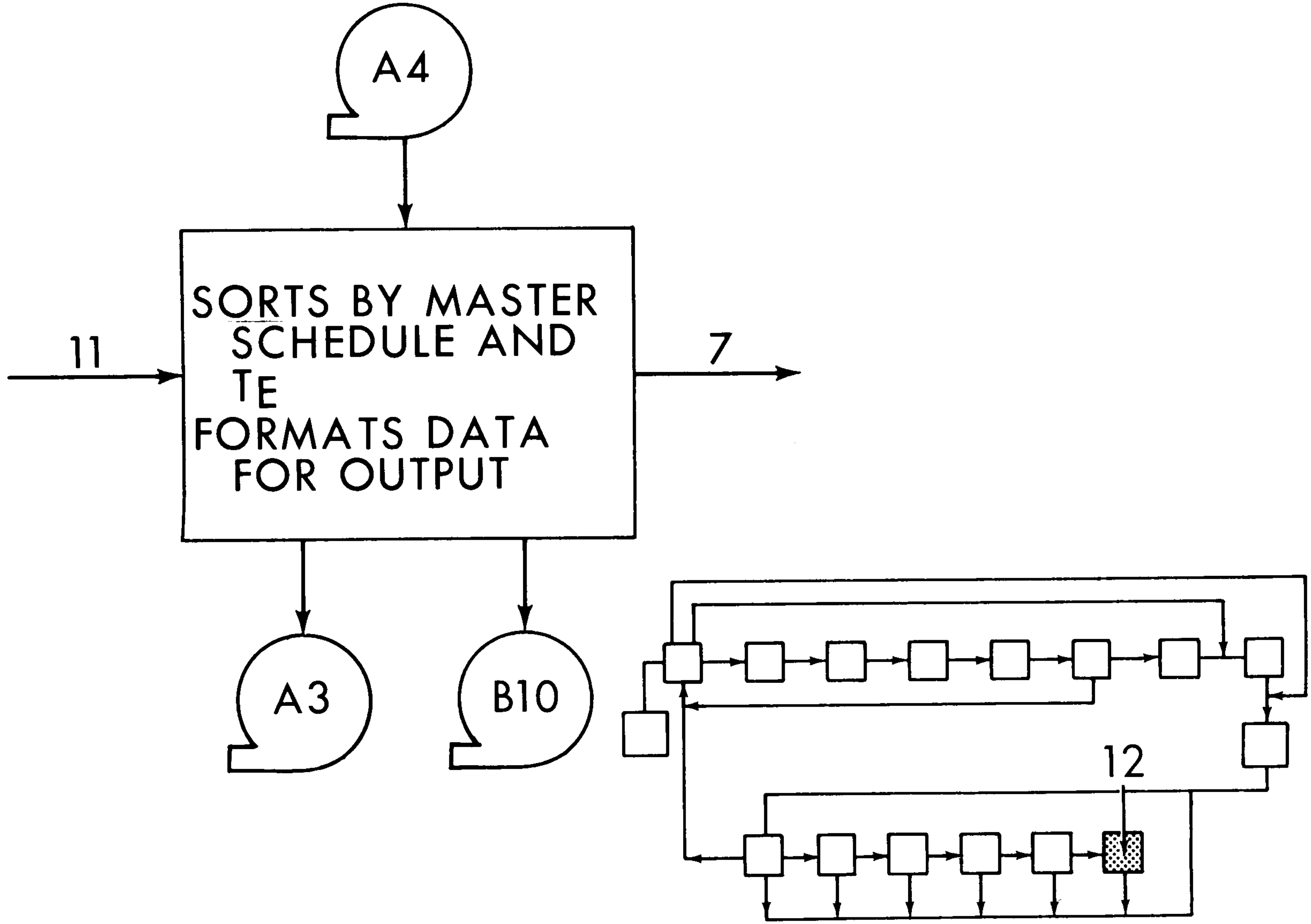


Figure 17

NASA-S-64-4770

PERT "C"

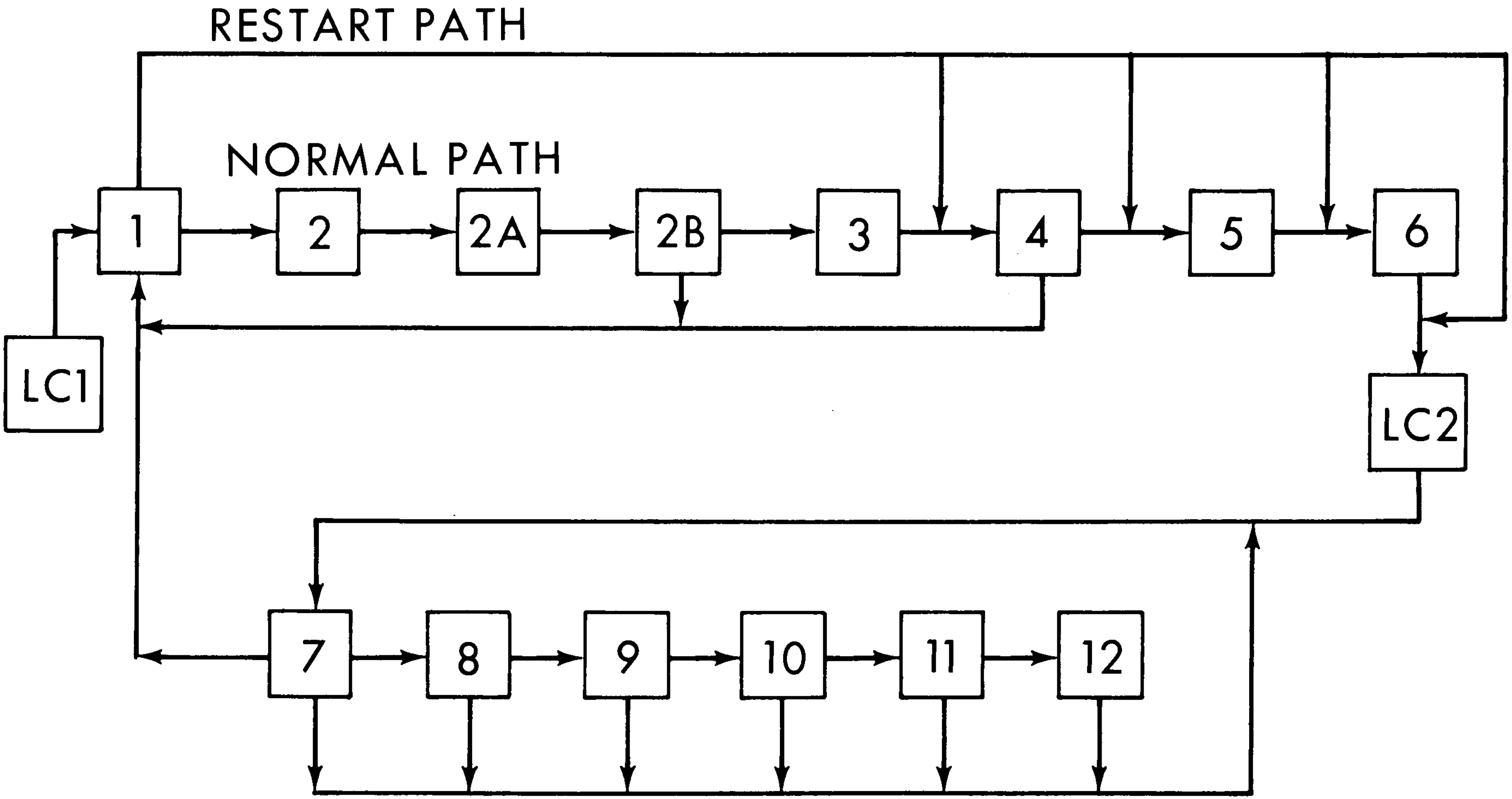


Figure 18

NASA-S-64-4745

SAMPLE NETWORK

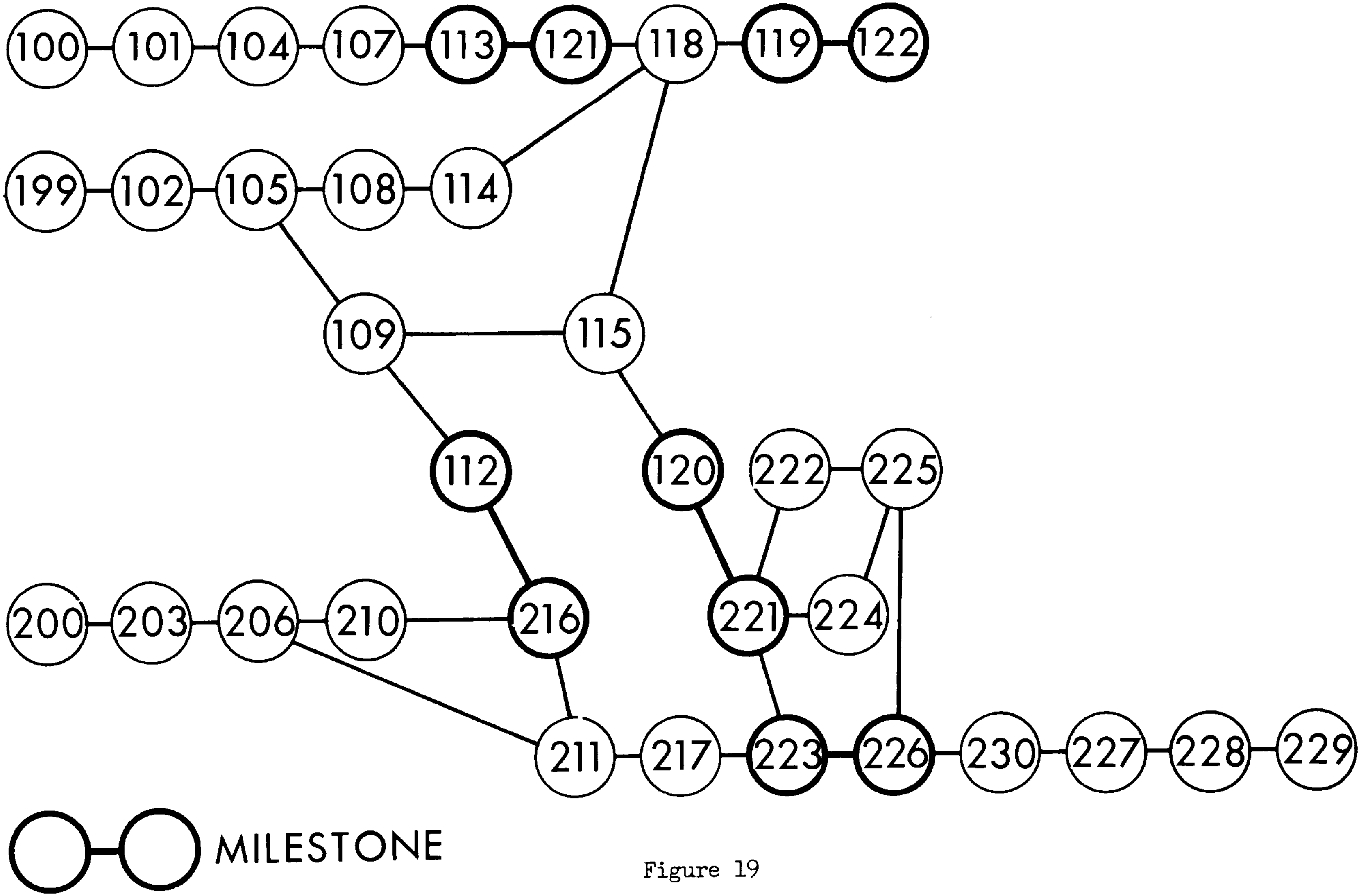


Figure 19

```
                                                    NASA PERT
                                             MANNED SPACECRAFT CENTER                              PAGE      1

  RUN    1                                                                      DATE OF THIS REPORT IS   01-28-64
  BY SUCCESSOR EVENT NUMBER AND PREDECESSOR EVENT NUMBER
  NETWORK  NP        PERT SAMPLE NETWORK
      EVENT                                                ACTIV.        DATE             DATE                 RE-  TIME
   PRE.        SUC.          ACTIVITY DESCRIPTION          TIME    EXPECTED  ALLOWED   SCHD/ACT.  SLACK   SOURCE REM. DEPT

 0100-000  0100-001  PREL. STRUCTURE DESIGN                                  02-16-64  A01-15-64   4.5              DESG
 0100-099  0100-002  PROJECT APPROVAL                                        01-02-64  A01-02-64    .0              HDQ
 0100-001  0100-004  INITIATE DESIGN CRITERION              2.2   01-30-64   03-02-64              4.5          .3  DESG
 0100-002  0100-005  SYSTEM SPECS DEFINED                                    01-28-64  A01-28-64    .0              ELEC
 0100-004  0100-007  COMP STRUCTURE MOCK-UP                 3.0   02-20-64   03-23-64              4.5         3.3  STRC
 0100-005  0100-008  COMP SYSTEMS BREADBOARD                4.0   02-25-64   03-16-64              2.9         4.0  ELEC
 0100-005  0100-009  PREL SYS CIRCUIT DESIGN                6.0   03-10-64   03-10-64               .0         6.0  DESG
 0100-009  0100-012  TRANSMIT CIRCUIT INFO. TO ELEC. DEPT.  2.0   03-24-64   03-24-64               .0         8.0  DESG
 0100-007  0100-013  FINAL STRUCTURE DESIGN                 4.0   03-19-64   04-20-64              4.5         7.3  DESG
 0100-008  0100-014  COMP SYSTEMS DESIGN                    2.0   03-10-64   03-30-64              2.9         6.0  DESG
 0100-009  0100-015  COMP FINAL SYS CIRCUIT DESIGN          4.0   04-07-64   04-07-64               .0        10.0  DESG
 0100-014  0100-018  PREPARE FOR HDQS REVIEW                4.0   04-07-64   04-27-64              2.9        10.0  DESG
 0100-015  0100-018   PREPARE FOR HDQS REVIEW               2.0   04-21-64   04-27-64               .9        12.0  ELEC
 0100-021  0100-018  PREPORE FOR HDQ REVIEW                 1.0   03-26-64   04-27-64              4.5         8.3  DESG
 0100-018  0100-019  HDQ REVIEW                             8.0   06-16-64   06-22-64               .9        20.0  HDQ
 0100-015  0100-020  RECEIVE FINAL CIRCUIT DESIGN           1.0   04-14-64   04-14-64               .0        11.0  DESG
 0100-013  0100-021      CENTER REVIEW                       .0   03-19-64   04-20-64   04-20-64   4.5         7.3
 0100-019  0100-022      HDQ REVIEW COMPLETE                 .0   06-16-64   06-22-64   06-22-64    .9        20.0
```

Figure 20

```
                                   NASA PERT
                            MANNED SPACECRAFT CENTER                              PAGE     1

 RUN    1                                                  DATE ØF THIS REPØRT IS   01-28-64
 BY PATHS ØF CRITICALITY
 NETWØRK  NP      PERT SAMPLE NETWØRK
     EVENT                                      ACTIV.        DATE              DATE                RE-   TIME
 PRE.      SUC.        ACTIVITY DESCRIPTIØN      TIME    EXPECTED  ALLØWED   SCHD/ACT.  SLACK   SØURCE REM. DEPT

0100-099  0100-002  PRØJECT APPRØVAL                              01-02-64  A01-02-64    .0               HDQ
0100-002  0100-005  SYSTEM SPECS DEFINED                          01-28-64  A01-28-64    .0               ELEC
0100-005  0100-009  PREL SYS CIRCUIT DESIGN         6.0  03-10-64  03-10-64              .0           6.0 DESG
0100-009  0100-012  TRANSMIT CIRCUIT INFØ. TØ ELEC. DEPT.  2.0  03-24-64  03-24-64       .0           8.0 DESG
0100-009  0100-015  CØMP FINAL SYS CIRCUIT DESIGN   4.0  04-07-64  04-07-64              .0          10.0 DESG
0100-015  0100-020  RECEIVE FINAL CIRCUIT DESIGN    1.0  04-14-64  04-14-64              .0          11.0 DESG

0100-015  0100-018   PREPARE FØR HDQS REVIEW        2.0  04-21-64  04-27-64              .9          12.0 ELEC
0100-018  0100-019  HDQ REVIEW                      8.0  06-16-64  06-22-64              .9          20.0 HDQ
0100-019  0100-022      HDQ REVIEW CØMPLETE          .0  06-16-64  06-22-64  06-22-64    .9          20.0

0100-005  0100-008  CØMP SYSTEMS BREADBØARD         4.0  02-25-64  03-16-64             2.9           4.0 ELEC
0100-008  0100-014  CØMP SYSTEMS DESIGN             2.0  03-10-64  03-30-64             2.9           6.0 DESG
0100-014  0100-018  PREPARE FØR HDQS REVIEW         4.0  04-07-64  04-27-64             2.9          10.0 DESG

0100-000  0100-001  PREL. STRUCTURE DESIGN                        02-16-64  A01-15-64   4.5               DESG
0100-001  0100-004  INITIATE DESIGN CRITERIØN       2.2  01-30-64  03-02-64             4.5            .3 DESG
0100-004  0100-007  CØMP STRUCTURE MØCK-UP          3.0  02-20-64  03-23-64             4.5           3.3 STRC
0100-007  0100-013  FINAL STRUCTURE DESIGN          4.0  03-19-64  04-20-64             4.5           7.3 DESG
0100-013  0100-021      CENTER REVIEW                .0  03-19-64  04-20-64  04-20-64   4.5           7.3
0100-021  0100-018  PREPØRE FØR HDQ REVIEW          1.0  03-26-64  04-27-64             4.5           8.3 DESG
```

Figure 21

NASA-S-64-4748

CRITICAL PATH CUT-OFF

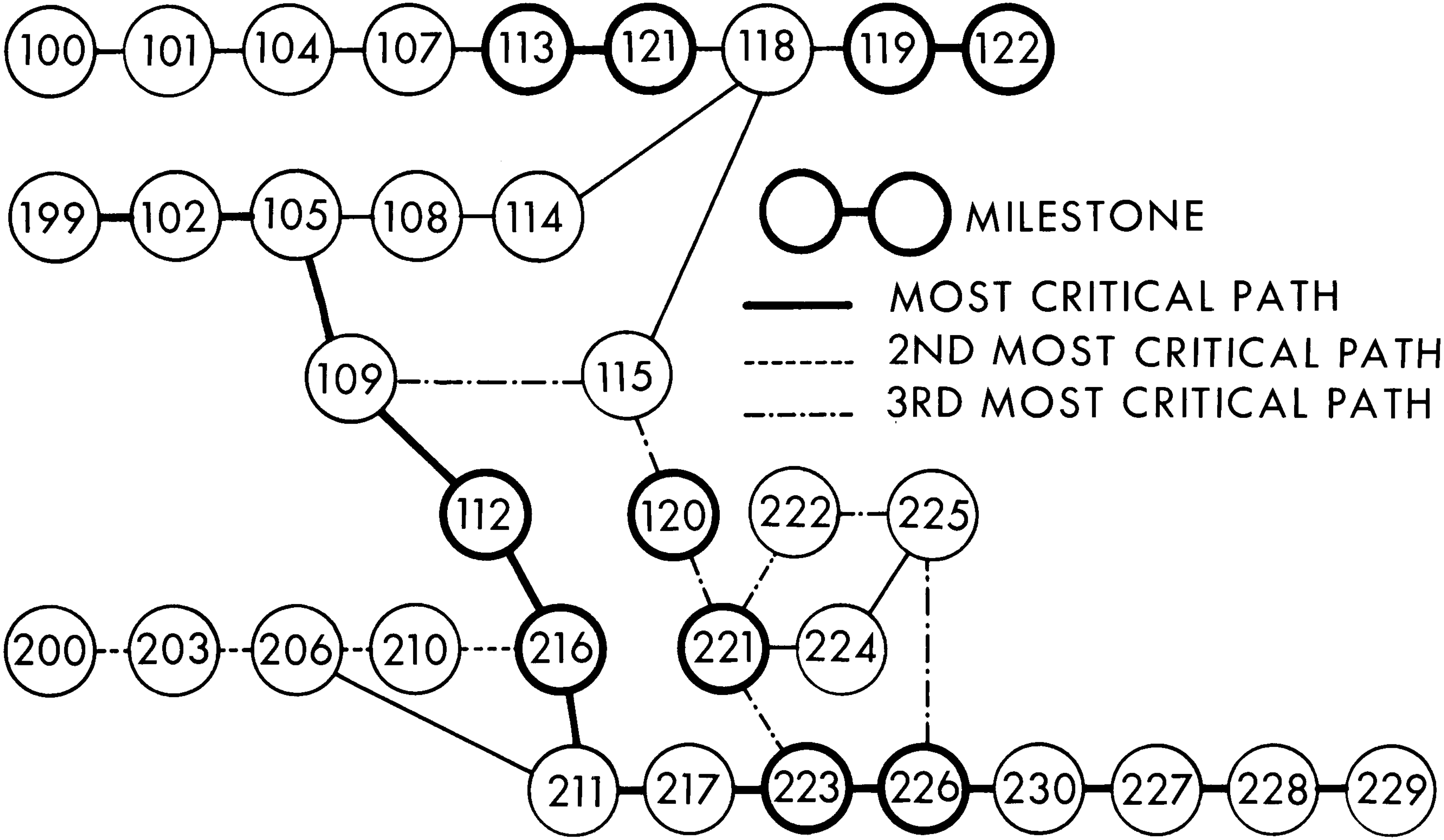


Figure 22

NASA-S-64-4746

EXPECTED DATE CUT-OFF

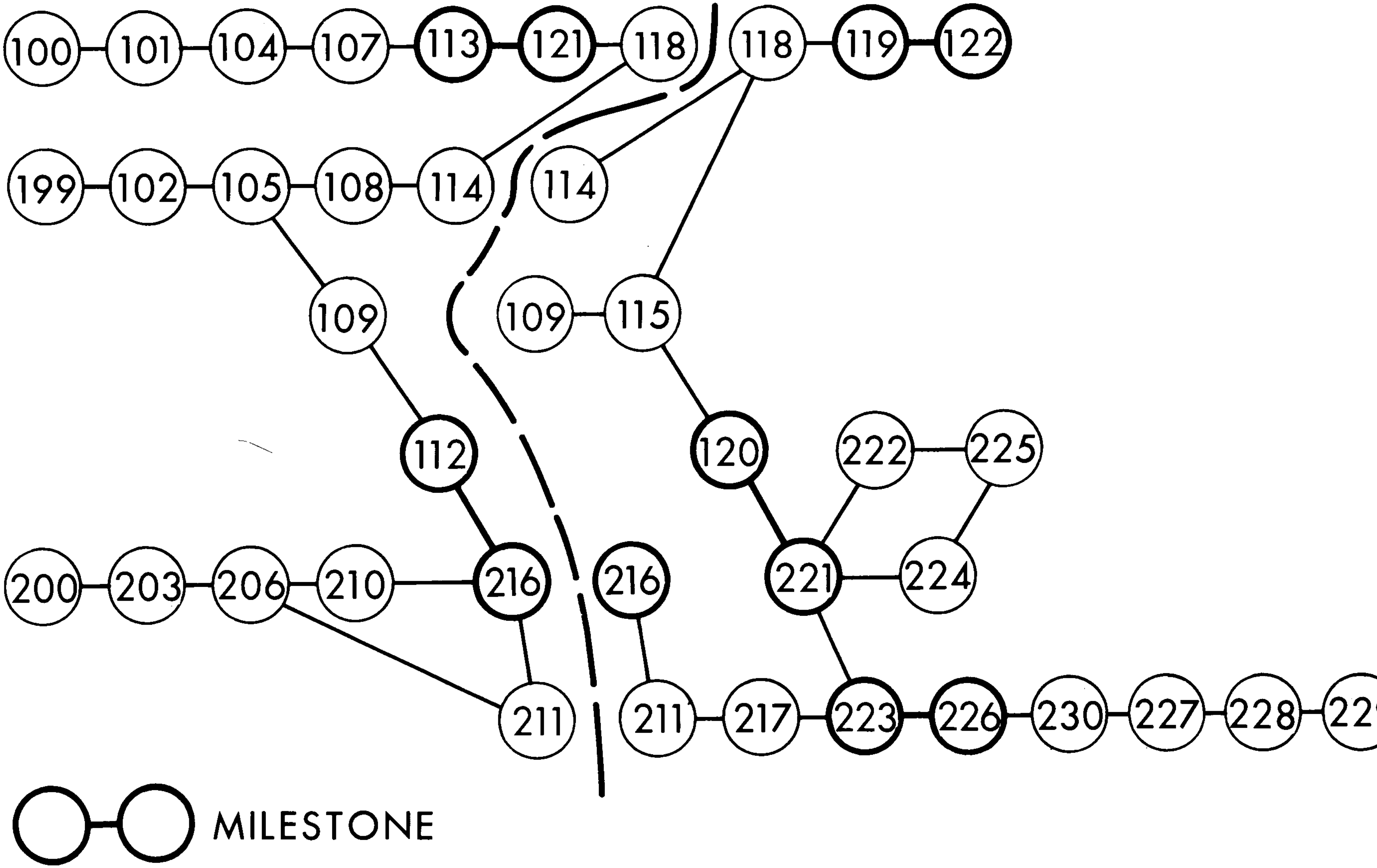


Figure 24

```
                                                  NASA PERT
                                         MANNED SPACECRAFT CENTER                                     PAGE       2
---------------------------------------------------------------------------------------------------------------------
 RUN    1                                                                 DATE ØF THIS REPØRT IS    01-28-64
 BY DEPARTMENT, EXPECTED DATF, AND SUCCESSØR EVENT
 NETWØRK  NP        PERT SAMPLE NETWØRK
     EVENT                                                    ACTIV.        DATE               DATE              RE-  TIME
 PRE.        SUC.          ACTIVITY DESCRIPTIØN               TIME    EXPECTED  ALLØWED    SCHD/ACT.  SLACK   SØURCE REM. DEPT
---------------------------------------------------------------------------------------------------------------------
0100-002  0100-005  SYSTEM SPECS DEFINED                                        01-28-64   A01-28-64     .0              ELEC
0100-005  0100-008  CØMP SYSTEMS BREADBOARD                    4.0    02-25-64  03-16-64                2.9          4.0 ELEC
0100-015  0100-018   PREPARE FØR HDQS REVIEW                   2.0    04-21-64  04-27-64                 .9         12.0 ELEC
---------------------------------------------------------------------------------------------------------------------
```

Figure 28

NASA-S-64-4706

NASA PERT
MANNED SPACECRAFT CENTER
MASTER SCHEDULE 02

RUN 1 — DATE OF THIS REPORT IS 01-28-64

NETWORK NP PERT SAMPLE NETWORK

EVENT	EVENT	DATE		DATE	
NO.	DESCRIPTION	EXPECTED	ALLOWED	SCH/ACT	SLACK
0200-030	HDQ REVIEW COMPLETE	07-14-64	05-07-64	05-07-64	- 9.7
0200-029	COMP EVAL OF PROPOSAL	10-27-64	08-20-64	08-20-64	- 9.7

Figure 29

NASA-S-64-4747

LATEST ALLOWABLE DATE CUT-OFF

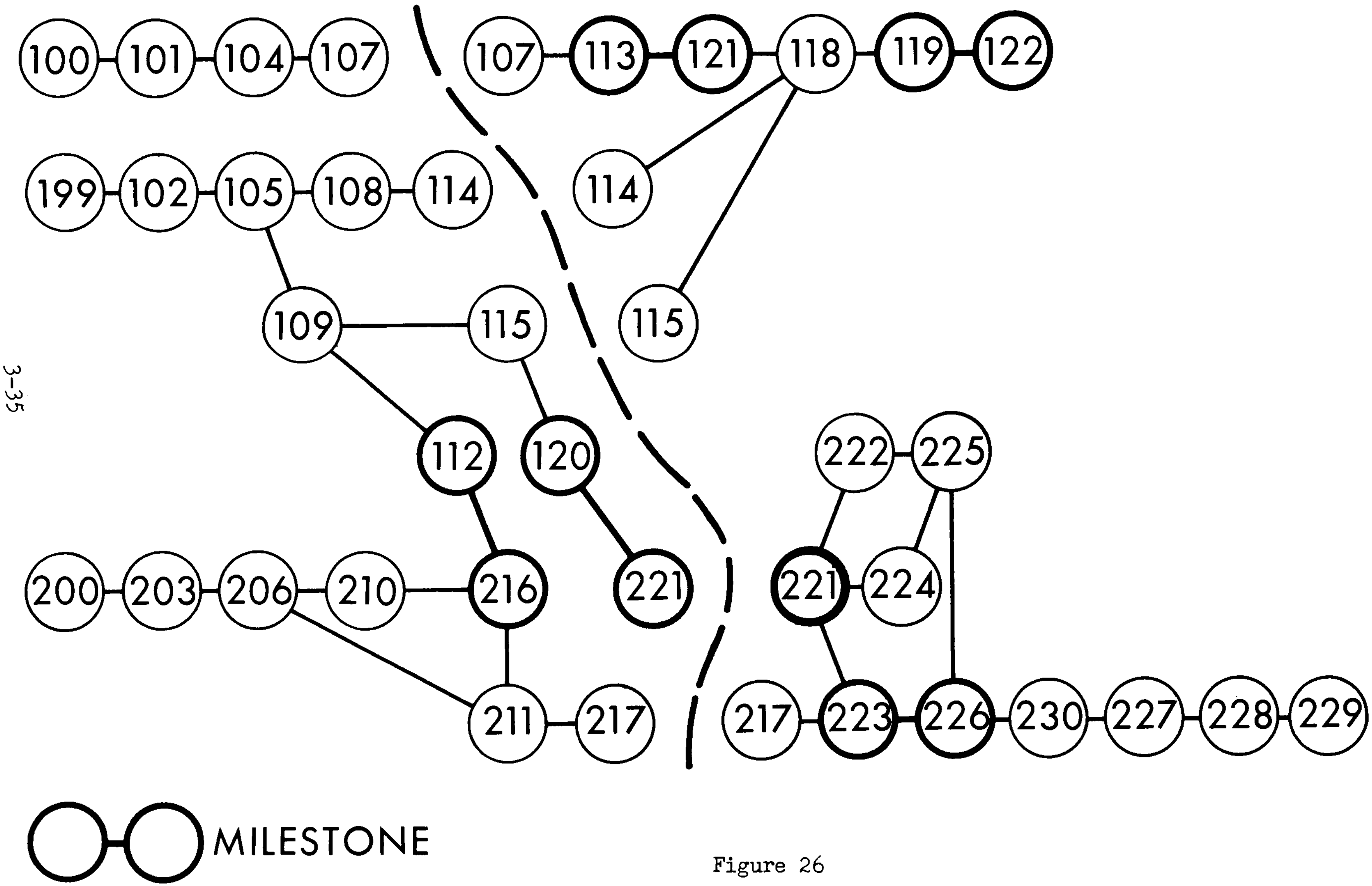


Figure 26

NASA PERT
MANNED SPACECRAFT CENTER

PAGE 1

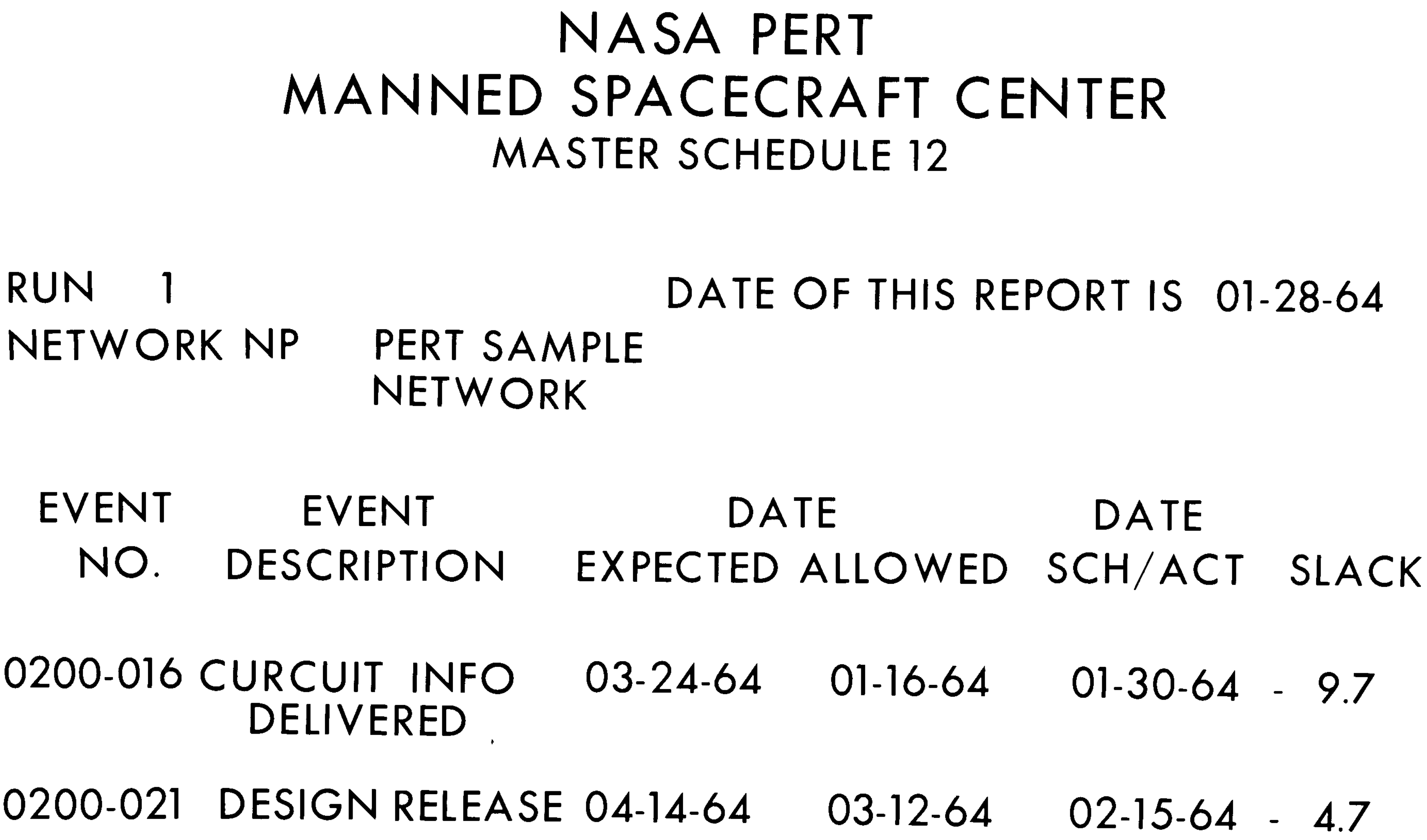
RUN 1
BY DEPARTMENT, EXPECTED DATE, AND SUCCESSØR EVENT
NETWØRK NP PERT SAMPLE NETWØRK

DATE ØF THIS REPØRT IS 01-28-64

EVENT			ACTIV.	DATE		DATE		RE-	TIME	
PRE.	SUC.	ACTIVITY DESCRIPTIØN	TIME	EXPECTED	ALLØWED	SCHD/ACT.	SLACK	SØURCE	REM.	DEPT
0100-000	0100-001	PREL. STRUCTURE DESIGN			02-16-64	A01-15-64	4.5			DESG
0100-001	0100-004	INITIATE DESIGN CRITERIØN	2.2	01-30-64	03-02-64		4.5		.3	DESG
0100-005	0100-009	PREL SYS CIRCUIT DESIGN	6.0	03-10-64	03-10-64		.0		6.0	DESG
0100-008	0100-014	CØMP SYSTEMS DESIGN	2.0	03-10-64	03-30-64		2.9		6.0	DESG
0100-007	0100-013	FINAL STRUCTURE DESIGN	4.0	03-19-64	04-20-64		4.5		7.3	DESG
0100-009	0100-012	TRANSMIT CIRCUIT INFØ. TØ ELEC. DEPT.	2.0	03-24-64	03-24-64		.0		8.0	DESG
0100-021	0100-018	PREPØRE FØR HDQ REVIEW	1.0	03-26-64	04-27-64		4.5		8.3	DESG
0100-009	0100-015	CØMP FINAL SYS CIRCUIT DESIGN	4.0	04-07-64	04-07-64		.0		10.0	DESG
0100-014	0100-018	PREPARE FØR HDQS REVIEW	4.0	04-07-64	04-27-64		2.9		10.0	DESG
0100-015	0100-020	RECEIVE FINAL CIRCUIT DESIGN	1.0	04-14-64	04-14-64		.0		11.0	DESG

Figure 27

NASA-S-64-4704



NASA PERT
MANNED SPACECRAFT CENTER
MASTER SCHEDULE 12

RUN 1 DATE OF THIS REPORT IS 01-28-64

NETWORK NP PERT SAMPLE NETWORK

EVENT NO.	EVENT DESCRIPTION	DATE EXPECTED	DATE ALLOWED	DATE SCH/ACT	SLACK
0200-016	CURCUIT INFO DELIVERED	03-24-64	01-16-64	01-30-64	- 9.7
0200-021	DESIGN RELEASE	04-14-64	03-12-64	02-15-64	- 4.7

Figure 30

NASA-S-64-4705

NASA PERT
MANNED SPACECRAFT CENTER
MASTER SCHEDULE *2

RUN 1
NETWORK NP PERT SAMPLE NETWORK

DATE OF THIS REPORT IS 01-28-64

EVENT	EVENT	DATE		DATE	
NO.	DESCRIPTION	EXPECTED	ALLOWED	SCH/ACT	SLACK
0200-016	CIRCUIT INFO DELIVERED	03-24-64	01-16-64	01-30-64	- 9.7
0200-021	DESIGN RELEASE	04-14-64	03-12-64	02-15 64	- 4.7
0200-030	HDQ REVIEW COMPLETE	07-14-64	05-07-64	05-07-64	- 9.7
0200-029	COMP EVAL OF PROPOSAL	10-27-64	08-20-64	08-20-64	- 9.7

Figure 31

NASA-S-64-4744

LIST OF PAST DUE ACTIVITIES

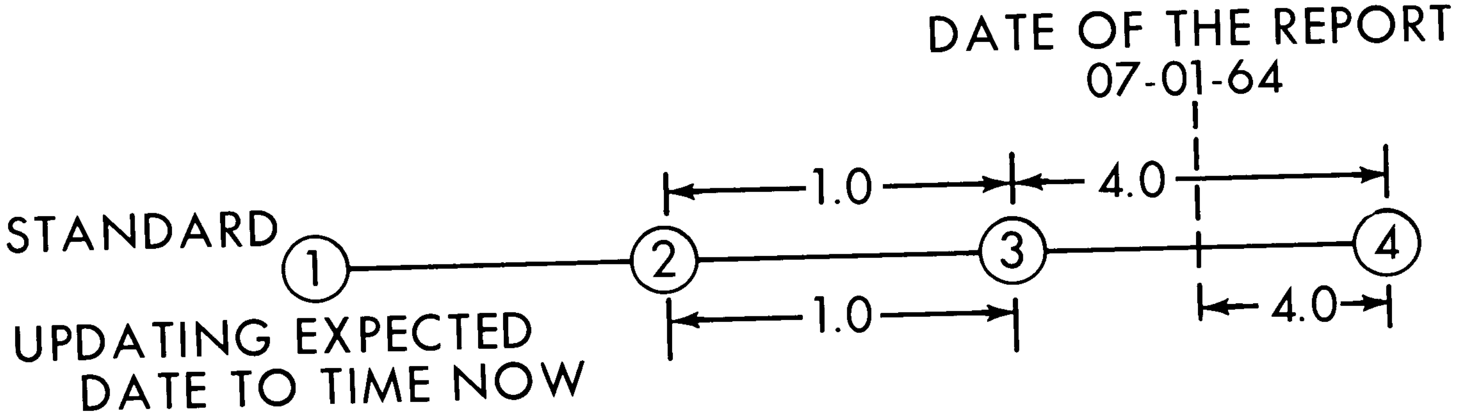
DATE OF THIS REPORT 03-01-64

PREDECESSOR	SUCCESSOR	ACTIVITY DESCRIPTION	ACTIVITY TIME	EXPECTED DATE
0100-001	0100-004	INITIATE DESIGN CRITERION	2.2	01-30-64
0100-005	0100-008	COMP SYSTEMS BREADBOARD	4.0	02-25-64
0200-003	0200-006	FEASIBILITY STUDY	3.0	01-30-64

Figure 31A

NASA-S-64-4693

UPDATING EXPECTED DATE



STANDARD PERT

PREDECESSOR	SUCCESSOR	TIME EST	EXPECTED DATE	ACTUAL DATE
1	2			06-15-64
2	3	1.0	06-22-64	
3	4	4.0	07-20-64	

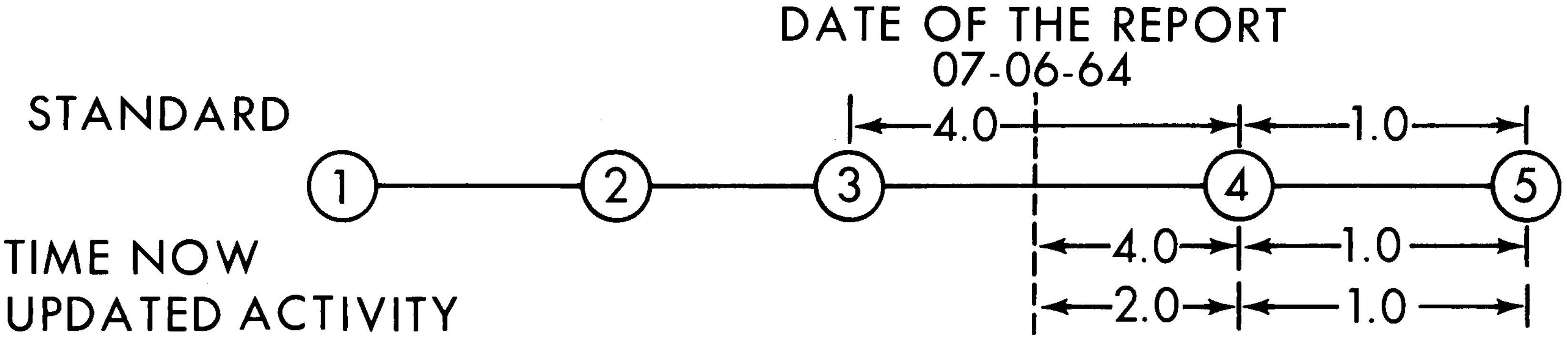
UPDATING EXPECTED DATE

PREDECESSOR	SUCCESSOR	TIME EST	EXPECTED DATE	ACTUAL DATE
1	2			06-15-64
2	3	1.0	07-01-64	
3	4	4.0	07-29-64	

Figure 32

NASA-S-64-4692

TIME NOW PERT



STANDARD PERT				
PREDECESSOR	SUCCESSOR	TIME EST	EXPECTED DATE	ACTUAL DATE
1	2			06-15-64
2	3			06-22-64
3	4	4.0	07-20-64	
4	5	1.0	07-27-64	

TIME NOW PERT				
1	2			06-15-64
2	3			06-22-64
3	4	4.0	08-03-64	
4	5	1.0	08-10-64	
3	4	2.0	07-20-64	
4	5	1.0	07-27-64	

Figure 33

NASA-S-64-4798

CODE 4 MERGED

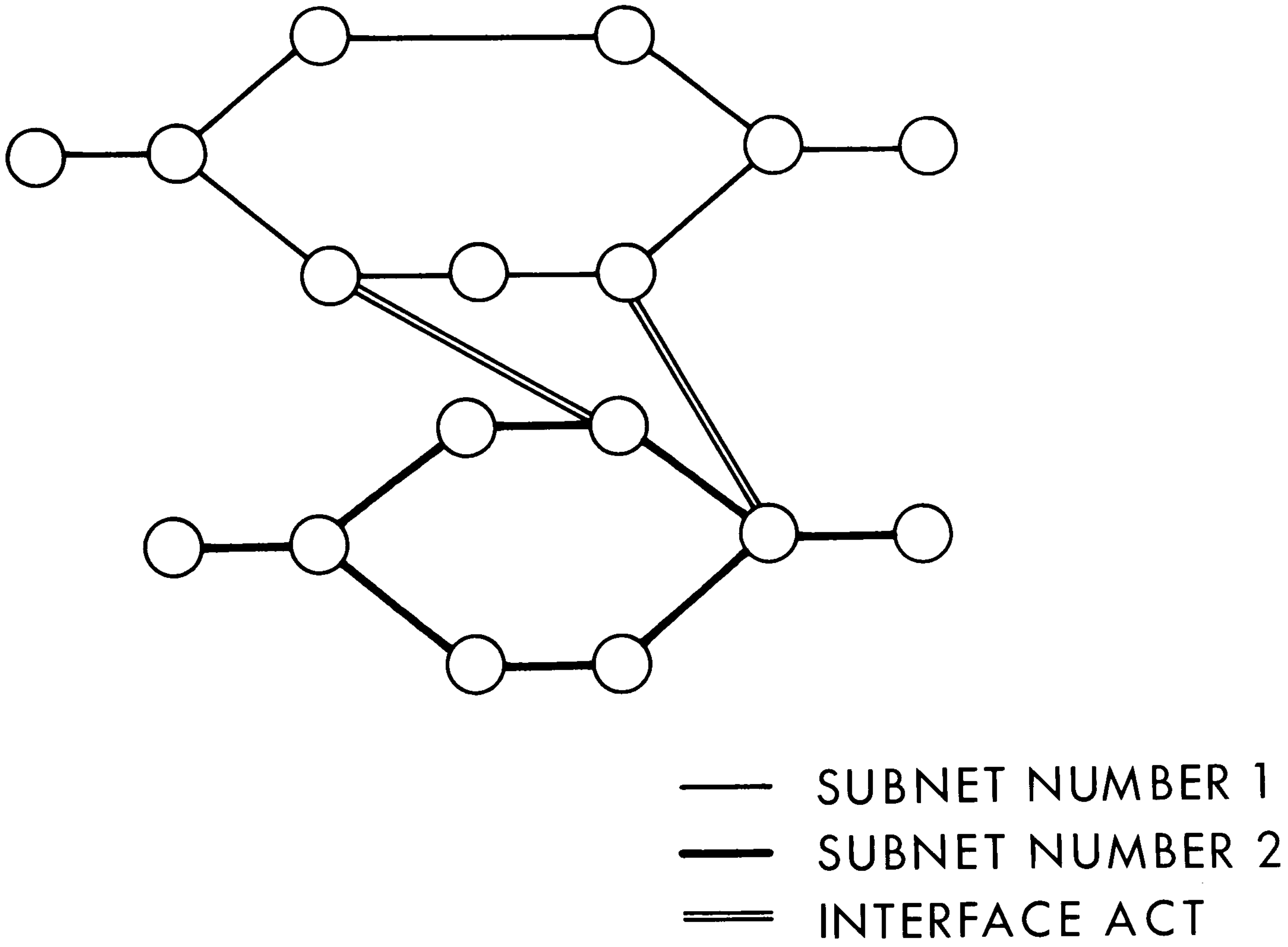


Figure 34

NASA-S-64-4799

CODE 4 DEMERGED

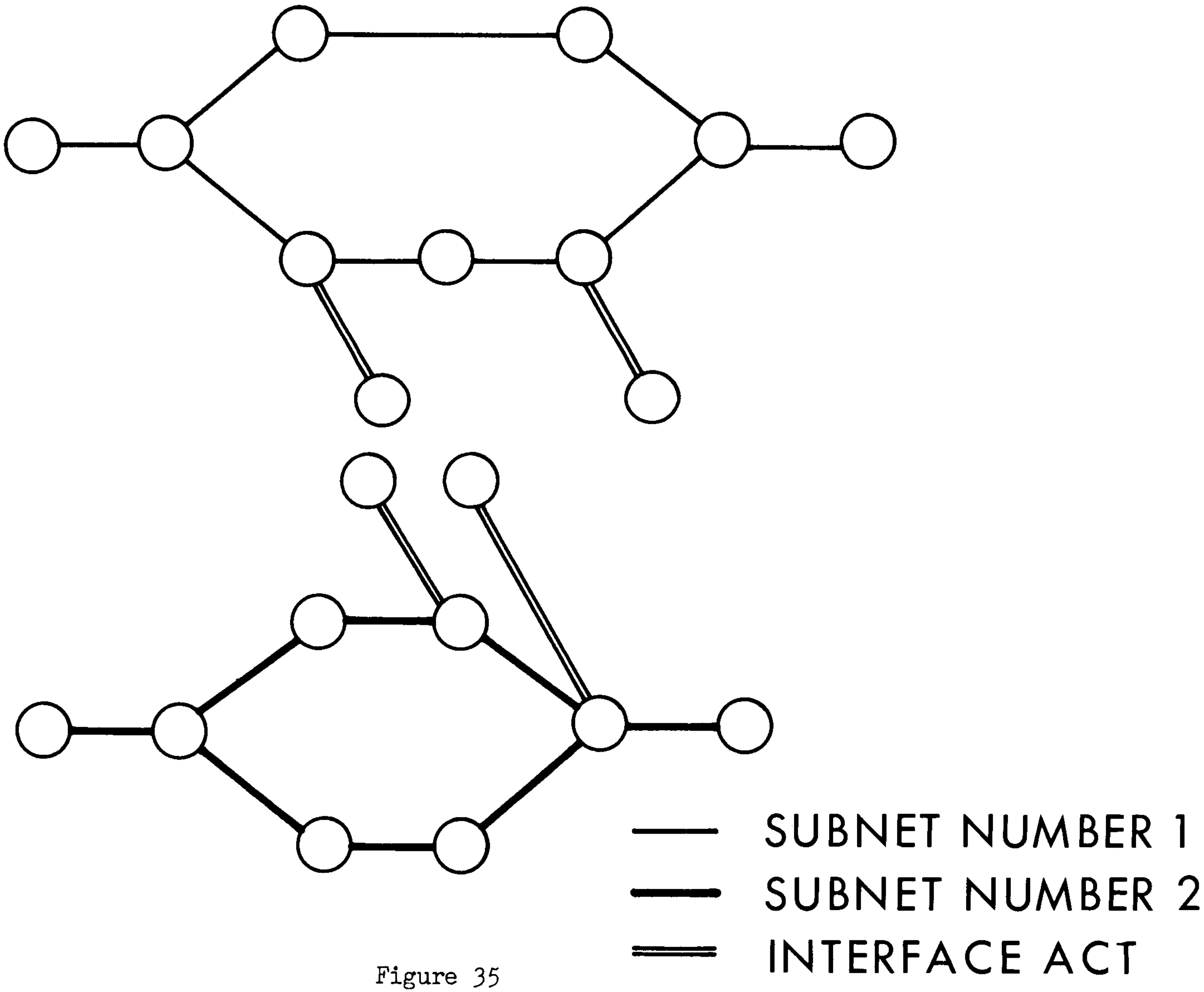


Figure 35

NASA-S-64-4689

7094
DELETION OF COMPLETED ACTIVITIES

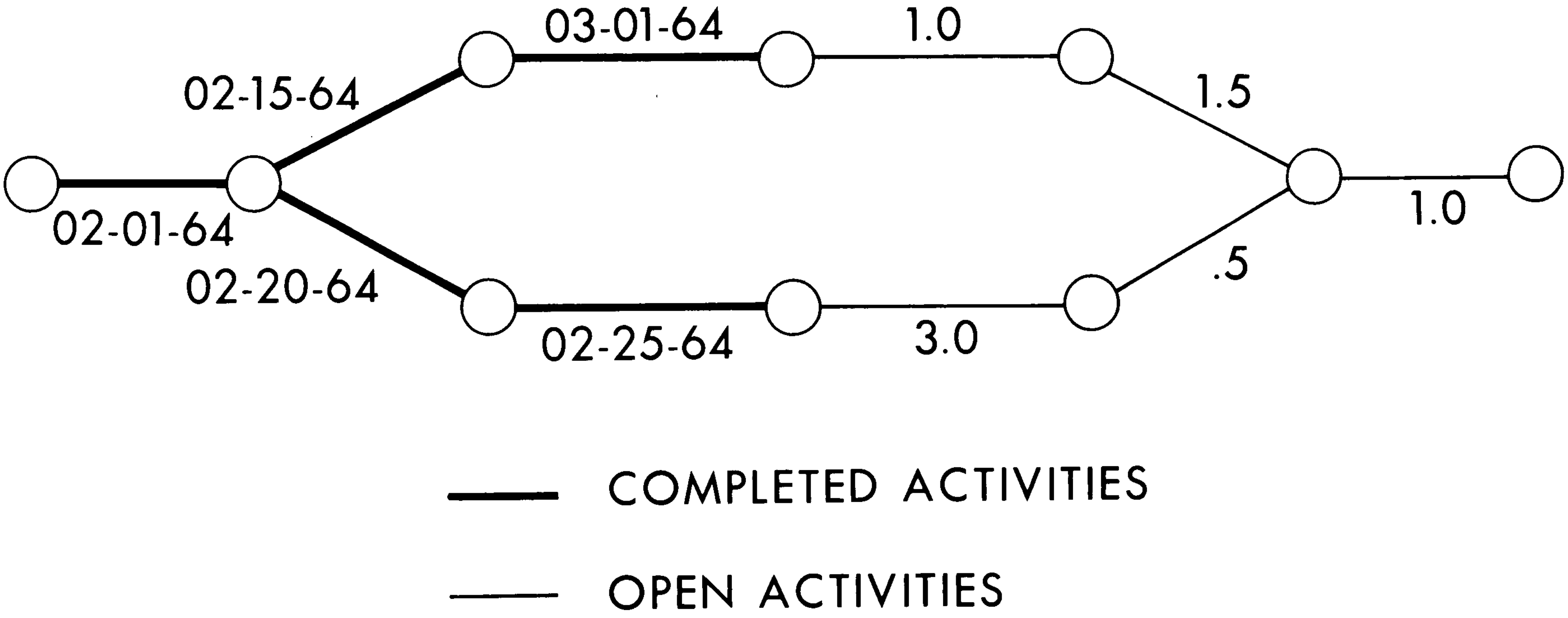


Figure 36

NASA-S-64-4690

7094 DELETION OF COMPLETED ACTIVITIES (CONT)

STANDARD PERT				
PREDECESSOR	SUCCESSOR	TIME EST	EXPECTED DATE	SCH/ACTUAL DATE
1	2			A02-01-64
2	3			A02-15-64
3	4			A03-01-64
4	5	1.0	03-08-64	
2	6			A02-20-64
6	7			A02-25-64
7	8	3.0	03-17-64	
5	8	1.5	03-18-64	
8	9	0.5	03-21-64	
9	10	1.0	03-28-64	
DELETED COMPLETED ACTIVITIES				
4	5	1.0	B03-08-64	
7	8	3.0	B03-17-64	
5	9	1.5	03-18-64	
8	9	0.5	03-21-64	
9	10	1.0	03-28-64	

Figure 37

NASA-S-64-4697

TOTAL NETWORK

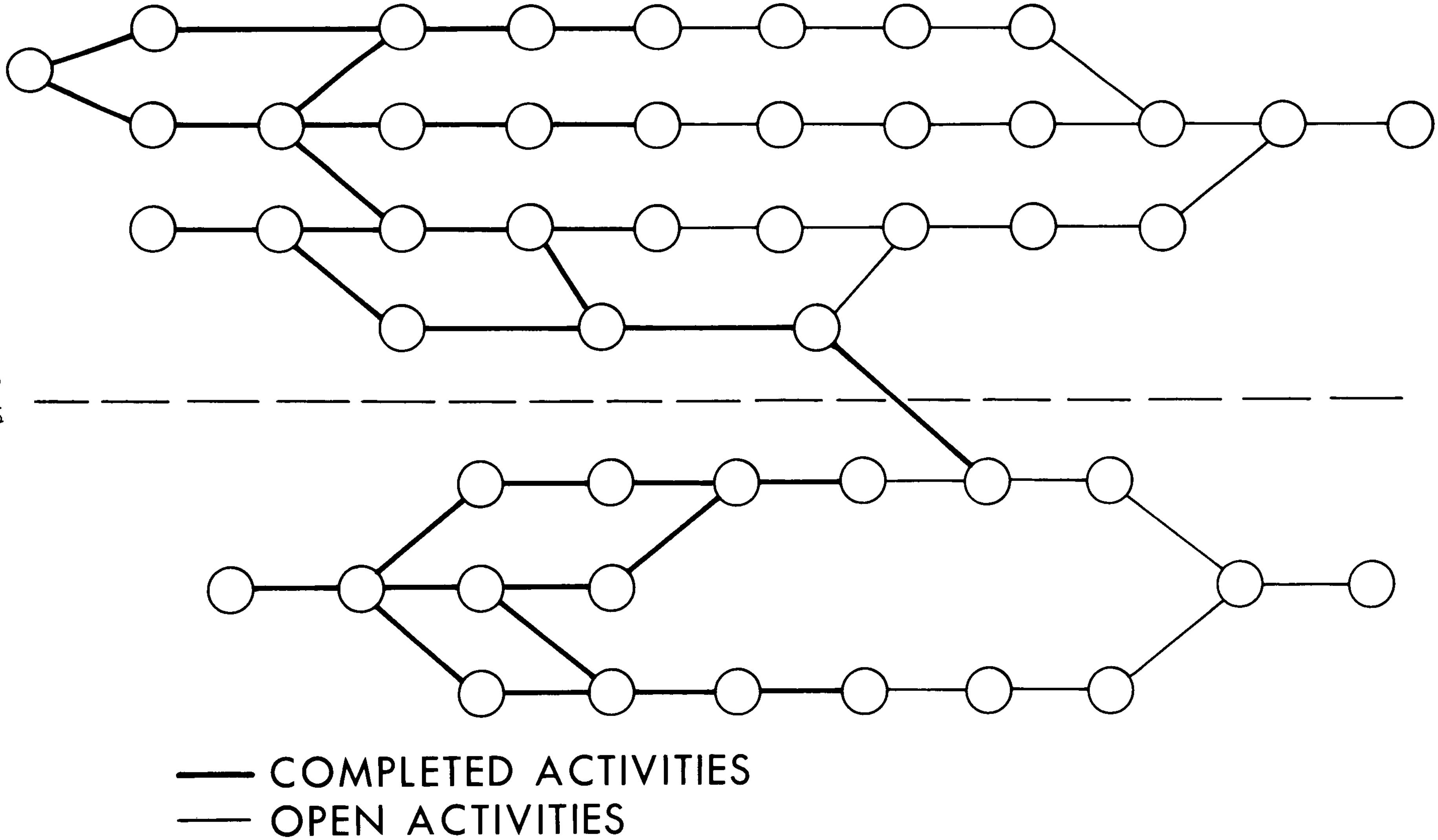


Figure 38

NASA-S-64-4696

ALL COMPLETED ACTIVITIES DELETED

STEP 1

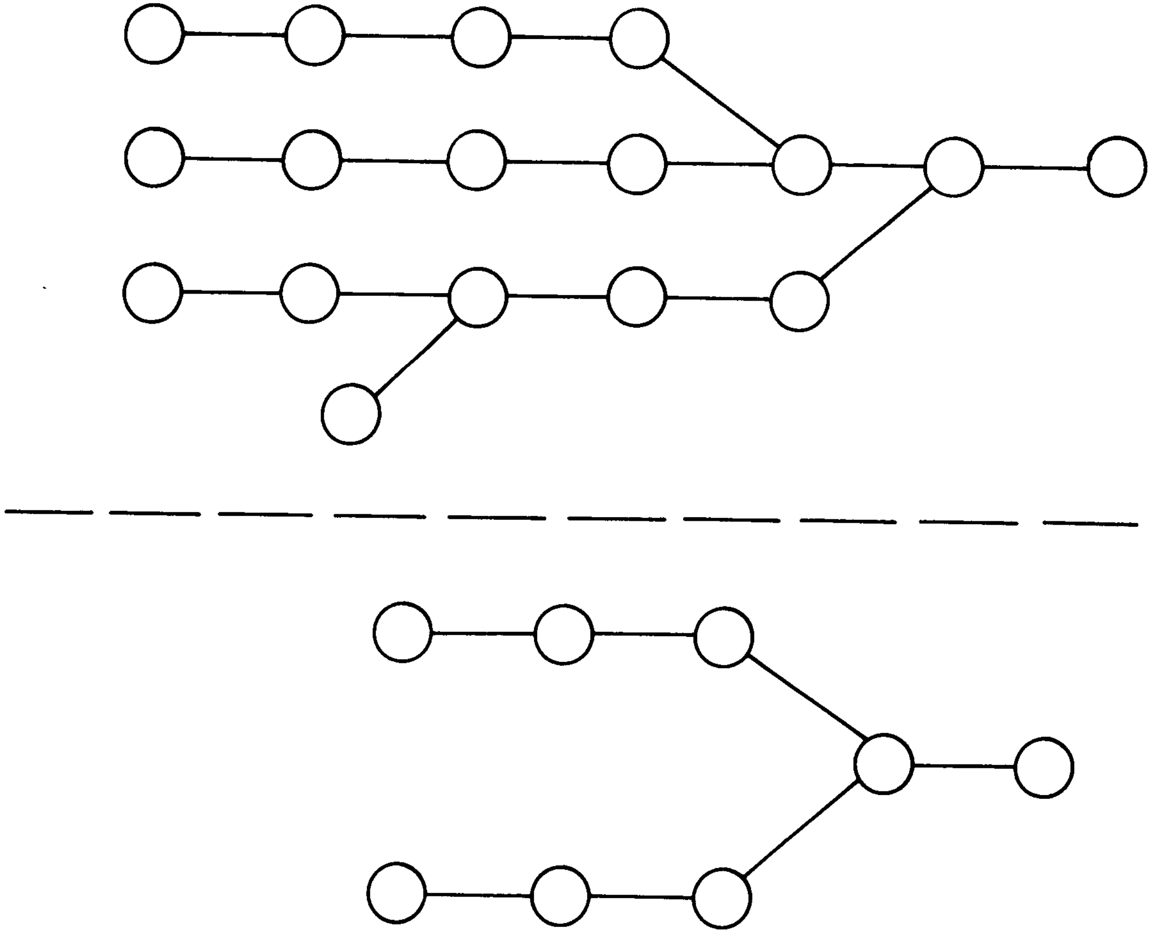
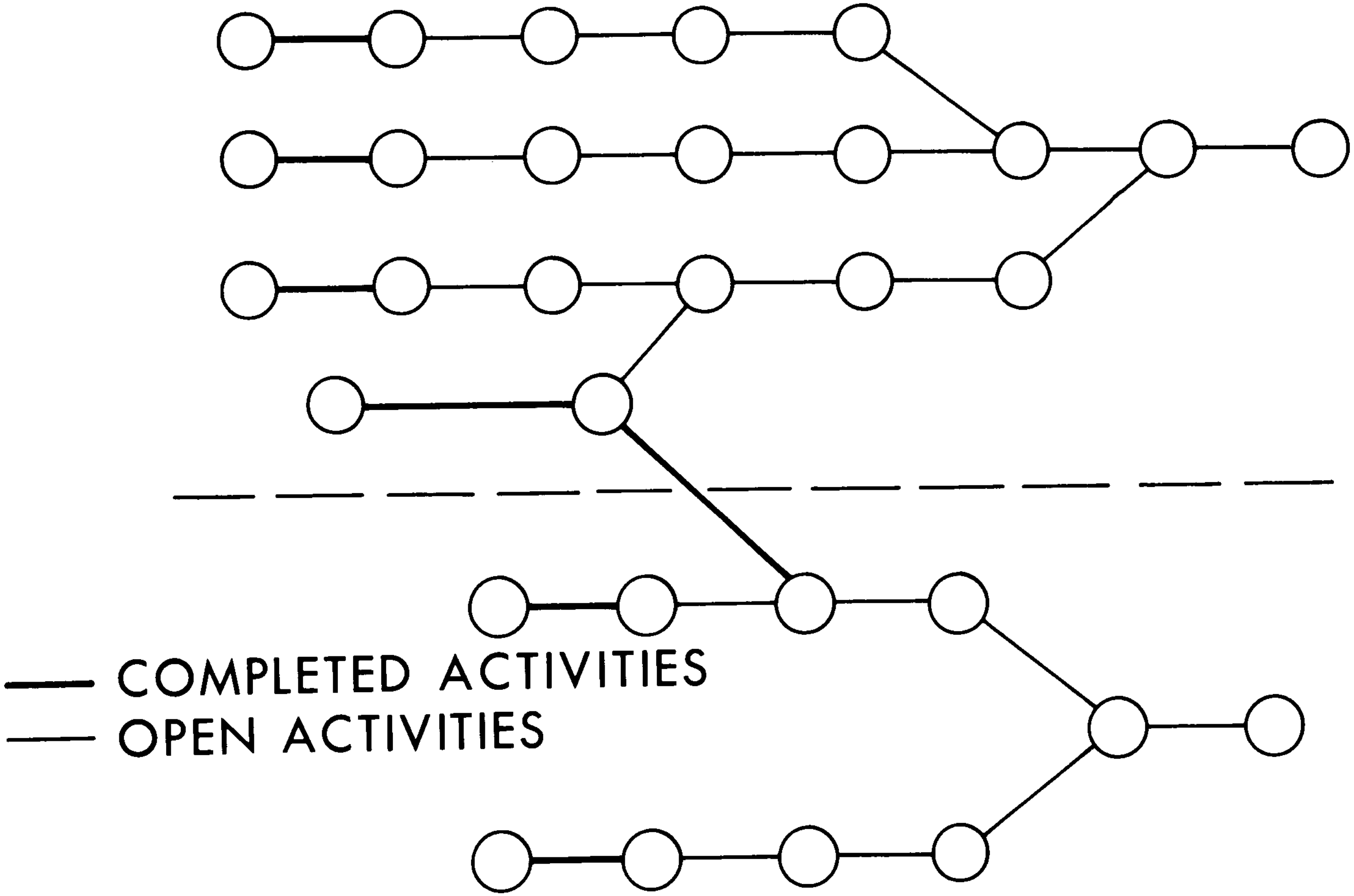


Figure 39

NASA-S-64-4695

ADDITION OF NECESSARY COMPLETED ACTIVITIES
STEP 2



— COMPLETED ACTIVITIES
— OPEN ACTIVITIES

Figure 40

NASA-S-64-4694

DUMMY STARTING ACTIVITIES ADDED

STEP 3

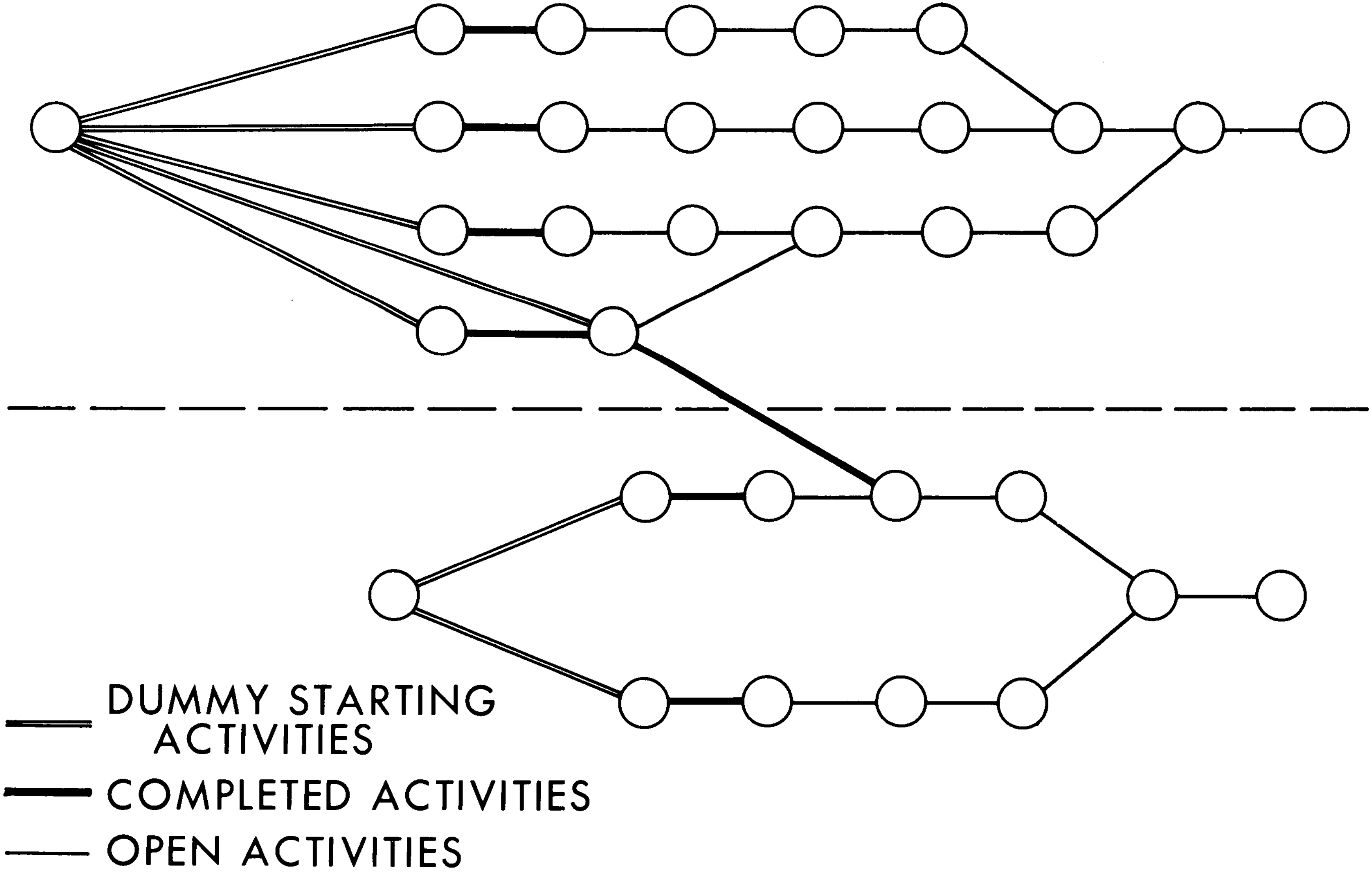


Figure 41

TOPOLOGICAL SORT TECHNIQUE

by

John E. Leonard
NASA Manned Spacecraft Center
Houston, Texas

INTRODUCTION

Over the past few years the size of PERT networks has grown to such proportions that special techniques are needed to efficiently process and analyze them. The purpose of this presentation is to give you a detailed description of a large capacity topological sort routine.

NASA PERT modification "B" was distributed to NASA Computer Centers and to industry in November 1962. In modification B capacity was increased to 7,000 activities and a built-in loop detection technique, developed by the NASA Goddard Space Flight Center, was incorporated into the program.

As MOD B was being distributed, the need for a system capable of handling larger networks was identified. With this in mind, investigations were made of various types of large capacity topological sort routines using the following criteria as a guide:

a. Efficiency on both large and small networks.

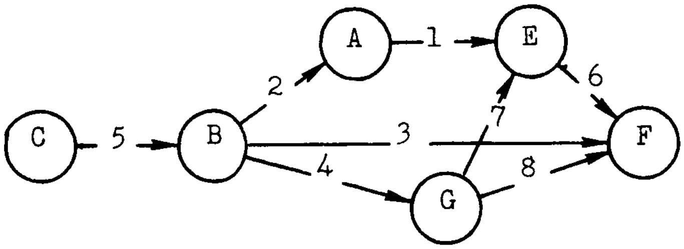
b. Cut down on setup time by using as few tapes as possible.

c. Ability to operate under the Fortran Monitor System.

Based on the above criteria, the decision was made to use a technique developed by Dr. Arthur B. Kahn of Westinghouse, Baltimore, Maryland. Basically, this technique is a ranking method in which the ranking is performed in parallel on all paths of the network. It has one big advantage over most other topological ordering techniques in that all information necessary to order up to 30,000 activities can be stored in core at one time. This eliminates input-output operations during execution of the basic algorithm and is a major factor in the efficiency of the computer routine.

Dr. Kahn's technique, as implemented at the Manned Spacecraft Center, was divided into three phases: Preprocessor, Basic Algorithm and the Postprocessor.

PERT NETWORK



BASIC INPUT LIST

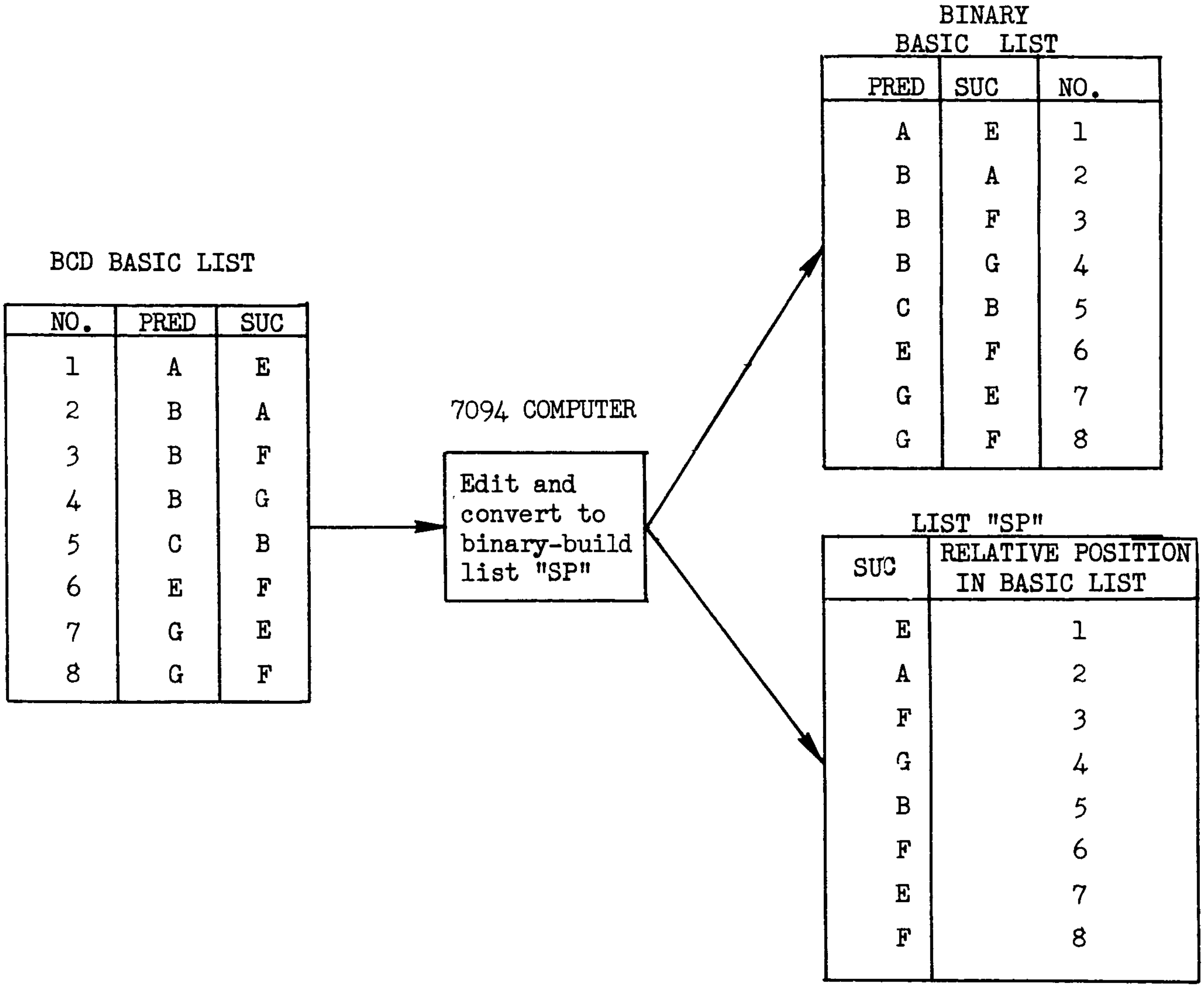
NO.	PRED.	SUC.
1	A	E
2	B	A
3	B	F
4	B	G
5	C	B
6	E	F
7	G	E
8	G	F

Figure 1

THE PREPROCESSOR

The preprocessor was the most difficult phase to program. Four tables had to be developed and packed four entries to a 36 bit word. Figure 1 shows a PERT Network and how this network would be broken down in the NASA PERT system. Notice, in the basic input list, there are eight activities in sort order by predecessor event.

The input data is first edited, converted to binary, and written back on tape. At the same time a two-word list of successor events, and their relative position in the basic list, is written on another tape. (This is shown in Figure 2).



BCD BASIC LIST

NO.	PRED	SUC
1	A	E
2	B	A
3	B	F
4	B	G
5	C	B
6	E	F
7	G	E
8	G	F

7094 COMPUTER

Edit and convert to binary-build list "SP"

BINARY BASIC LIST

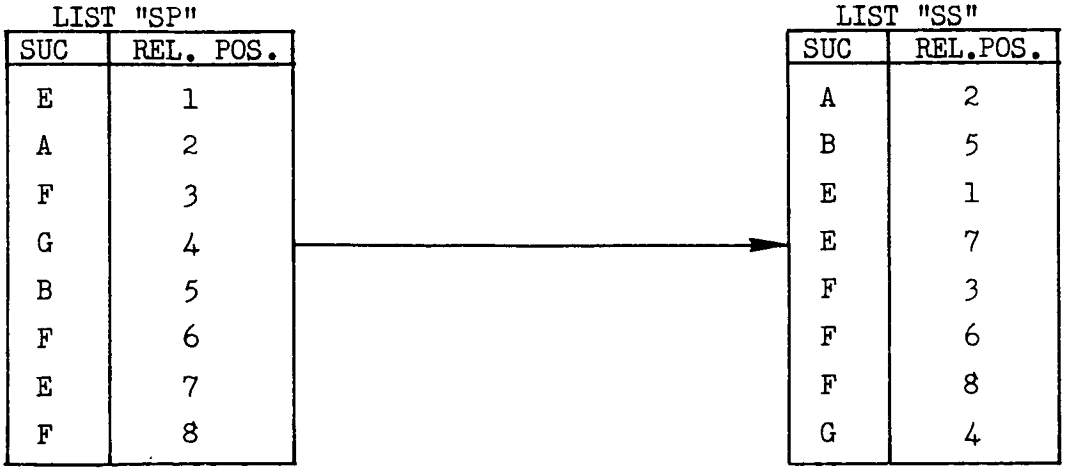
PRED	SUC	NO.
A	E	1
B	A	2
B	F	3
B	G	4
C	B	5
E	F	6
G	E	7
G	F	8

LIST "SP"

SUC	RELATIVE POSITION IN BASIC LIST
E	1
A	2
F	3
G	4
B	5
F	6
E	7
F	8

Figure 2

The table of successor events is then sorted by event number and list "SS" is developed. This is shown in Figure 3.



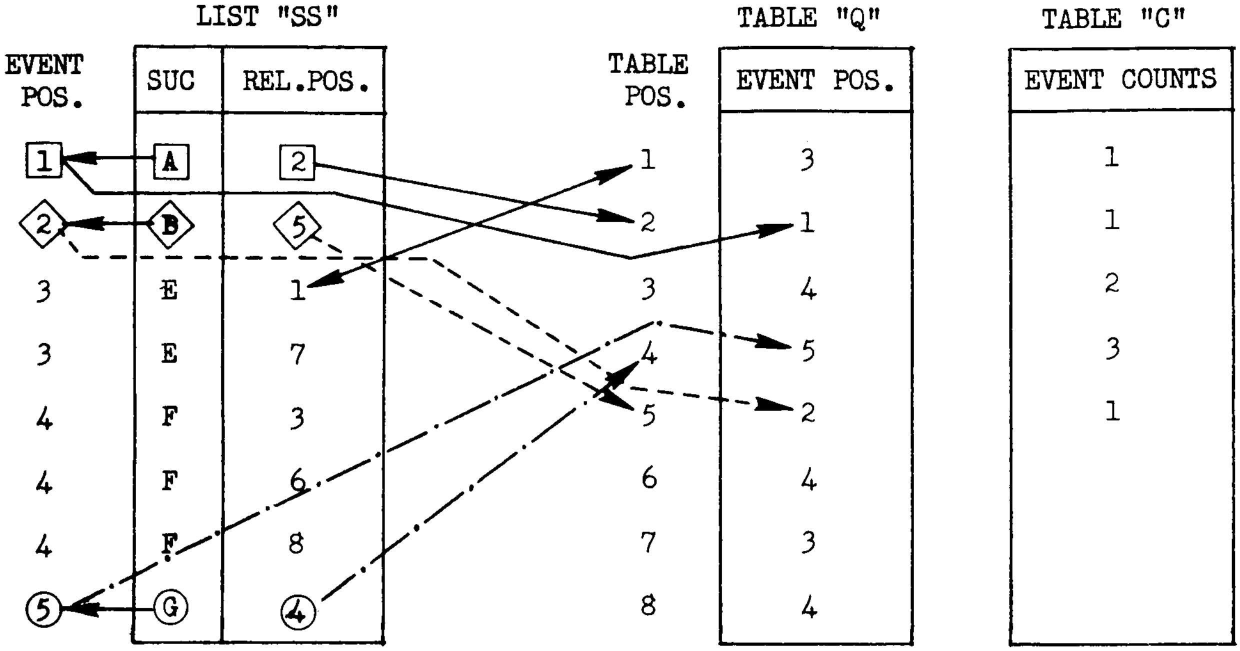
LIST "SP"

SUC	REL. POS.
E	1
A	2
F	3
G	4
B	5
F	6
E	7
F	8

LIST "SS"

SUC	REL.POS.
A	2
B	5
E	1
E	7
F	3
F	6
F	8
G	4

Figure 3



LIST "SS"

EVENT POS.	SUC	REL.POS.
1	A	2
2	B	5
3	E	1
3	E	7
4	F	3
4	F	6
4	F	8
5	G	4

TABLE "Q"

TABLE POS.	EVENT POS.
1	3
2	1
3	4
4	5
5	2
6	4
7	3
8	4

TABLE "C"

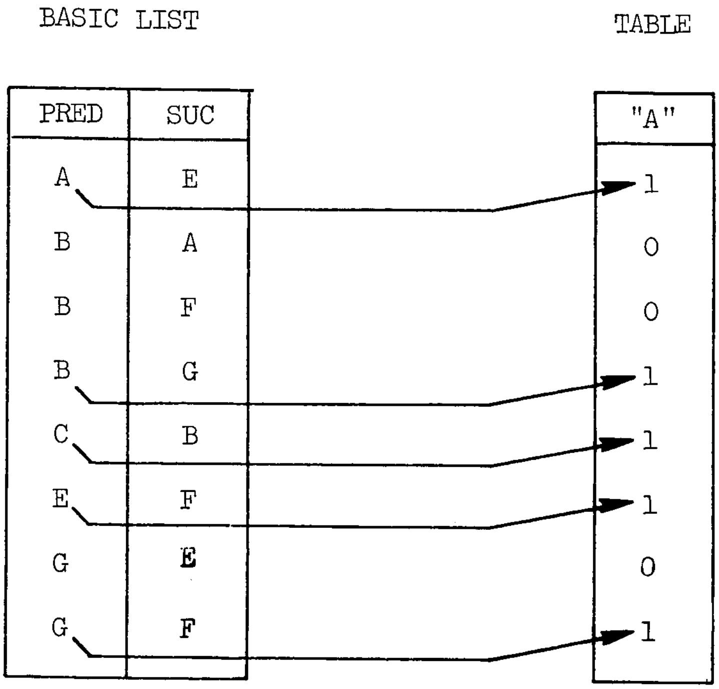
EVENT COUNTS
1
1
2
3
1

Figure 4

Two of the required tables, Table "Q" and Table "C", can be developed from one pass of list "SS". Figure 4 shows how Table "Q" is formed. The predecessor sequence number in list "SS" relates to the location in Table "Q" in which to store an event position number. For instance, Event A is the first unique event in list "SS" therefore, since A has a relative position of 2, a one (1) is stored in the second location of Table "Q". Event B is the second event in list "SS"; therefore, a two (2) is placed in the fifth position of Table "Q".

As another example, Event G is the fifth unique event in list "SS". Therefore, a five (5) is placed in location four of Table "Q". At the same time that Table "Q" is being constructed, Table "C" can be formed. Table "C" is simply a count of the number of times each event appears in Table "SS". This turns out to be a count of the number of predecessors for each event.

Block Ending Flags



BASIC LIST

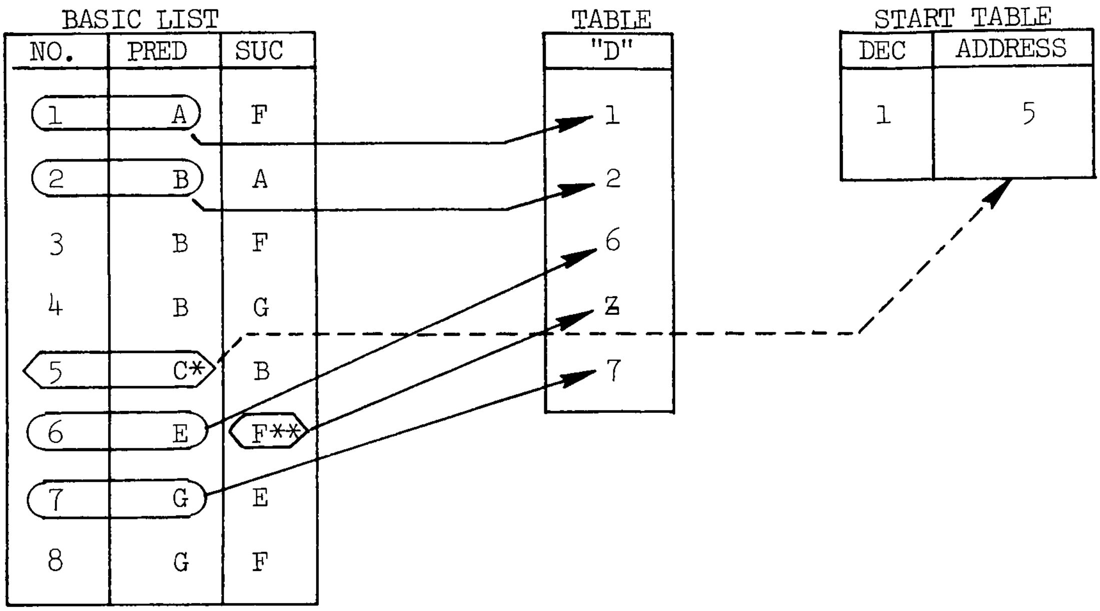
PRED	SUC
A	E
B	A
B	F
B	G
C	B
E	F
G	E
G	F

TABLE

"A"
1
0
0
1
1
1
0
1

Figure 5

Figures 5 and 6 are developed in one pass of the basic list. Figure 5 illustrates the relationship between the predecessor list and Table "A". Table "A" is simply a table showing where each predecessor event block ends.



BASIC LIST

NO.	PRED	SUC
1	A	F
2	B	A
3	B	F
4	B	G
5	C*	B
6	E	F**
7	G	E
8	G	F

TABLE "D"

"D"
1
2
6
Z
7

START TABLE

DEC	ADDRESS
1	5

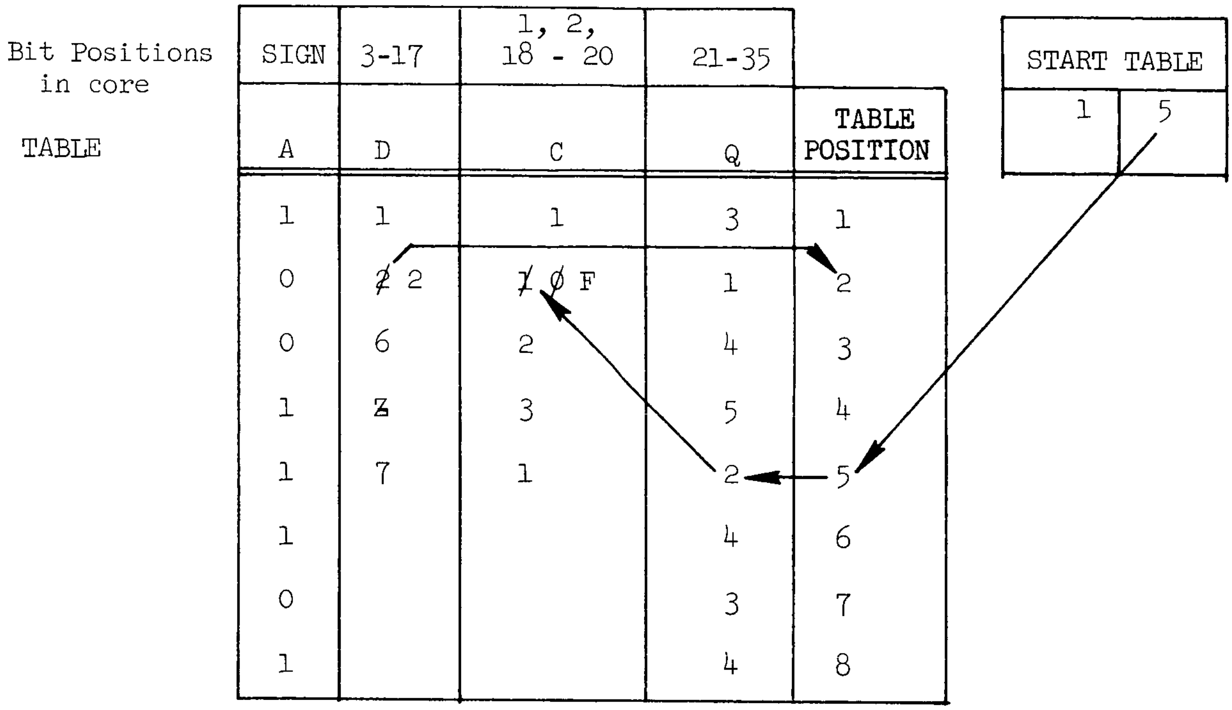
Figure 6

Table "D", figure 6, contains the locations in the basic list where each event set (A, B, C, etc.) begins. Ending events are also included. Figure 6 illustrates how Table "D" is formed. Following the logic in Figure 6, we note that not only are we building Table "D", but at the same time we are finding the starting and ending events. The criterion for determining the starting event is that the event is not used as a successor. Conversely, the criterion for determining an ending event is that it is not used as a predecessor.

The start event table has not been mentioned until now because it is not a part of the four tables which are necessary to execute the basic algorithm. There is one word of memory allotted for each start event. The relative location of each start event in the predecessor list is stored in the address, and the topological sequence number assigned to that start event is stored in the decrement.

It should be noted that as each of the four tables is being formed, it is packed in core. As construction of Table "D" is completed, all tables have been stored in core and are ready for execution of the basic algorithm.

CORE STORAGE AT START OF BASIC ALGORITHM



Bit Positions in core	SIGN	3-17	1, 2, 18 - 20	21-35	
TABLE	A	D	C	Q	TABLE POSITION
	1	1	1	3	1
	0	~~2~~ 2	~~1~~ ~~0~~ F	1	2
	0	6	2	4	3
	1	Z	3	5	4
	1	7	1	2	5
	1			4	6
	0			3	7
	1			4	8

START TABLE	
1	5

Figure 7

Table "A" is in the sign position. Table "C" is in bits one and two and the tag (bits 18 through 20). Table "D" is in the decrement (bits 3 through 17), and Table "Q" is in the address (bits 21 through 35).

BASIC ALGORITHM

Execution of the basic algorithm is begun by ordering all the events in the START table. As shown in Figure 7, the first and only entry in the START table refers to the fifth position in Table "Q", which in turn refers to the count in the second position of Table "C". A one is subtracted from this count thus reducing it to zero. Since there are no more entries in the START table, the next step is to search the "C" Table for zeros. A zero is found in the second position of Table "C". A two is assigned in Table D as the topological number of that event (the START event was assigned topological number (1)) and a flag (F) is set in Table "C" showing that the event has already been ordered. The two in the second position of Table "D" then refers to the second position in Table "Q" and the cycle is complete.

BASIC ALGORITHM #1

TABLE A	D	C	Q	TABLE POSITION
1	1	~~1~~,0	3	1
0	~~7~~,2	F	1	2
0	6	2	4	3
1	Z	~~3~~ 2	5	4
1	7	~~1~~ 0	2	5
1			4	6
0			3	7
1			4	8

Figure 8

Starting in the next cycle, the one in the second position of Table "Q" refers to the first location in Table "C" and a one (1) is subtracted from the count. There is no block ending flag for two more positions in the "A" Table so the same operation is performed using the next two values in Table "Q". That is, a one (1) is subtracted from the count in the fourth and fifth positions of the "C" Table. (See Figure 8)

BASIC ALGORITHM #2

TABLE A	D	C	Q	TABLE POSITION
1	~~1~~,3	F	3	1
0	2	F	1	2
0	6	2	4	3
1	Z	2	5	4
1	7	0	2	5
1			4	6
0			3	7
1			4	8

Figure 9

Once again the "C" Table is searched for the zeros. A zero is found at location one (1) and an "event ordered flag" is set in Table "C". Note in Figure 9 that the value in Table "D" has to be picked up and saved temporarily while a topological number is stored in its place. Then the saved value is used as an indicator of the location in Table "Q" to go to next. Figure 10 illustrates how the tables will look in memory before and after completion of the basic algorithm. Notice at completion, all entries in the "C" Table have a flag and all entries in the "D" Table represent a topological number of some event.

Actually, the mechanical task of each table used in the basic algorithm is as follows:

Table "A" controls, through block ended flags, the number of counts in the "C" Table from which a one (1) is subtracted on each cycle of the basic algorithm.

Table "D" indicates the location in Table "Q" from which the next cycle will be started.

Table "Q" indicates the location or locations in Table "C" from which a one (1) is to be subtracted.

Table "C" indicates, as its counts become zero, the location in Table "D" in which to store topological numbers. When there are no more zero counts left in Table "C", the basic algorithm is complete, that is, unless you are in a loop. Note in Figure 10 that Table "D" now has topological numbers for all the predecessor events of the network (except starting events), and that it also has topological numbers for ending events. Ending events are assigned the largest fixed point number that will fit in the decrement, that is 32,767.

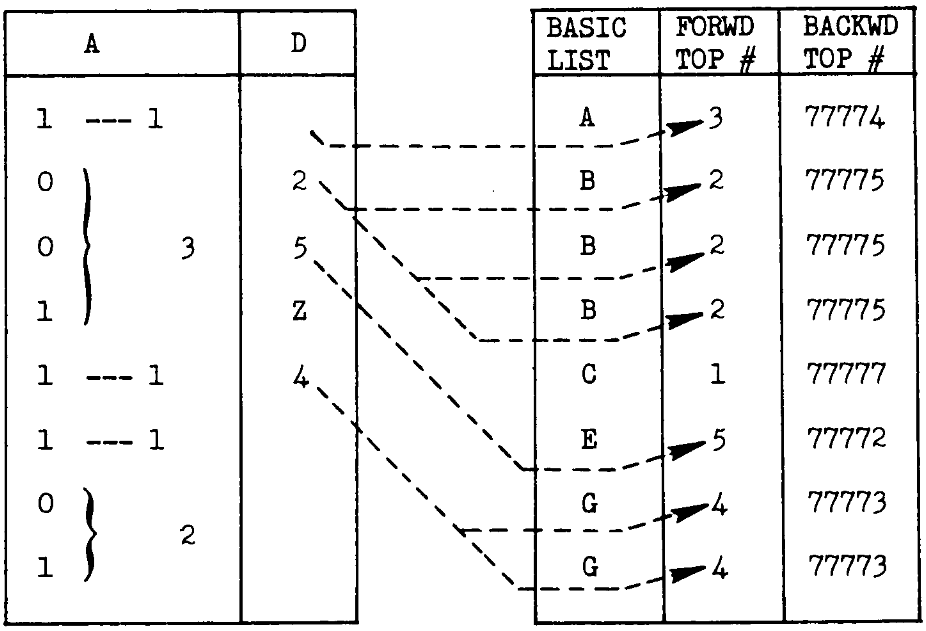
TABLE	A	D		C		Q
		Before	After	Before	After	
	1	1	3	1	F	3
	0	2	2	1	F	1
	0	6	5	2	F	4
	1	Z	Z	3	F	5
	1	7	4	1	F	2
	1	BEFORE	AFTER	BEFORE	AFTER	4
	0					3
	1					4

Figure 10

POSTPROCESSOR

Using Table "A" as a guide (See Figure 11), the basic activity list is passed against Table "D". As a topological number is assigned to an activity record, its complement is also assigned, thereby topologically ordering the networks forward and backward at the same time. The fact that a backward ordering can be obtained free is very significant. This, of course, is an important characteristic of the algorithm, because if its backward topological order was not the complement of the forward order, the whole ordering process would have to be done over, only in reverse, and it would take twice as long to order the network.

POST PROCESSOR



A		D
1 --- 1		
0		2
0	3	5
1		Z
1 --- 1		4
1 --- 1		
0		
1	2	

BASIC LIST	FORWD TOP #	BACKWD TOP #
A	3	77774
B	2	77775
B	2	77775
B	2	77775
C	1	77777
E	5	77772
G	4	77773
G	4	77773

Figure 11

LOOP DIAGNOSTIC

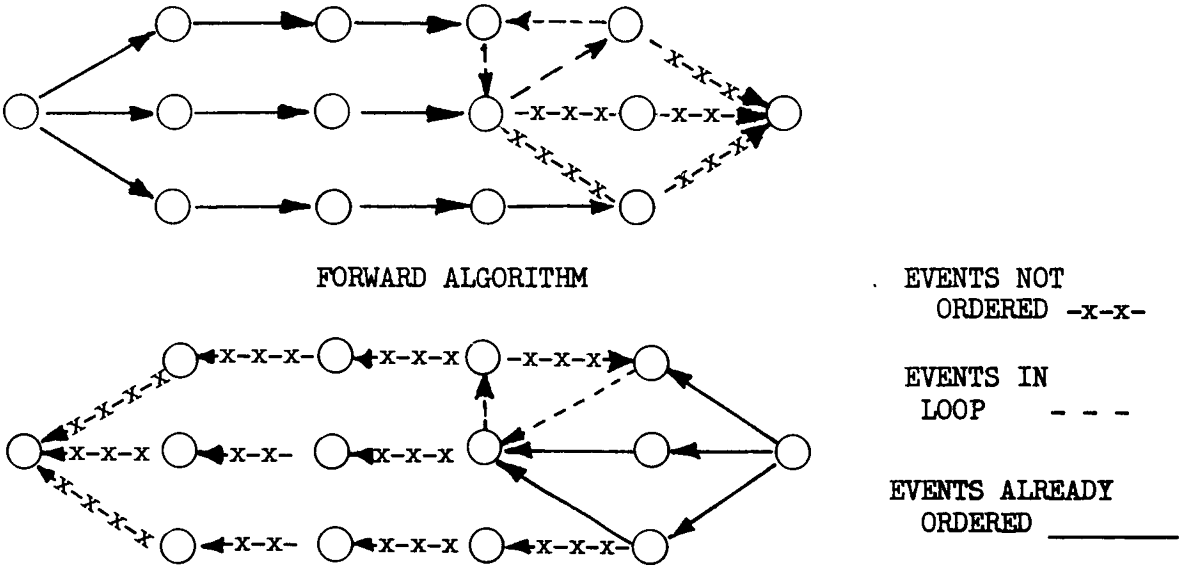


Figure 12

LOOP DETECTION

When no more zeros exist in Table "C", the basic algorithm is terminated. Either all events have been ordered and all entries in Table "C" contain a flag, or there is a loop in the network. In the latter case, those entries in Table "C" which are not flagged represent all events in the loop. However, events leading out of the loop are also included. This complicates the situation because at this point there is no way of distinguishing between those events in the loop and those leading out of it. To remove the unnecessary events, the basic algorithm is reversed. Using the flags in Table "C" as a guide, only those events not ordered by the forward algorithm are ordered backward. At the completion of the reverse algorithm, all events corresponding to unflagged entries in Table "C" are in a loop.

TIMING

The major concern in using Dr. Kahn's method was its efficiency. As the topological sort was programmed, two (2) files were written on tape and read back at least once. Also, one sort was performed. This was accomplished in three passes or core loads. Figure 13 shows the amount of time it takes for the 7094 Computer to perform the topological sort.

NET WORK SIZE	5,000 ACTIVITIES	12,800 ACTIVITIES	17,000 ACTIVITIES	
Edit & binary conversion	19 sec	48 sec	1 min 6 sec	
Preprocessor	55 sec	1 min 55 sec	2 min 35 sec	TOPOLOGICAL SORT
Basic Algorithm & Post Processor	20 sec	48 sec	1 min	

Figure 13

Of course, all our input and output is blocked and double buffered and at one point in the preprocessor, a complete table is written on tape as one record.

OUR EXPERIENCES IN FIELD TESTING NASA PERT "C"

by

Harry Parsh
McDonnell Automation Center
St. Louis, Missouri

First I would like to give you a short history of the NASA Programs at McDonnell. We have been using the NASA Programs since they were first distributed. We have used them as a result of a contractual requirement and also on a voluntary basis. I would say that 80% of our PERT effort is connected with the NASA PERT Program. In the past it has shown itself to be a reasonably smooth running program exhibiting few tape troubles.

I tested the additional features of the new program and found them to work satisfactorially. I did not check the loop diagnostic or the activity card column 1 code of value 4. I feel the main improvement is from the PERT user's standpoint in the excellent ability to scale down the amount of output given to approximately that which is desired.

Another improvement is a reduction in machine time required. I had always thought NASA "B" to be a very efficient program, and I was quite surprised when NASA "C" was significantly better. For a comparison of the 2 programs, I used a network of 1746 activities and requested only the report of Paths of Criticality. NASA "B" ran 4.31 minutes while NASA "C" ran 2.18 minutes. I understand the new program has more efficient sorts in it, but since I asked for only one report I feel this very favorable time improvement is due primarily to the method of topological sequencing chosen by the NASA people.

I also feel that the tremendous increase in capacity may someday prove to be very beneficial.

Objections that I have to the program seem to be of a minor nature. I will mention five of them in what I feel is the order of their decreasing importance.

1. Three time estimate capability--The reason this one is first is that it seems this would be easy to add and from our viewpoint is a very desirable feature primarily due to some of our contractual requirements.

2. File maintenance--Last Monday I took a rough count of our cards presently being run under the NASA program. There were 27 networks comprising 1300 activities. These are generally on a two-week

reporting cycle. Now we do all of this file maintenance on EAM equipment and it does seem to operate fairly well. However, I do feel that a well designed, easy to use file maintenance would reduce the number of errors and correspondingly lower the average turn around time. I feel also that the file maintenance should be done on the 7090. Peripheral hardware situations differ from installation to installation. Even within an installation as happens to be the case with our area at McDonnell there may be only one 1401 equipped to do file maintenance.

3. A more flexible calendar to incorporate weekends and holidays--The value of this lies not so much in more accurate calculations but in being able to present management with more meaningful information.

4. We have one manager in particular who wants a report on all slack paths up to +2 weeks. It seems like this might be a little better approach than to choose the number of slack paths desired. Maybe even a combination of these two ideas would be best.

5. I noticed one significant oversight in the manual--Page 29, paragraph 4B, Listings Within Each Run. Between Items 3 and 4 should be inserted "By Schedule or Latest Allowable Date."

In summary I would like to say that I feel certain that we will adopt NASA "C". The improved efficiency and output characteristics should produce a reduction in turn around time to be of significant importance to our organization.

OUR EXPERIENCES IN FIELD TESTING NASA PERT "C"

by

Homer L. Smith
The Boeing Company
Seattle, Washington

When NASA asked the Boeing Company to test the NASA PERT "C" Computer Program, it was with the full knowledge that we had never seen the program or any of its predecessors before. The difficulties that we encountered in testing this program have all been corrected but are discussed here because we feel that they are typical of those encountered in testing any large PERT computer system.

The Boeing Company IBM 7094's have 3 channels, and 6 tapes per channel. The monitor systems used are the FORTRAN monitor for scientific work and IBSYS Version 11 for information systems. We are trying to get as much of our work load under IBSYS as soon as possible. Our computers are located in a separate facility and all work flow to and from this facility is handled by messengers. Peripheral processing is by 1401's using standard software. The load on our 1401's is such that getting 1401 time on prime shift is next to impossible due to the priority of normal peripheral processing. Since our normal output from a PERT run is at least 50,000 lines, you can understand what the effect on normal workload may be.

When we received the NASA PERT "C" program from MSC, we had a little re-educating to do because we have been leaning heavily on COBOL for data processing and management information systems. Since we were more familiar with IBSYS and because the rough draft manual we received said we could, we embarked upon the task of running the program under IBSYS Version 11.

The first problem we encountered was that we hadn't received the B5 loader card with the manual. This obstacle was quickly overcome after preparing our own. We then discovered the channel problem. The program was written for a 2-channel machine and we didn't have enough tape units on channels A&B to run the program. A call to Houston initiated a flurry of activity, and a few days later we received some 500 instructions in cards by Data Phone to be inserted into the program.

While waiting for the changes from MSC we tried to execute the program in its original two channel version by using the program as a stand-alone system. This venture was not very successful due to loops and stops and after much head scratching we discovered that the channel traps were not disabled. This trapping caused some very interesting gyrations. This problem was easily solved in the B5 loader, and we got a successful run to the point where we ran out of tapes.

By this time we had the three channel version ready so with renewed confidence we attempted to run. Timing problems caused the run to abort, but a call to MSC brought the single error to light and after correcting it, we were running.

After making runs with several small networks that tested most of the options, we ran a larger network of 5,200 activities. This network has been run several times with various outputs. All runs have been successful.

There are only two things that we know have not been tested. Since we do not yet have an SC4020, we have not tried the microfilm output. The second thing that we have not tested is the secondary merge in the report generations. Since the secondary merge does not require any more instructions than the primary merge, we did not feel it necessary to test this feature at this time.

Our experience with this program and its associated documentation have led us to make the following recommendations:

1. A document be made available containing more detailed programming information. It should contain deck setup diagrams, detailed flow charts, and a description of how the calculations are made and how schedule and actual dates are handled in the program.

2. Additional information should be made available on the I/O & SORT subroutines and the information flow (tape formats) in the various passes of the program.

3. All machine stops should be removed and replaced by a comment (both on and off-line) and a transfer to the monitor.

4. Provision should be made for saving the IBSYS monitor, if used.

5. The calendar routine should be made flexible enough to compensate for holidays and weekends, if desired.

6. File maintenance should be provided on the 7094 for those who need it, and input to this routine should not be in a specified order.

7. Serious consideration should be given to eventual use of standard IBM I/O and sort routines.

In conclusion I would say that we feel that this program has features that make it a valuable contribution to the management information field. Of special significance are the report cut-off, restart, and end run features. The capacity of the program is more than adequate, but the addition of a network summarization feature would prove useful. The help that we received during the test was excellent, and anyone who can make 500 plus changes to a 25,000 instruction program, without local testing, send them by Data Phone, and have them run with only one error deserves a medal.

NASA PERT FORTRAN IV PROGRAM

by

Elizabeth Ryan and Ross Bainbridge
NASA Lewis Research Center
Cleveland, Ohio

INTRODUCTION

The program to be described is a PERT time program written entirely in compiler language and with a capacity in excess of 30,000 activities. The program was written at Lewis Research Center with the assistance of Hans Bremer and N. H. Dillard of Goddard Space Flight Center. (Topological ordering by the pushdown technique is described in the conference paper by Hans Bremer).

To best understand the reasons for production of this new program, a brief review of the history for writing the program and also of the programming philosophy at Lewis Research Center will be presented.

The project was proposed by Lewis in March of last year as a solution to several problems that had arisen in using the machine-coded programs. Briefly, these problems can be summarized as follows:

(1) The machine-coded program could be run only on one manufacturer's equipment. This required sole source replacement of the computing equipment when replacement was required for only 5 percent of the load.

(2) A great deal of time of systems personnel was being spent in maintaining machine-coded programs and in modifying them each time a systems or handware change was implemented.

(3) The adoption of new hardware by an installation without a change of manufacturer often requires extensive rewriting of the machine-coded programs. For example, the substitution of disks or drums for tapes requires considerable program revision.

It appeared that these problems were typical to the exclusive use of machine-language programs and that compiler-written programs designed to run under a typical monitor system would eliminate these problems. A compiler-written program would have the added advantage of discouraging the use of undocumented binary patching to the program decks. Since modifications to a compiler-written program can much more easily be made simply by recompiling a source deck, language documentation would automatically be provided for all modifications.

Lewis proposed to write this PERT program in FORTRAN IV because it is the compiler that has been implemented by most computer manufacturers, and it is the compiler language most used by industry as well as NASA. It is known that for a given algorithm an optimum machine-coded program is faster than an optimum compiler program. But as the algorithm becomes larger or more complex, practical considerations of time and personnel prevent the production of an optimum machine language program whereas an optimum compiler-written program can still be obtained. For example, Lewis has written several large systems entirely in FORTRAN with excellent results. These include a production IBM 1401 SPS assembler written originally in FORTRAN II and a FORTRAN II compiler and assembler written in FORTRAN II, which were more efficient in running time than the FORTRAN II compiler and assembler on the IBM 7090. In fact, this compiler and assembler have since been converted to FORTRAN IV and are still heavily used in production status producing data reduction programs. A final example of compiler-written programs is the SIFT program written in FORTRAN II, which made most FORTRAN II programs FORTRAN IV compatible.

PROGRAM DEVELOPMENT

In June 1963 the Lewis proposal was approved with the following restrictions:

(1) The program should be written in compiler language. Machine language would be permitted only where large gains could be made in efficiency.

(2) The running time of the new program should not exceed running times of the existing program.

(3) The program should have an ultimate capacity of 30,000 activities. The use of a modular or skeletonizing technique to achieve this would be considered.

(4) The format of input and output data would not be altered.

(5) The program should be compatible with the data processing equipment then being used for PERT throughout NASA.

The first phase of the project - the production of a limited capacity PERT time program entirely in FORTRAN IV - was completed in October. The program, Lewis-Goddard PERT TIME I, has a capacity of 3500 activities and has been distributed to the following installations and manufacturers:

NASA
Goddard Space Flight Center
Langley Research Center
Ames Research Center
Lewis Research Center
George C. Marshall Space Flight Center

Manufacturers
CDC
UNIVAC
Honeywell

Industrial
Bellcomm
Aerojet General
Westinghouse
Goodrich

SHARE

The program is in exclusive production use at Lewis, Ames, Goddard, and Langley. The program is in operation at Aerojet and Bellcomm on an IBM 7040 and IBM 7044. It has been submitted to and is available through SHARE. The number of installations that have received the program through SHARE is not known.

PERFORMANCE DATA

Run times have been considered favorable on all machines used thus far. The following performance data are for the first phase, PERT TIME I, as recorded on an IBM 7094, running tape to tape using 729V tape drives at 800 BPI on two data channels. Times are exclusive of load time and reflect some time savings obtained by the blocking of output at 5 lines per record.

Activities	Outputs	Time, Min
200	3	0.2
200	5	.3
300	4	.4
1000	5	2.5
1600	3	2.5
2830	4	7.5

Time studies were run using the configuration against the NASA PERT Mod-B machine-coded program that was then still in production at Lewis. Comparative timings on the machine indicate that the program is 50 percent faster than the Mod-B PERT machine-coded program for networks under 1100 activities, requiring no output merging; equal in speed for networks between 1100 and 2100 activities, requiring a single output merge; and within 10 percent for networks over 2100 activities and requiring multiple output merges.

Since the original time study was conducted, computer configurations have been switched and the following times for a directly coupled IBM 7094 model II IBM 7040 with a disk, drum, and four model VI tape units can be reported:

Activities	Outputs	Time, min
0 to 1000	3 to 5	Under 0.5
1000 to 2000	3 to 5	Under 2.0
2000 to 3000	3 to 5	Under 5.0

Also on subnetted jobs where no more than a single merge is ever needed, subnetted jobs have been run with a total of 4000 activities in under 6 minutes.

The next set of performance data was provided by Mr. E. Kilroy of Computer Usage Company subcontracted to Bellcomm of Washington, D. C., from runs made on a 7040/44 direct couple system with partial use of disks in place of tape. The overlay feature utilized when running on our 7094 was not necessary at Bellcomm.

Activities	Outputs	Time, min
1839	3	7.4
204	3	.2
1330	2	3.2

Again times do not include load time. Mr. Kilroy also estimated that this one time conversion cost to an IBM 7040/44 system was approximately $3800. This cost included personnel and computing. A complete rewrite of the program would have an estimated cost in excess of $50,000. We feel this is a good illustration of the cost savings of a compiler-written program.

Running times from Aerojet using an IBM 7044 with 10 tape drives, 4 disks, 2 channels, and a 1401 off-line are as follows:

Activities	Outputs	Time, min
81	2	0.8
270	3	1.5
325	4	2.9
1300	4	3.0
2000	4	4.4

This actually shows a 24-percent reduction in running cost over the NASA PERT B, which runs on Aerojet's IBM 7094. These data were supplied by Mr. T. C. Adams, a systems analyst at Aerojet General, Sacramento. The appendix is an internal memorandum written by Mr. Adams in which he evaluated the Lewis-Goddard PERT time program summary on the IBM 7044 versus the Mod-B program on the IBM 7094. We found this evaluation to be very informative as to the use of an IBM 7044 as a PERT management production tool. The ease of modifying the program is attested not only by the variety of machines on which it is running but also by the many features that have been added by individual installations.

EXTENSION OF PROGRAM TO HANDLE LARGER NETWORKS

The second phase of the project was begun in December 1963. Its aim was to build the PERT TIME I program into a program of much greater capacity with several new features. This was done while still retaining ability to process all smaller networks already using the program. That program, Lewis-Goddard PERT TIME II, has been in production and will be made available to NASA installations along with a detailed system manual and a separate looseleaf users manual that can be updated.

The capacity of the PERT TIME II program is in excess of 30,000 activities. The increased capacity is obtained using a subnet technique. It is of interest to note that it is possible to maintain the size of the basic subnet at 2200 activities, which is felt to be more than generous enough for these people having experience with the IBM Cost-Time program with the restriction of a basic subnet size of 750 activities.

At this point it is best to define what is meant by a subnet. A subnet is simply any collection of interrelated activities belonging to a PERT network. In the PERT network shown in figure 1, where the circles represent event points and the connecting lines represent activities, the shaded activities make up a subnet, as do the activities enclosed in the broken lines. Note that there are three event points (with crosses in them) common to both subnets. These events are called interface points between the two subnets. Subnets can be connected only in this way, that is, by one or more interface events. In practical terms, a subnet is often a logical entity of some kind. For instance, in a network representing a project involving four contractors (A, B, C, and D) there could be four subnets each representing the work assigned to one of the four contractors (Figure 2).

To facilitate this usage, it is not required that the interface points have the same event number interior to each subnet in which they appear. This eliminates the necessity of coordinating numbering of common events among many contractors each of whom may be maintaining his own network. To eliminate this, each interface point is given an alphabetic name, the interface label, when its subnet is to be integrated. In figure 2, for example, the interface point that has been labeled I1 may be known in contractor A's subnet as event 5000 and as event 4 in contract B's subnet. With reference to the network as a whole, however, it is simply interfaceevent I1. An equivalence card concept was used to show this relationship. Lewis experience finds these very flexible without adding excessive card input to the program. This will be discussed later in more detail.

PROGRAM FEATURES

A useful feature of the program is the provision for a different type of subnet, the summary network. Suppose the subnet shown in figure 3 below the dotted line is being maintained by a department for its own use. It may be that only those events with upward pointing broken arrows need to be reported to higher management. These events can then be made interfaces to a subnet, which consist only of the interface events. The resulting subnet can be represented by the figure above the dotted line. It is a summary of the original subnet or subnets and shows not only event relationship but also PERT network logical flow as indicated by the solid arrows. The program will compute time estimates along each path of the summary network using the detailed paths from the original. If requested, activity cards for the summary network can be punched out with delta time estimates. This deck can then be sent on to the higher management to be run as this department's subnet in a larger network.

For truly effective management reporting, for example, summarized reports are necessary as high levels of management are reached. This is illustrated in figure 4 as a pyramid of PERT reports.

Now with the ability to place time estimates into an output form (the same form as the standard NASA input) with activity times that truly reflect the interrelationships of the base network, a method of even further upward reporting is established. As shown in figure 5, the program could support basic networks of 30,000 activities at Lewis, Goddard, and Langley. In turn, various summaries are sent as basic subnets to the program running on a computer used by NASA Headquarters. The program could support a base in the example of 90,000 activities and still present top management only the few hundred activities needed at the top level of command.

People who have had much contact with PERT networks are well acquainted with dummy activities. There are several kinds of dummy activities, but this is one of the most popular (see Figure 6).

The activity connecting events 7 and 17 is a dummy inserted for the express purpose of inventing a place to hang the label END TESTING. What is actually needed here is a way of identifying event 7 as the end of testing. The insertion of the dummy has added an extra event and activity to the network. This practice as a substitute for event nomenclature is quite common and can cause a significant increase to the size of a network. While large PERT networks may be regarded as a sort of status symbol, they can be expensive.

By using the PERT TIME II program, the event 7 can be named directly by the use of an event card:

```
                    00000070000007        END TESTING
```

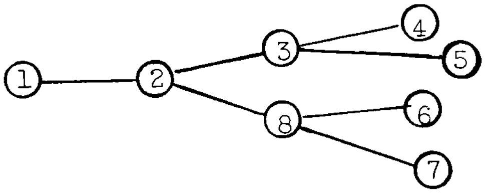
The event number is entered in both the predecessor and successor columns, and nomenclature appears in the normal field. At report time, the event will appear in normal sort order with its expected and allowed dates and slack. The event card does not in any way enter into the PERT calculation and so does not increase the size of the network.

The updating or file maintenance technique used in PERT TIME II also represents a new approach. Previously the master file has been nothing more than a tape bearing the activity cards for a given network. When it was desired to change the network, the tape was first updated to obtain a new master file, and the new master file was used as input for a complete reexecution of the network. The PERT TIME II program performs updating as a part of the normal PERT run, thus eliminating duplication of operations. A tape developed as part of the PERT calculation is used as the master file. This tape contains not only the activity cards (in blocked form) but also all other information needed to make reports directly from the tape without recalculation. All this information is separated by subnet. Since many times not all subnets need be changed on a given update run, only those that are changed need be recalculated. The master tape is read only once as updating and recalculation of a subnet are overlapped. In addition to providing a fast and efficient update, this technique eliminates dependence on the availability of a second computer.

As a further aid to maintaining networks, completed activities can be automatically deleted from the master file as an option. This feature has two important results. First, by eliminating past activities that no longer alter the project schedule, it reduces the effective in-core size of the network. Secondly, it has been found that a great deal of updating is done for the purpose of removing completed activities. This type of routine updating can now be completely eliminated.

TOPOLOGICAL PROCEDURES

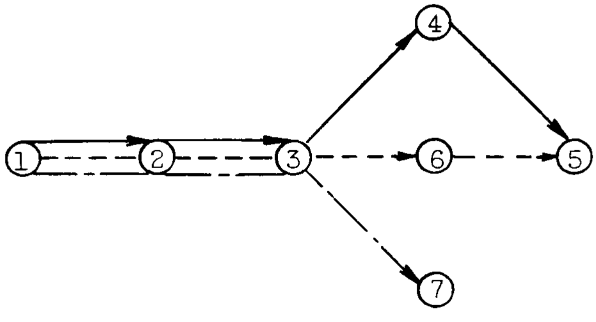
The topological or network analyzing procedures used in the Lewis-Goddard PERT time program are not the familiar topological sorting techniques used in other PERT programs. The technique used here is an application of pushdown lists or tables more commonly used in compilers and recursive routines. The pushdown table is actually a memory device used to remember decision points or alternate decision routes. For example, in the game or decision tree



it is often desirable to know the branches (paths or decisions in a game tree sense) not taken at points 2, 3 and 8. A pushdown table is used to do this by placing the alternate routes not taken at 2, 3, and 8 into a table. Upon retracing the path that actually was followed, the last nonentered branch would appear at the last entered position of the pushdown table. By extracting this branch, the alternative decision or path can also be analyzed. For example, in the figure taking path 1 to 2 to 3 to 4 would result in entering in order in a pushdown table 2 to 8 and 3 to 5. Working back from the end point 4 would result in looking at branch 3 to 5 and then the path from branch 2 to'8.

It now becomes apparent that the game tree figure actually can represent a PERT time or cost network with the decision points 1, 2, 3, etc. as events and the decisions or paths between them 1 to 2, 2 to 3, etc. as activities. Program adaptation of the pushdown table to analysis of PERT networks can actually be divided into three steps. The first step is getting the branch activities into the pushdown table. The second step with taking the last activity from the pushdown table and placing it into another table (the path list) that in final form contains all the activities on a particular network path. The third step then reduces the path list until a branch or start event is located on the pushdown table.

This procedure can be illustrated as follows:



The activities in a table are as follows:

1 to 2
2 to 3
3 to 4
3 to 6
3 to 7
4 to 5
6 to 5

The following analysis is performed:

Step	Pushdown table	Path list
(1)	1 - 2	-----
(2)	-----	1 - 2
(1)	2 - 3	1 - 2
(2)	-----	1 - 2 2 - 3
(1)	3 - 4 3 - 6 3 - 7	1 - 2 2 - 3
(2)	3 - 4 3 - 6	1 - 2 2 - 3 3 - 7

End event 7 encountered with path ——→ is located in the path list. Expected and allowed times can be calculated at this point.

Step	Pushdown table	Path list
(3)	3 - 4 3 - 6	1 - 2 2 - 3
(2)	3 - 4	1 - 2 2 - 3 3 - 6
(1)	3 - 4 6 - 5	1 - 2 2 - 3 3 - 6
(2)	3 - 4	1 - 2 2 - 3 3 - 6 6 - 5

End event 5 encountered with — — — — — → is located in the path list. Expected and allowed times can be calculated at this point.

Step	Pushdown table	Path list
(3)	3 - 4	1 - 2 2 - 3 3 - 6
(3)	3 - 4	1 - 2 2 - 3
(2)	-----	1 - 2 2 - 3 3 - 4
(1)	4 - 5	1 - 2 2 - 3 3 - 4
(2)	-----	1 - 2 2 - 3 3 - 4 4 - 5

End event 5 again encountered with path ————→ is located in path list. Expected and allowed times can be calculated at this point. Step (3) now finds the activity list complete and will then go on to another start or if no other starts exist into another program phase.

The expected times are calculated as a path is completed by the use of a table of events. These event tables are used to keep expected and allowed times for each network event. The expected times are forward calculated and replace the previously calculated event expected time in the events table (TSUPE) only when the expected time now being calculated is greater. Allowed times are calculated from the last to the first event with replacement of the previously calculated allowed time in the events table (TSUPL) only when the currently calculated allowed time is smaller.

When output reports are required, it becomes a simple matter to interrogate the events tables to get the predecessor and successor event times.

The following methods were used to modify the basic topological procedure and to increase its efficiency:

(1) Sequential numbering of the events eliminated events table searching. This sequential numbering also eliminated the necessity for retaining internally the actual event numbers.

(2) Retention of the activity position counter in an events table eliminates any activity table searches. This in turn eliminates the necessity for retaining internally the predecessor event of an activity.

(3) Experimentation with the expected and allowed time calculations resulted in the determination that if an expected time was less than or an allowed time greater than the previous value found in the events tables, then the path analysis could be terminated at that point. This innovation results in a considerable savings in actual internal computing times.

By introducing the pushdown table techniques into the processing of PERT networks, by doing as much in core processing as possible, and by limiting table searching and unnecessary calculation, the FORTRAN IV program developed into a highly efficient topological technique. This technique also makes modular·networks easier to analyze and gives flexibility in experimentation with faster methods.

REPORTING AND SORTING TECHNIQUES

In preparing reports it is necessary to determine the order, with respect to several possible formats, of the activity records that make up each subnet. Because this ordering must be performed many times during the execution of any network, the procedure used must be as efficient as possible. The ordering method developed for use in Lewis-Goddard PERT time is now described.

The activity buffer into which the activity records have been placed constitutes a table of activities and their associated information. For each activity on the network there is an activity record and each record contains several storage words of information about its activity. Each item of activity information (predecessor and event numbers, expected and allowed dates, slack, department code, etc.) is assigned a fixed position in the activity record. With each item of information, then, can be associated two subscripts; the first refers to the position of its activity record in relation to all other records, and the second to the particular item's position relative to all other activity information in the record.

An item of information pertaining to the 10th activity and which was assigned the 4th word in the activity record would have subscripts 10 and 4. That same item of information about activity 25 would have subscripts 25 and 4. Rather than rearranging the activity records themselves, which would be costly both in terms of execution time and core storage usage, the ordering routine rearranges their associated subscripts. At the termination of the ordering procedure there will have been produced a list of subscripts whose order indicates the order of their associated activity records with respect to the given key.

The initial phase of the process is a scanning of the activity keys to determine the extent of natural order as the records lie in core; both ascending and descending order is detected. The following list is constructed. Position 1 of the list contains the number of activity records that make up the first sequence of ordered records - the sign is made negative to indicate ascending order or positive to indicate descending order. The second position refers in the same way to the second sequence and so on, so that if the activity buffer consists of n such sequences, there will be n entries in the list. (If the activities lie in the buffer as shown in step 1, the list produced would be as shown in $LIST_1$. The first four activities are in ascending order as are the next 3. The four activities following the second sequence, however, are in descending order so that the entry is positive. The 25 activity records consist of 7 sequences as described in $LIST_1$).

The remainder of the ordering procedure consists of combining consecutive pairs of sequences to form half as many sequences of combined length. The smaller activity key from the first sequence is compared to the smaller from the second sequence. The subscript of the activity whose key is smaller is placed in the first position of a second list (depicted in step 2 as $LIST_2$). If the smaller key came from sequence 1, the key for the next activity in sequence 1 is compared to the first activity's key in sequence 2. The subscript of the smaller is placed in the second position of $LIST_2$. Comparisons continue until subscripts of all activities in one of the sequences have been placed in $LIST_2$. The subscripts from the remaining sequence are then placed in $LIST_2$ and the combining process is repeated for the next two sequences. As each pair is combined, $LIST_1$ is revised to reflect the combined length of the sequences. (Step 2 shows $LIST_1$ and $LIST_2$ following the first stage of sorting whereby the 7 original sequences were reduced to 4. $LIST_1$ then indicates that the activities associated with the first 7 subscripts form a sequence as do the activities associated with the next 10, etc. All entries in $LIST_1$ are now left positive since after the first combination pass all sequences have been constructed in ascending order.)

The four sequences given by $LIST_2$ are now combined in the same manner to produce two sequences that are described by a list of subscripts in $LIST_3$. (The conclusion of this pass is represented in step 3.) Ordinarily at this point, $LIST_3$ together with $LIST_1$ would be used to produce a new list that will be placed in $LIST_2$, so that $LIST_2$ and $LIST_3$ are alternately used and overwritten. In practice, however, once the number of sequences has been reduced to 2 the activity whose key is smaller is simply written as output after each compare.

TABLE I. - ORDERING PROCEDURE

Activities	Step 1		Step 2		Step 3	
	Keys	$LIST_1$	$LIST_1$	$LIST_2$	$LIST_1$	$LIST_3$
1	3	-4	7	1	17	11
2	5	-3	10	5	8	1
3	7	+4	6	2		12
4	8	-6	2	6		5
5	4	+3		3		10
6	6	+3		4		13
7	9	+2		7		2
8	7			11		9
9	5			12		6
10	4			10		14
11	1			13		3
12	3			9		8
13	4			14		15
14	6			8		4
15	7			15		7
16	9			16		16
17	11			17		17
18	10			23		23
19	8			22		25
20	6			20		24
21	7			21		22
22	5			19		20
23	1			18		21
24	4			25		19
25	3			24		18

SUMMARY

With the completion of PERT TIME II Lewis is confident that it has fulfilled its original goals.

1. The program is written in the FORTRAN IV compiler language.

2. The running times do not exceed those of the NASA Mod -B program.

3. The enlarging of the subnet technique has led to extended capabilities. The use of summary networks and the pyramid use of the program at varying levels of management are two examples of this. It is felt that both these features add greatly to the power of a PERT time program and will prove extremely useful to management groups at all levels. It is possible to handle over 30,000 activities with a maximum limit on subnets of 2200 activities.

4. Formats and outputs are compatible with existing standards.

5. The program is capable of running on data processing equipment within NASA. It has been found that the program is easy to use, as evidenced by the number of installations presently using PERT TIME I, and correspondingly easy to implement at individual installations. Additions and modifications can easily be made to adapt the program to the differing requirements of these installations.

6. The entire project was completed by only two programers, each devoting less than 1 man-year.

In short, it is felt that the Lewis-Goddard PERT TIME II is an efficient and powerful program that is easy to use and can easily be implemented. The program should prove valuable especially to computer systems groups who have long needed a standard PERT time program that can be used regardless of hardware or system demands.

APPENDIX

To:	Distribution	Date:	19 May 1964 TCA:mit:Ext.3090 2001A:2360
From:	T. C. Adams		

Subject: Lewis-Goddard PERT TIME-I for the 7040/44

Distribution: S. A. Chappell, F. W. Eagen, J. D. Poulsen, J. C. Richardson, J. V. Rizzo, J. R. Soll, R. E. Stienmuller, E. C. Wolf

Copies to: Sacto: A. Feinberg, R. T. La Sarge, R. W. Lee, R. J. Machado, File
Glendale: D. B. Cyrog

Enclosure: (1) Lewis-Goddard PERT TIME-I for the 7040/44
(2) Lewis-Goddard PERT TIME-I versus PERT System 'B'
(3) 'ASPERT,' NASA PERT TIME System on the 7040/44

I. BACKGROUND

A. The LG-PERT program (Job 24040) for the 7044 was compared to the existing NASA PERT TIME program (Job 1041AA) for running time comparisons. The test data consisted of three 'stacked' networks approximately 400 total activities. The results are below:

	1041AA (7094)	LG-PERT (7044)
Load time	0.16	1.20
Execute time	1.33	1.62
Total time	1.49	2.82
Cost	$295 - $7.35	$180 - $8.45

The results indicate the 1041AA is less expensive to run than the new '44' version. The obvious reason for this is the excessive load time cost which results while using the '44' version of LG-PERT.

B. Based on a statement by us, that this load time could be reduced significantly, Project M-1 has requested Job 24040 be put into production. The request also made provisions for the three (3) minor modifications to LG-PERT proposed in the Lewis-Goddard PERT status memo dated 3 April 1964.

II. STATUS TO DATE

A. The program is now in production with the binary program on the systems library taps. The program is called for by a "$EXECUTE LGPERT" card. The results with the previous test case data, run under the production status conditions, are as follows:

	LG-PERT (7044)
Load time	0.24
Execute time	1.62
Total time	1.86
Cost	$180 - $5.58

B. We encourage you to use this new system. We have demonstrated it is less expensive to run and expect you to take advantage of its new features. (A cost savings of 24 percent was realized in this test case.) Job reports explaining the program can be obtained from the Divisional Librarian (ask for Job 24040). Additional information can be obtained from us (refer to COSMOS Ref. v-II, p-7). An outline of the differences between Job 1041AA and Job 24040 have been included with this letter (Enclosure (2)).

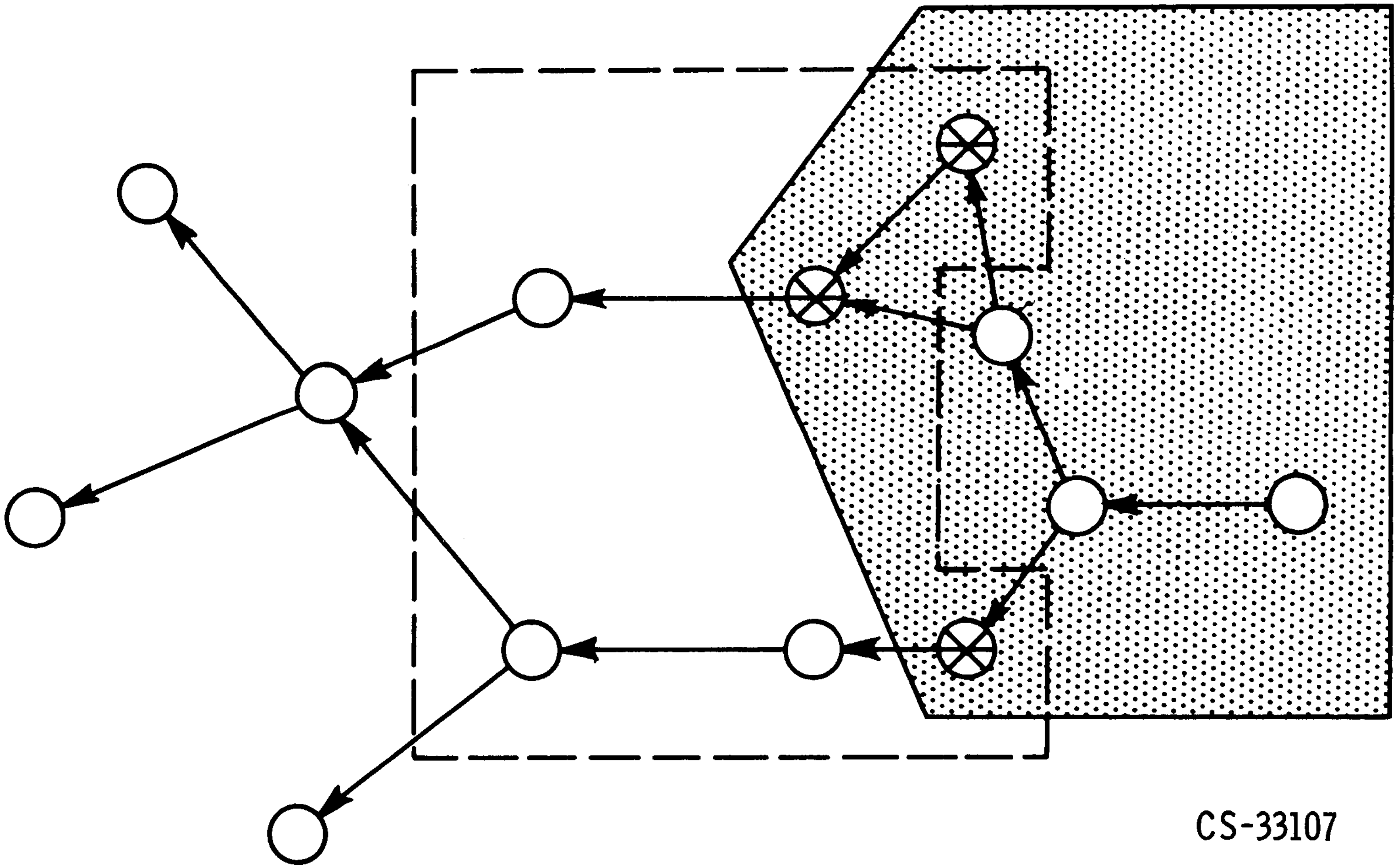
C. An outline of LG-PERT's features and capabilities has been enclosed with this memo. The outline is organized by categories recommended by Corporate Systems for PERT software evaluation (Enclosure (1)).

III. FUTURE ACTIONS

A. For the first time since AGC 5-digit PERT was used by today's NASA PERT users, Computing Sciences is capable of providing programer support on the NASA PERT system. This means the requests for modifications, new features, etc., proposed by you can now be processed.

B. Project M-1 has for some time requested the capability of non-sequential data input to a NASA PERT program. Also they have been interested in Master File Maintenance for NASA PERT. The enclosed paper called 'ASPERT' (COSMOS Ref. v-II, p-9) is an overall PERT systems plan for NASA PERT users. The plan provides an overall framework for PERT user's needs; it is designed on the subsystem concept and thus could be constructed in parts if so desired (Enclosure (3)).

T. C. Adams, Analyst
Management Analysis & Programing
Computing Sciences Division



CS-33107

Figure 1. - PERT network.

E-2714

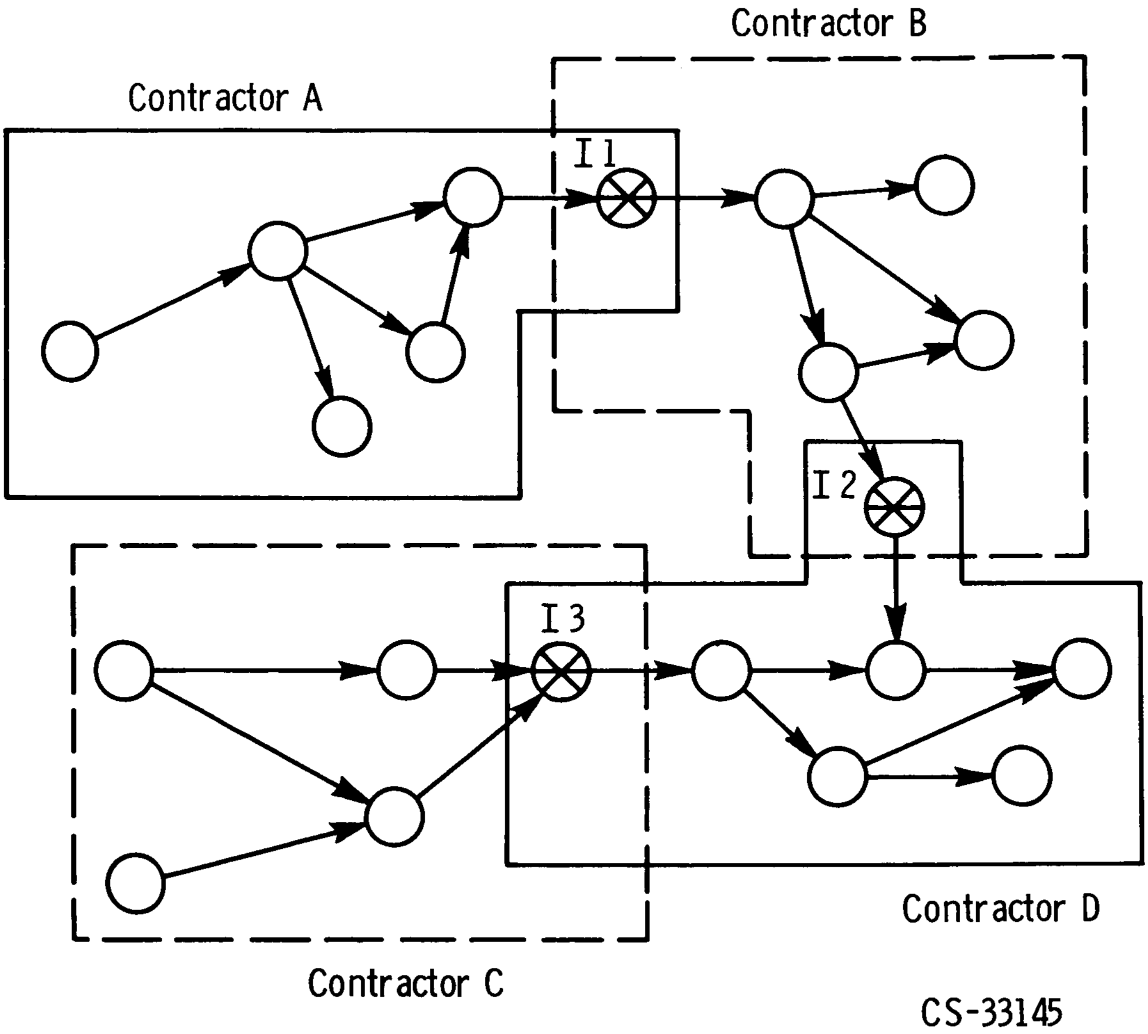
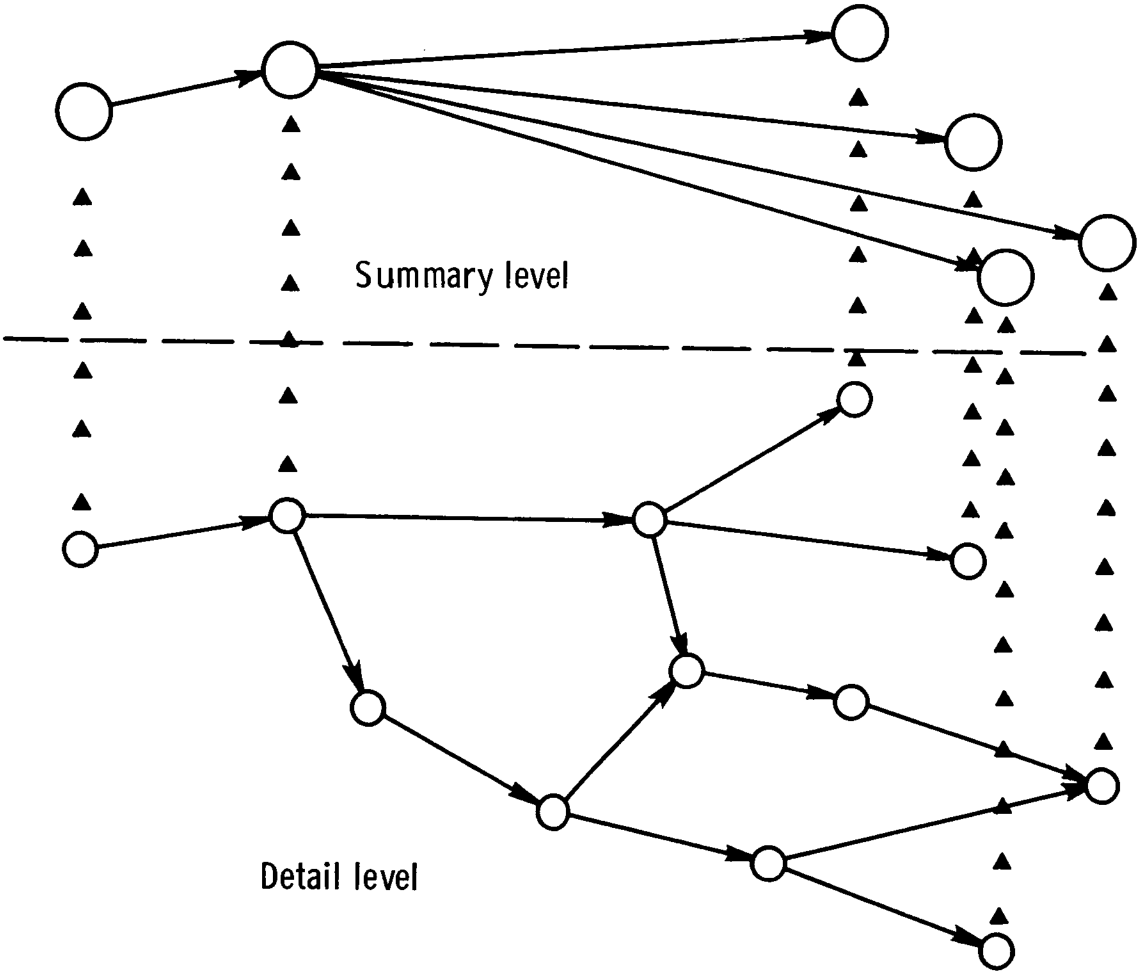


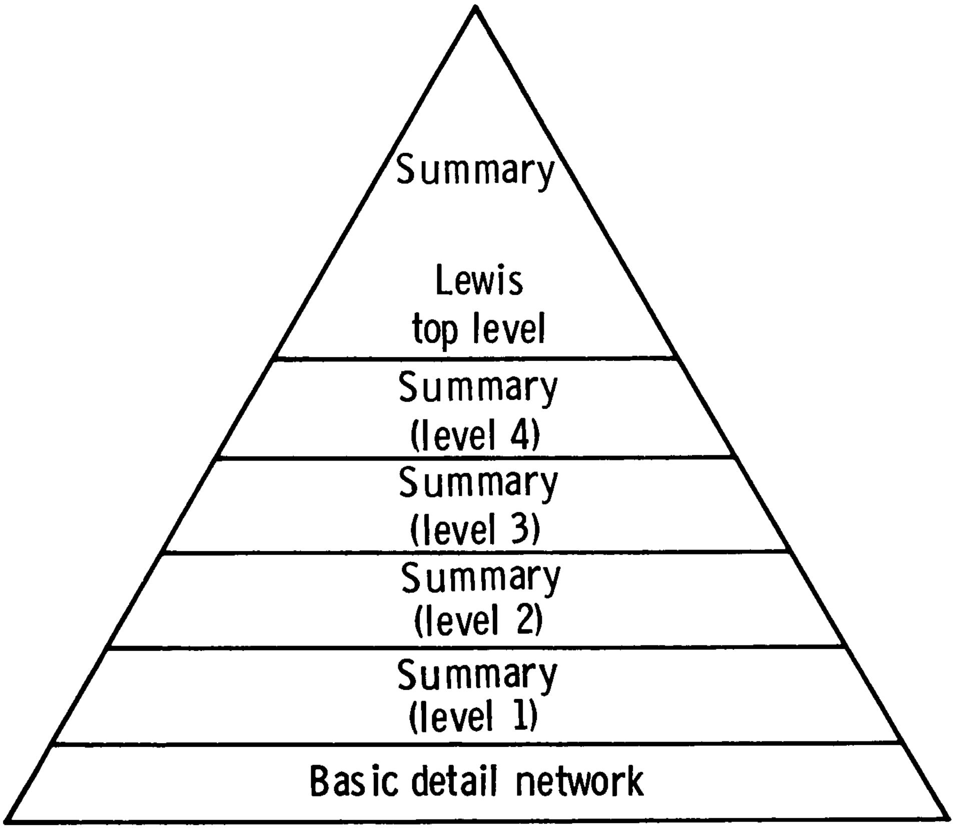
Figure 2. - Project network.



CS-33106

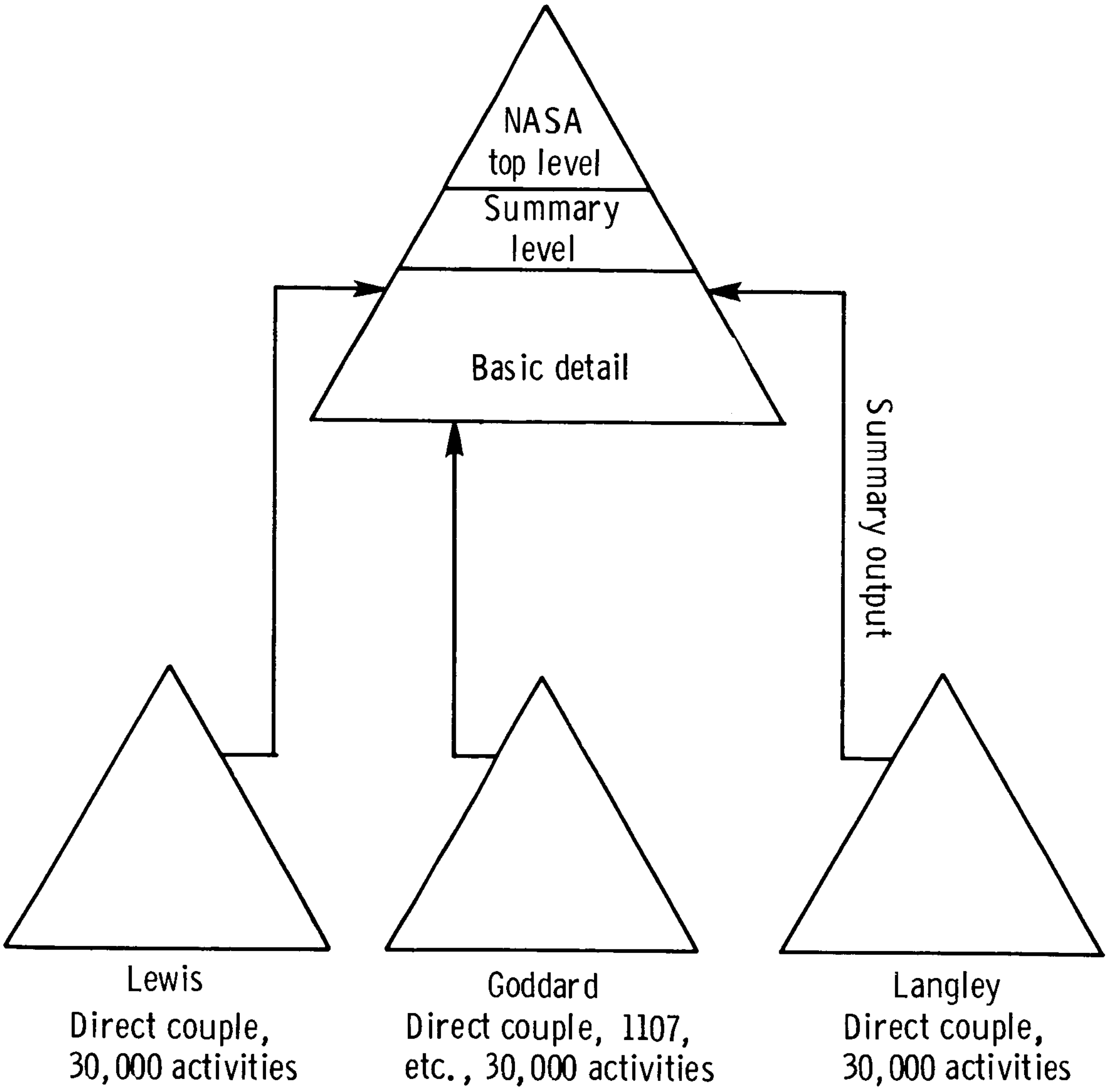
Figure 3. - Summary subnet.

E-2714



CS-33142

Figure 4. - Lewis PERT summary for management.



CS-33143

Figure 5. - Summary output becomes detail level input.

E-2714

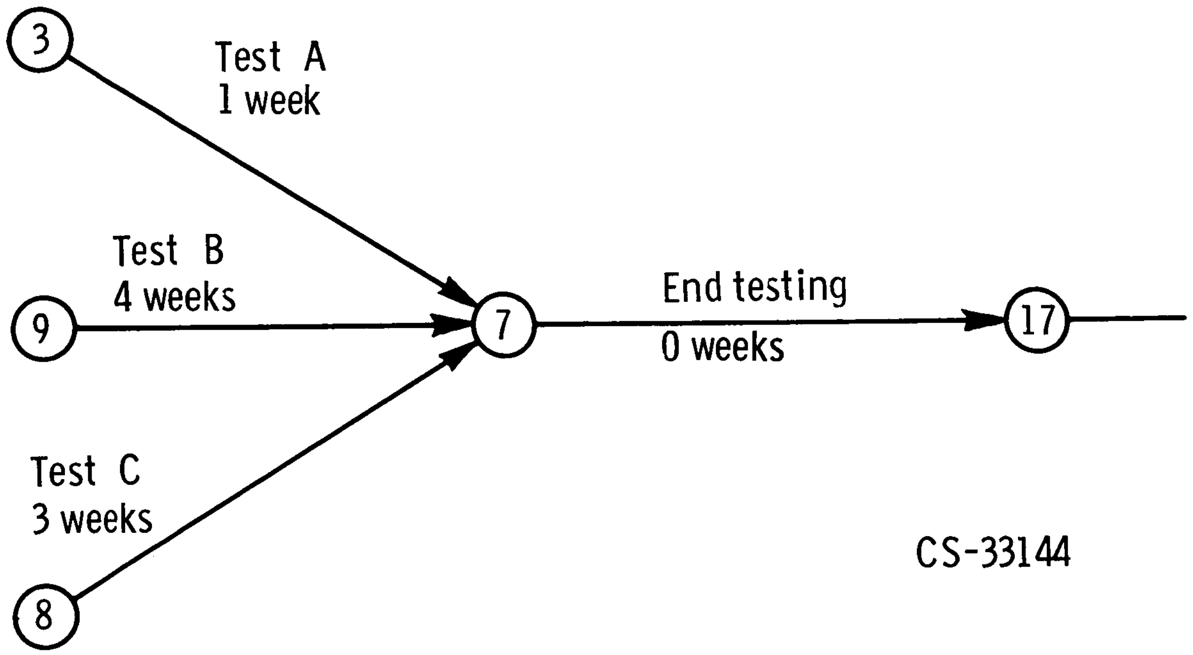


Figure 6. - Dummy activity.

TOPOLOGICAL ORDERING USING THE PUSHDOWN TECHNIQUE

by

Hans Bremer
NASA Goddard Space Flight Center
Greenbelt, Maryland

In the analysis of all PERT networks, the problem reduces itself to one of computing T_E, T_L, and the resultant slack for each activity. To accomplish this, one has to proceed in some orderly fashion if one is going to have a computer do this. Now, normally when we look at a PERT chart, we are looking at the data in an entirely different fashion than the computer does. That is, we can perceive data in one, two or even in three dimensions and proceed to act upon this data. The computer, however, can accept data only in one dimension, so that we have to break a PERT chart down into a deck of cards, one activity per card and feed this deck serially to the computer. Thus, we are presenting the computer with data in one dimension. Normally we order such a deck either numerically or alphabetically, but this order relationship, generally has no bearing on the network because the labelling of the network is arbitrary. Therefore, in order to make the computer see the network list in an organized fashion, such as we see it when we look at it in two dimensions, we have to order the activity list by topological order. Once we order the list topologically, it is quite easy to compute T_E, T_L, and slack. This, then is why we worry about topological ordering.

Now, topological ordering is actually a type of rank ordering, which, while it is similar to numerical ordering or alphabetical ordering, is different in the sense that the ordering criteria is implied by the list itself and is not imposed by some external criteria as is the case in numerical or alphabetical ordering. For instance, we all know what is meant by numerical ordering. It is a common criteria which we all apply to a list of numbers.

If we were going to order a list, containing employee's serial numbers, we would simply match the employee's numbers against all the integers and arrange them in the order of the integers.

Notice here that the employee's numbers are single element entities, as opposed to activities which are double element entities. Furthermore, in a list of activities, it is understood that the first element points to the second, and that the second element may or may not represent the first element of some other activity on the list.

Topological ordering, thus involves only lists made up of two elements per entity and where one element points to the other, as is the case of the activity list. A reasonable definition of topological ordering can be given as follows; a list is said to be in topological order if every element in the list preceeds all other elements that it points to in the list.

As a small example consider three arrows labelled x to y, y to a and a to b. In a list they might appear as

y to a

x to y

a to b

Arranged in topological order this list would appear as

x to y

y to a

a to b

It is the fact that the ordering is implicit in the list that makes topological ordering a relatively difficult sort problem. The reason being that we first have to scan the list to find the ordering criteria and then we have to apply it, whereas in ordinary numerical or alphabetical sorting, the criteria is known beforehand. The pushdown list technique seemed to be the easiest to implement using the Fortran language.

An understanding of how the pushdown technique was used to perform the topological ordering necessary to compute T_E and T_L will provide future users of the NASA-FORTRAN PERT Program with the ability to modify this particular subroutine in the program, if their needs so dictate.

The pushdown technique is a very powerful device. It is used extensively by people who write compilers. It is also known to the people in business data processing world as the "Last in-first out" file. In one line of machines at least, there exists actual hardware instructions to implement the pushdown technique. This is the Burroughs B-5000. IBM and SHARE have also considered the technique to be of sufficient merit as to include an instruction set for it in the new program language that is being proposed for IBM's 360-Line.

Basically, the pushdown list can be thought of quite conveniently as one of those holders of plates that one sees in cafeterias, where as we pile more plates on a pile, the top one is always level with the serving table. In our use of the pushdown technique we have 2 pushdown lists - one which we actually call the pushdown list and the other the path list.

Conceptually, the technique has the effect tracing out all of the paths in a network. At first blush this would appear to present a formidable amount of work, both in terms of the amount of bookkeeping instructions required and the amount of machine time required. But, this is precisely the power of the technique, there is very little bookkeeping involved, and consequently the amount of machine time used is kept minimal.

To illustrate how the technique works let us take the example in the NASA-PERT Computer Systems' Manual, dated June 1963, see Figure 1.

This provides us with a list of activities sorted by successor within predecessor order and will be called the activity list, as in Figure 2.

All in all we will need four lists: The Activity List, the Pushdown List, the Path List and an Order List.

The first step consists of taking the activities associated with a Start event and putting these activities on the Pushdown List. Starts and ends can either be identified as input data or searched for and flagged. Thus we put the activities AC, AD, BC and BE on the Pushdown List. See Figure 3. Having just placed some items on the Pushdown List we go and take the first activity off of this list and place it on the Path List. Thus activity AC appears on the Path List, as in Figure 4.

Having just placed an activity on a Path List, we first check to see if the associated successor is a terminal event. In this case it is not, consequently, the process continues. This consists of examining the successor event and using it as a search argument for pulling off of the activity list all those activities whose predecessors match the search argument. In this case the activities CD and CE match this criterion and therefore are placed on the Pushdown List as in Figure 5.

Again, having just put some items into the Pushdown List we proceed to immediately remove the top one and add this to the Path List. See Figure 6. Having just added an item to the Path List we again check to make sure that this isn't the end of the path by comparing the successor event with the terminal events. Since this is not a terminal event, we continue as before. That is, we take the successor event and use it as the search argument for searching the activity list for all activities with matching predecessors. In this case DE and DF, and we place these on the Pushdown List, as in Figure 7. From here we immediately place the activity DE, which was the last item entered into the Pushdown List onto the Path List. Hence the activity DE is also the first out of the Pushdown List. This then is also the reason for the other name for the Pushdown List, namely, last in-first out, or LIFO.

By now the process is rather obvious. That is, we check to see if the new item on the Path List is an end item. If not we proceed to pull new items off of the Activity List and onto the Pushdown List. Thus EG goes to the Pushdown List and from there to the Path List as in Figure 9 and Figure 10. This time the successor event is a terminal event so the algorithm comes to a branch point, for it is at this point that we can begin to rank some of the events in the events list, namely those that appear on the Path List. We rank these events as shown in Figure 10B.
After we have noted the event ranking we remove the last item from the Path List, that is the activity EG and proceed to the next item, namely DE and again using the successor event as the search argument, that is the event E, we now begin to search the Pushdown List for activities to match. Note the difference here, as long as we are adding activities to the Path List, we search the Pushdown List for follow-on activities to make up the path.

At this point in time the activity DE has no follow-on activities in the Pushdown List, hence it too is removed from the Path List, as in Figure 12 and we proceed to the next activity on the Path List. The activity, namely CD does have a follow-on activity on the Pushdown List and this is the activity DF. This activity is now removed from the Pushdown List and is added to the Path List. We now proceed as we did earlier, that is having added something to the Path List, we examine it to see if it has a terminal event which in this case it does. Thus we pause again long enough to compute the ranking, as in Figure 13.

Having computed rank order we remove DF from the Path List and proceed to match successor events on the Path List with Predecessor events on the Pushdown List and removing activities from the Path List until a match occurs. Thus, eventually AC and CE match this fashion and so CE is finally added to the Path List (Figure 16), and the point at which this happens is then the signal to move in a forward direction as we have already outlined.

At some point the Path List will be exhausted and at this juncture, we look to the Pushdown List to see if there are still one or more items left in the list. If none are left the process is terminated. If some are left, we continue the process by taking the first item in the Pushdown List and placing it in the Path List, and proceed from there as outlined above.

At some point we will have traced out all of the paths and consequently we will have established a topological ordering within the events. If we wanted to we could also rank the activities now.

This would be done by taking the corresponding rank number for the event that matches the successor event of the activity and assigning the rank number to the activity; and doing this for all activities. Then it is simply a matter of performing a numerical sort on this number to obtain the required topological ordering of activities.

It should be pointed out that we have done all of this work, of arranging the activities in topological ordering for the sole purpose of computing T_E and T_L. This is the process followed in most of the available PERT programs.

In the current NASA-FORTRAN Program, however, we have departed from this approach, with the result that we have been able to eliminate the sort step just described, and thus save some additional machine time. Remember the thing that we are really after, after all is T_E (and T_L), and it can actually be computed at the time that we rank the events on the event list. Thus, instead of assigning rank value to the events, we merely assign values in terms of the number of weeks that will elapse before the current event can take place with reference to some base date. The events are thus ranked by order, but even more than that, they are ranked by time.

Some additional comments should be made about the actual implementation. The major share of the implementation was done by Liz Ryan and Ross Bainbridge of the Lewis Research Center.

The following methods were used by them to modify the basic topological procedure and to increase its efficiency.

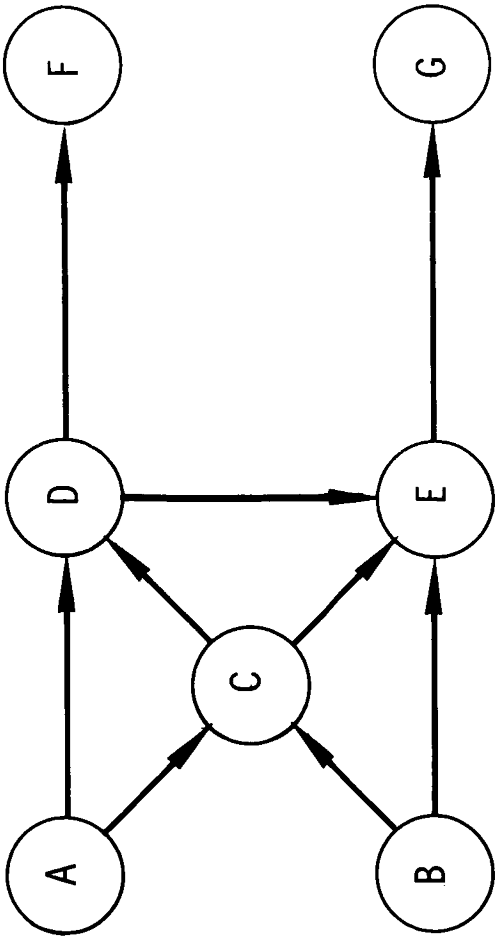
1. Sequential numbering of the events eliminated events table searching. This sequential numbering also eliminated the necessity for retaining internally the actual event numbers.

2. Retention of the activity position counter in an events table eliminates any activity table searches. This in turn eliminates the necessity for retaining internally the predecessor event of an activity.

3. Experimentation with the T_E and T_L calculations resulted in the determination that if T_E was less than, or if T_L was greater than the previous value computed than the path analysis could be terminated, for this path. This innovation resulted in a considerable savings in actual internal computing times.

Mr. Don Hiller of the Control Data Corporation, Washington Data Center, has informed me that they have successfully run the Fortran version of NASA PERT on the CDC 3600.

I have been told by Mr. Lou Thomas of UNIVAC, Federal Government Marketing, out of Washington, D. C., that he has successfully run the Fortran PERT on Univac's 1107.


A
B
C
D
E
F
G

ACTIVITY LIST

AC
AD
BC
BE
CD
CE
DE
DF
EG

2

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	AC	
AD	AD	
BC	BC	
BE	BE	
CD		
CE		
DE		
DF		
EG		

3

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	AD	AC
AD	BC	
BC	BE	
BE		
CD		
CE		
DE		
DF		
EG		

4

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	CD	AC
AD	CE	
BC	AD	
BE	BC	
CD	BE	
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	CE	CD
AD	AD	AC
BC	BC	
BE	BE	
CD		
CE		
DE		
DF		
EG		

6

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	DE	CD
AD	DF	AC
BC	CE	
BE	AD	
CD	BC	
CE	BE	
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	DF	DE
AD	CE	CD
BC	AD	AC
BE	BC	
CD	BE	
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	EG	DE
AD	DF	CD
BC	CE	AC
BE	AD	
CD	BC	
CE	BE	
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AD	DF	EG
BC	CE	DE
BE	AD	CD
CD	BC	AC
CE	BE	
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST	EVENT LIST	
AC	DF	EG	A	1
AD	CE	DE	B	1
BC	AD	CD	C	2
BE	BC	AC	D	3
CD	BE		E	4
CE			F	
DE			G	5
DF				
EG				

IOB

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	DF	DE
AD	CE	CD
BC	AD	AC
BE	BC	
CD	BE	
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	DF	CD
AD	CE	AC
BC	AD	
BE	BC	
CD	BE	
CE		
DE		
DF		
EG		

12

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST	EVENT LIST	
AC	CE	DF	A	1
AD	AD	CD	B	1
BC	BC	AC	C	2
BE	BE		D	3
CD			E	4
CE			F	4
DE			G	5
DF				
EG				

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	CE	CD
AD	AD	AC
BC	BC	
BE	BE	
CD		
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	CE	AC
AD	AD	
BC	BC	
BE	BE	
CD		
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	AD	CE
AD	BC	AC
BC	BE	
BE		
CD		
CE		
DE		
DF		
EG		

ACTIVITY LIST	PUSHDOWN LIST	PATH LIST
AC	EG	CE
AD	AD	AC
BC	BC	
BE	BE	
CD		
CE		
DE		
DF		
EG		

THE NASA PERT 1410 SYSTEM

by

Al Sheffield
NASA John F. Kennedy Space Center
Cocoa Beach, Florida

INTRODUCTION

My presentation today is on the NASA PERT Time System as handled by the IBM 1410 Computer.

We at Cape Kennedy started about 1 year ago with 1 Pert network. Now we have grown to about 18 running networks with 13 more on the Planning Board to come in shortly.

We are presently producing runs on the transporters, the construction of LUT, Complex 37, Complex 34, and the Cryogenic Test Building. The 13 networks now in the planning stage will cover the activation of checkout facilities at the new NASA MILA installation.

Our 1410 Pert system is completely tape oriented and uses the new GE-415 as its I/O Slave. We have had the GE-415 for about 1½ months now. It was obtained to replace the 1401 we were previously using.

Our 1410 program utilizes 40K positions of memory, 7 tape drives, and two channels. The GE-415 is used for card-to-tape, and tape-to-printer functions.

SYSTEM DESIGN OVERALL PROBLEM

The major problem was to design a system that could handle Pert Time as prescribed by NASA Headquarters for the 7090, for the 1410 having less core memory storage ability and fewer tape drives.

The necessary memory for containing entire tables in core was not available. This would have increased operational speed by use of address modification for direct addressing.

This was overcome by use of work tapes containing the tables, calling in by segments, when and as needed. This, however, necessitates many READ-WRITE-REWIND operations, which causes considerable loss of time. At present this is a very slow running system.

Our system is arranged into 3 separate sections, each one linked program-wise to the next. The system operates from one section to the other without operator intervention.

EDIT

We at Cape Kennedy felt that because of the slow processing of our Pert system. We should have a real strong edit program that would assure a successful Pert Run.

We wanted an edit that would catch every error possible that could cause a Pert Run to go haywire.

Things to be handled were:

1. Card edit of every field.

2. Dangling predecessor.

3. Dangling successor.

4. Constraints.

5. End-Events that are not legitimate.

Our Pert ADP Staff at Cape Kennedy is very small. It is as small as two people, and these two people have other areas of responsibility besides Pert. Therefore, our ADP Staff does no correcting of Pert networks at all.

We run our edit and any errors detected are sent back to the user for correction. We monitor the Run itself, but not the correction of the networks. This edit is designed to run independent from the main Pert Programs and allows for customer intervention.

This edit is divided into 3 phases:

Phase I - Performs a complete card inspection, Field by Field.

A. Valid code

B. Valid predecessor No.

C. Valid successor No.

D. Valid date

E. Valid activity time.

F. Completion date FON 3 cards, which are completed items.

Phase II - Checks for the following items:

A. Dangling predecessor

B. Dangling successor

C. Points of constraint.

Phase III - Produces an 80/80 listing of the entire network in succ. sequence as an aid for user to check out his errors.

Every message on an error listing should be checked out by user, whether he be a Pert user or the user of some other computer system. Therefore, only valid rejects should be placed on such a listing, for user checking.

This edit program, with this thought in mind, has eliminated the unnecessary printing out of ending events as dangling successors. This is accomplished by keeping a table of all ending events in the network. When a dangling successor condition is detected by the program, it checks against this table to see if it is an Ending-Event. If a match occurs, the item is not printed out. If no match occurs, then it is printed out. This has eliminated some unnecessary checking that user previously was having to perform.

Another error message that was causing considerable concern was that of constraints.

A point of constraint is where more than one activity leads into a given event. Result is subsequent activities should not be completed until all legs leading into this starting event are completed. However, it is possible for activities beyond this point to be started, yet prior activities not being completed. This program makes this check and by checking all activities leading from this event for Code 3's.

As a result, constraints printed out on error list are only legitimate constraints that would cause Time Schedule in processing to get out of proportion and run to blow.

This edit has accomplished two things for us at Cape Kennedy:

1. It has cut down on rerun time.

2. By eliminating erroneous error messages and increasing the turn-around time in furnishing user with product.

SUMMARY

Our 1410 Pert system will not handle the new Pert "C" changes. This, along with the slowness of the system brought about a re-write of our Pert, which our ADP Staff is undertaking now.

Once REWRITE is completed, we expect Run Time to be decreased by at least 50 percent of its present time.

Our REWRITE will be in Auto Coder. Once this is accomplished, we intend to undertake the translation of our Rewrite to COBOL.

USE OF THE SC-4020 FOR NASA PERT GRAPHIC OUTPUT
(The PAGE System)

by

Jerry Huffman
NASA George C. Marshall Space Flight Center
Huntsville, Alabama

INTRODUCTION

My portion of the presentation is concerned with graphical PERT output, or the condensation of vast amounts of data into meaningful form for management. With the advent of computers, management has been flooded with vast amounts of data. Whereas before, too little data was available for valid management decisions, now too much data for adequate assimilation is available.

Prior to the development of the system I will discuss, OMSF Schedule and Review Procedures (SARP) charting was done manually. SARP charts are a reporting media by which NASA Field Centers report key milestone information from PERT to Headquarters just as PERT is a system by which industry reports to Field Centers. Figure 1 is an example of a SARP chart. The volume of SARP charts had increased to the point that it had become burdensome to prepare them manually.

APPROACH

Utilizing the IBM 1410 computer and the General Dynamics SC4020 Recorder, the PERT Automatic Graphical Extensions, or "PAGE", was designed as one answer to this program. PAGE selects milestone data from a normal NASA PERT 7094 run and displays it in a meaningful format on graphical charts.

The SC4020, a cathode ray tube device, has the ability to plot curves, draw axes, determine vectors, and in a fraction of a second, type highly accurate graphs and charts.

ADVANTAGES DERIVED BY USE OF SC4020 GRAPHICS

Several benefits can be readily obtained by use of the PAGE system for automating management charts. Some of these are:

1. Interpretation of graphic schedules can be quickly made.
2. Various parts of the same PERT network can be automatically grouped and graphically compared.

3. Data from parts of different PERT networks can be integrated and summarized for quick comprehension.

4. Trends can be readily projected and examined.

5. The marriage of NASA PERT and PAGE can provide the program and project manager and other management personnel with a picture that can be grasped rapidly so that rescheduling and direction can be quickly provided.

6. Compared with manual chart illustration techniques and conventional photographic processes, monetary savings can be expected.

7. With a relatively simple method of manual coding and input, the PAGE system can serve many applications without the necessity of developing a concurrent PERT system.

INPUTS

The "A-4" binary output tape from a 7094 NASA/PERT run is the input to the PAGE system. Milestone symbols in columns 2 and 3 are used to identify the arrangement and selection of data. Controls have been established that permit page-grouping of data by column 2 and/or column 3 individually, or by both at the same time, compatible with NASA/PERT Mod. "C".

PROCESSING

PAGE 1410 processing is accomplished in three phases. In Phase I, selection for a detailed network is done by consideration of activities past due, activities due during the next period, latest allowable activities of the current period, and the latest allowable activities of the next period. Selection for a summary network can also be made by predetermined selected level or milestone symbols and by organization designators.

Phase II involves sorting of the selected entries by M/S, date, slack, or selection of any of five fields that are in the input record.

In Phase III, the sorted data is formatted into necessary plotting instructions, which permit the SC4020 to prepare 35 mm microfilm and graphical charts displaying selected data. Scales and titles are added during this run.

OUTPUTS

Figure 1. Master Schedule

Figure 2. T_L This Period

Figure 3. T_L Past Due

Figure 4. T_E Next Period

Figure 5. T_L Next Period

Figure 6. $T_L - T_E$ (Slack)

This is only a partial list of examples. Other output groupings are available.

The following information (1-9) appears at the top of each chart, is given by three control cards, and is inserted on a master record by Phase I:

1. Title for each master record.
2. Contractor and contractor number.
3. Schedule number.
4. Project number.
5. Level.
6. Original schedule date.
7. Latest change date.
8. Schedule responsibility.
9. Status responsibility.
10. Network description.
11. Network identification.
12. Date of the report.

Items 10-12 listed above and the detail activity information for each entry on a chart is taken from the PAGE input tape which has all the basic calculations from a PERT 7094 run. The additional information supplied by this source is as follows:

1. Expected completion date.
2. Latest allowable completion date.
3. Scheduled completion date.
4. Actual completion date.
5. Successor event number.
6. Predecessor event number.
7. Activity time.
8. Activity description.
9. Organization.
10. Slack.
11. Resources.
12. Master schedule.

The vertical broken line on the output chart denotes "time now", the date of the report. Arrows may be used to represent any date in the above list except "Latest allowable completion date". Arrows are used in the example in this presentation to designate "Expected completion dates". "E"s and "L"s on the chart entries are used to denote expected completion dates (unless arrow used for this purpose) and latest allowable completion dates, respectively.

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP __________

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN
SATURN V LAUNCH VEHICLE PR
MASTER SCH

CONTR' SCHED PROJ.HD

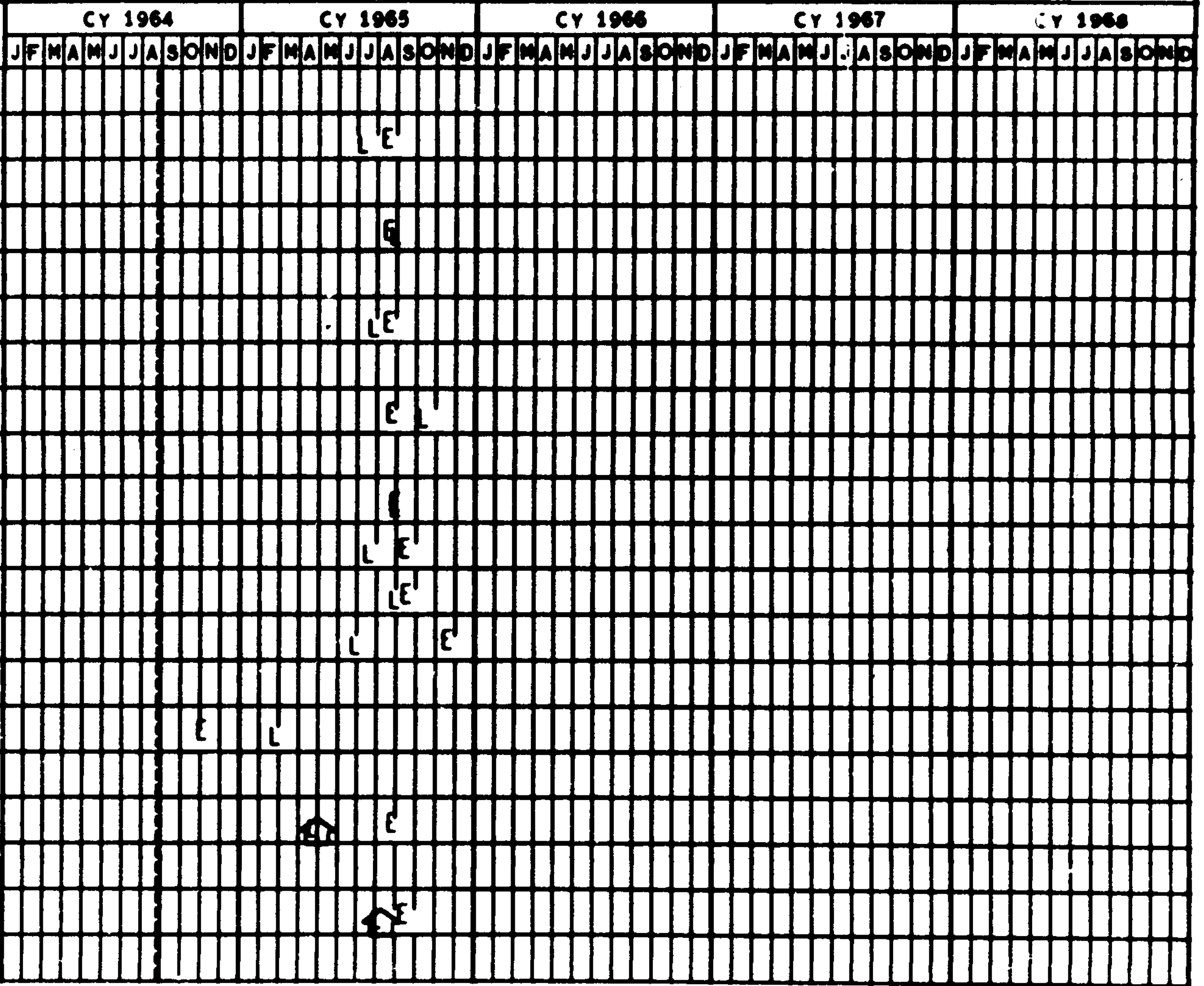
1 LEVEL

ORIG SCHED APRVL __________

LAST SCHED CHG __________

STATUS AS OF 8/27/64

	MILESTONES
01	OBTAIN&DELIVER SII STAGE
02	DDAS EQUIP
03	OBTAIN&DELIVER IU
04	UMBILICAL CONN ASSEMBLY
05	OBTAIN&DEL SIVB UMBIL
06	CARRIER
07	OBTAIN&DELIVER GROUND
08	TELEMETRY STATION
09	FAB&DELIVER THEODOLITE
10	HOLDING FIXTURE
11	OBTAIN&DELIVER S4B ESE
12	OBT AIN&DELIVER IU ESE
13	OBTAIN&DEL SIVB MOCKUP
14	COMPLETE MODIF/BLDG
15	4708/SAT-V SDF
16	START SIC STG
17	OPERATIONAL CHECKOUT
18	START SII STAGE
19	OPERATIONAL CHECKOUT
20	



NOTES

NASA FORM 796 DEC. 62

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN
SATURN V LAUNCH VEHICLE PR
TL THIS PER PROJ.

CONTR' SCHED

1
LEVEL

ORIG SCHED APRVL

LAST SCHED CHG

STATUS AS OF 8/27/64

	MILESTONES	CY 1964												CY 1965												CY 1966												CY 1967												CY 1968											
		J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D
01	PREP&ISSUE FINAL EQUIP																																																												
02	L/O ARRGT SIVB								L	E																																																			
03	START DES DRAFTING IC																																																												
04	CABLE DRAWING/SIC								L	E																																																			
05	PREPARE&ISSUE REQUEST/SII																																																												
06	EQUIPMENT DEL								L	E																																																			
07	PREP&DELIVER FAC CBL DES																																																												
08	INFO/PWR SUPP								L			E																																																
09																																																													
10																																																													
11																																																													
12																																																													
13																																																													
14																																																													
15																																																													
16																																																													
17																																																													
18																																																													
19																																																													
20																																																													

NOTES

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN
SATURN V LAUNCH VEHICLE PR
TL PAST DUE

CONTR'

SCHED

PROJ.

1 LEVEL

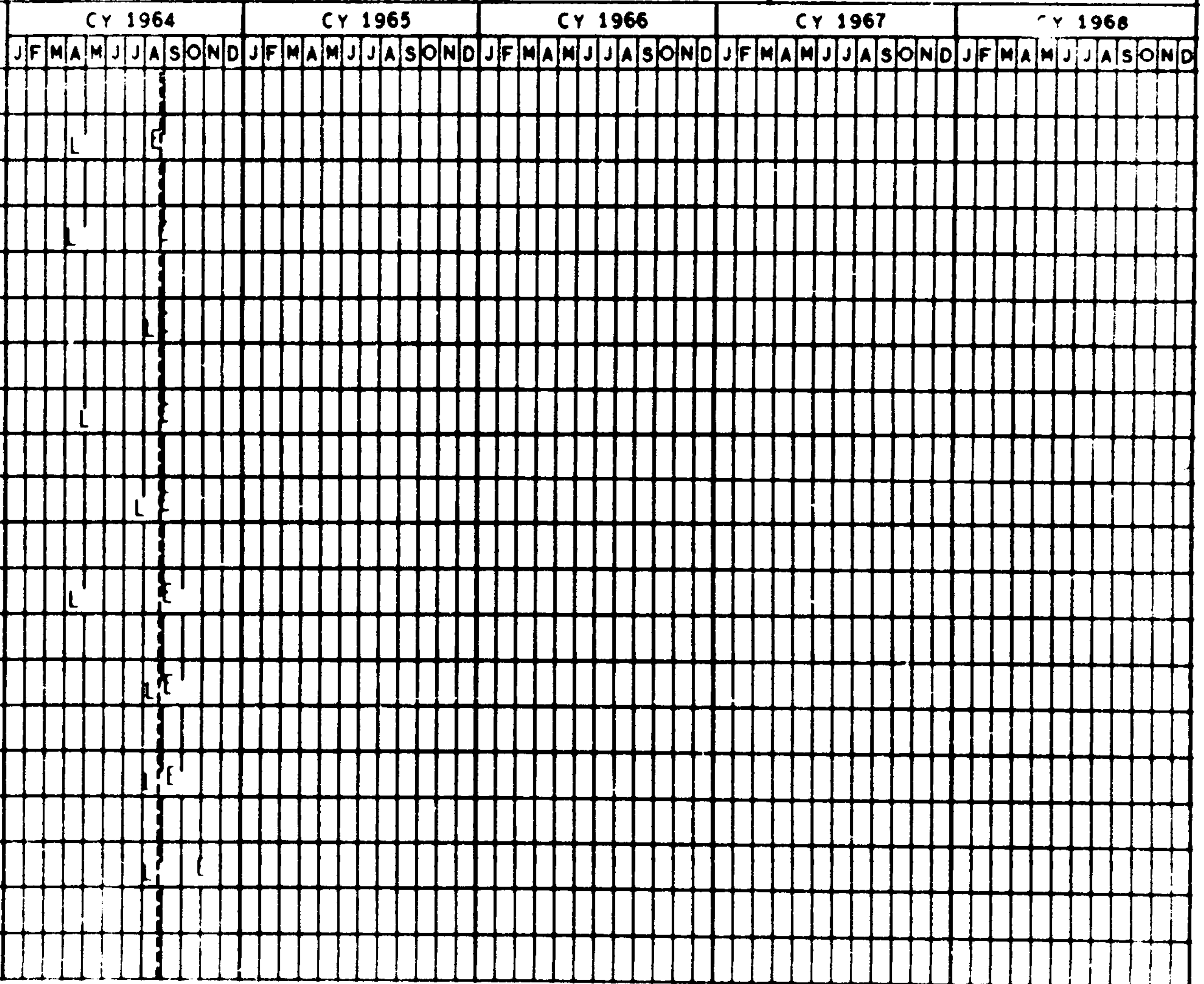
ORIG SCHED APRVL

LAST SCHED CHG

STATUS AS OF 8/27/64

MILESTONES

01 ISSUE FINAL SAT-V
02 EQUIPMENT ARPGMT DRAWING
03 PREP&ISSUE FINAL EQUIP
04 LAYOUT ARRGMT SIVB
05 PREP&ISSUE FINAL EQUIP
06 L/O ARRGT SIVB
07 ISSUE MEMO REQUEST FOR
08 SIC ESE DELIVERY
09 PREPARE&ISSUE FINAL SII
10 EQUIP ARRGT DWG
11 PREPARE MEMO REQUEST/ESE
12 DELIVERY S4B
13 START DES DRAFTING IC
14 CABLE DRAWING/SIC
15 PREPARE&ISSUE REQUEST/SII
16 EQUIPMENT DEL
17 PREP&DELIVER FAC CBL DES
18 INFO/PWR SUPP
19
20



NOTES

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN
SATURN V LAUNCH VEHICLE PR
TE NEXT PER

1 LEVEL

PROJ.OO

CONTR'

SCHED

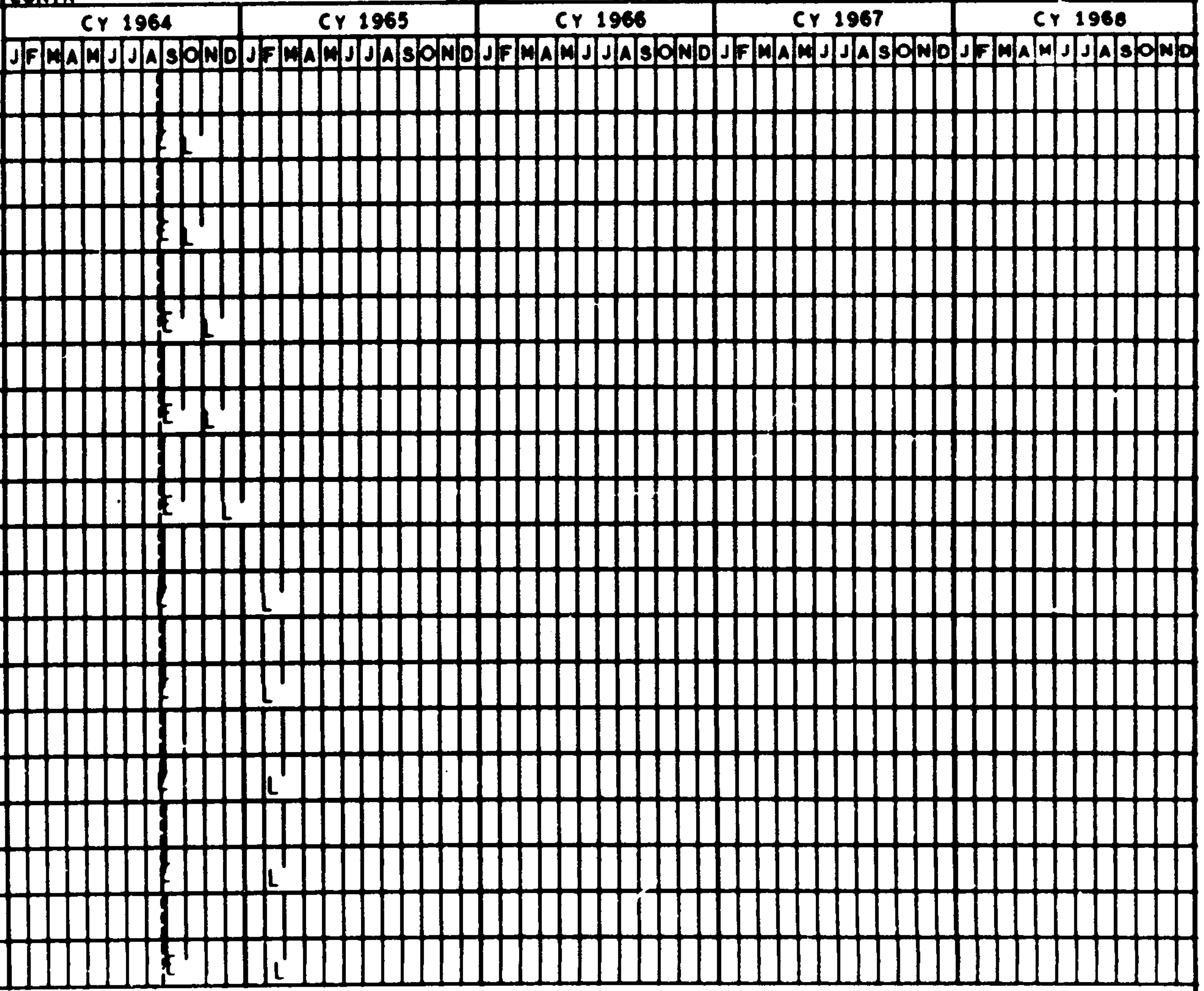
ORIG SCHED APRVL

LAST SCHED CHG

STATUS AS OF 8/27/64

MILESTONES

01 ADD SIC DDAS/PRELIMINARY
02 IC CABLE DRAWING
03 ADD SIC SIMUL/PRELIMINARY
04 IC CABLE DWG
05 RECEIVE POWER SUPPLY&START
06 FAB/RACKS&PNLS
07 VENDOR DELIVER POWER
08 SUPPLY HARDWARE/SDF
09 OBTAIN MATERIAL FOR
10 CABLE&CONNECTOR/P.S.
11 ADD RACS/CABLE REVIEW
12 DRAFT DRAWING
13 ADD DDAS/CBL REVIEW DRAFT
14 DRAWING
15 ADD RACS EQUIP/CABLE
16 REVIEW DRAWING SIVB
17 ADD DDAS EQUIP/CABLE
18 REVIEW DRAWING SIVB
19 RECEIVE AND START C/O
20 POWER SUPPLY HDWE



NOTES

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP ______

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN
SATURN V LAUNCH VEHICLE PR
TL NEXT PER

CONTR'

SCHED

PROJ.

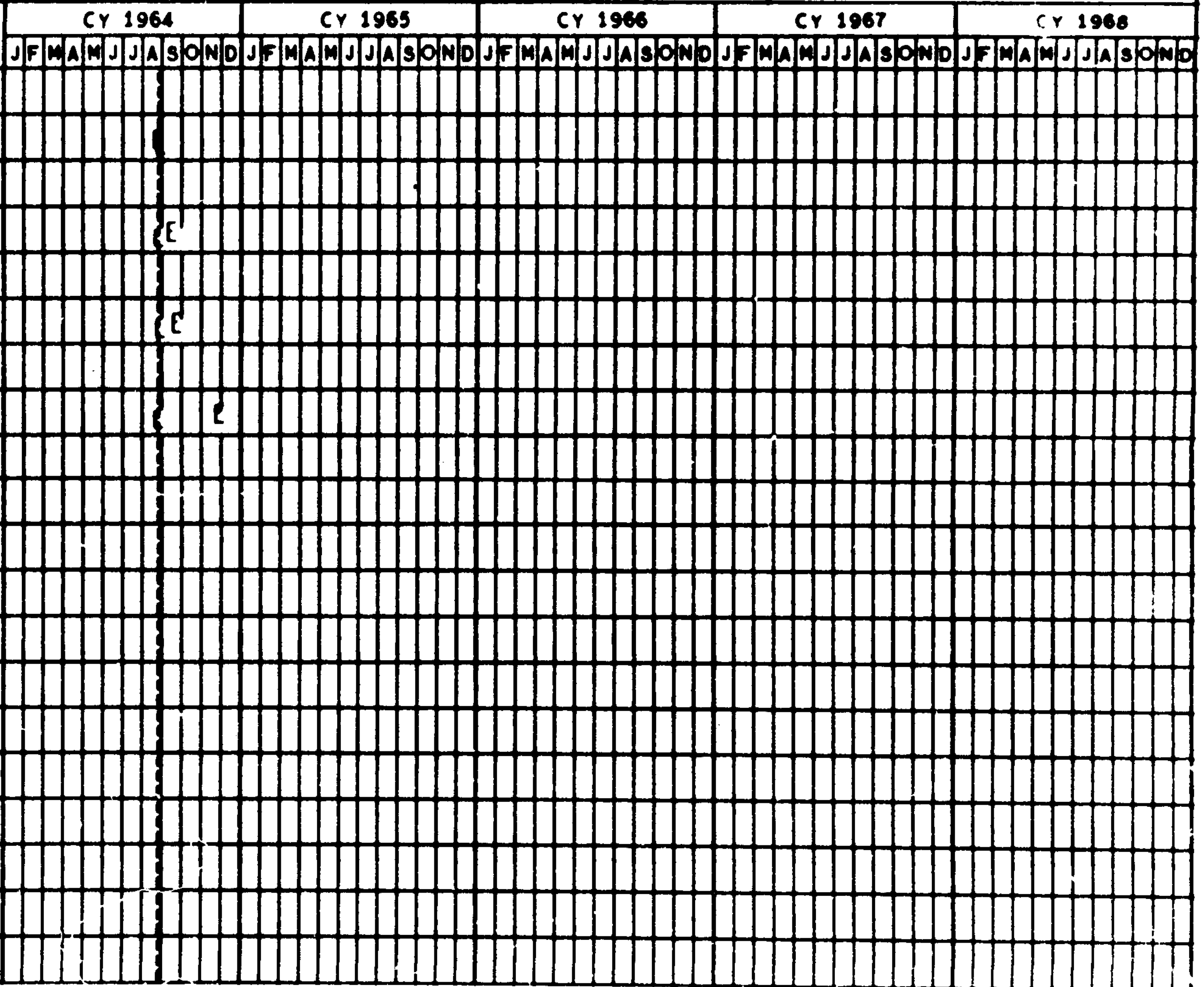
1 LEVEL

ORIG SCHED APRVL ______

LAST SCHED CHG ______

STATUS AS OF 8/27/64

	MILESTONES
01	PREPARE AND DELIVER CBL
02	DES INFO/SYS INTEG
03	PREP&ISSUE REQUEST FOR
04	DELIVERY IV ESE
05	PREPARE&DELIVER CABLE DES
06	INFO C.D. CLOCK
07	PREPARE AND DELIVER CBL
08	DES INFO/SIC ESE
09	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	



NOTES

SCHED RESP DR. WERNHER VON BRAUN

STATUS RESP

MANNED SPACE FLIGHT SCHEDULE

SYSTEMS DEVELOPMENT FACILITY PLAN

SATURN V LAUNCH VEHICLE PR

TL LESS TE

LEVEL 1

PROJ.OO

ORIG SCHED APPVL

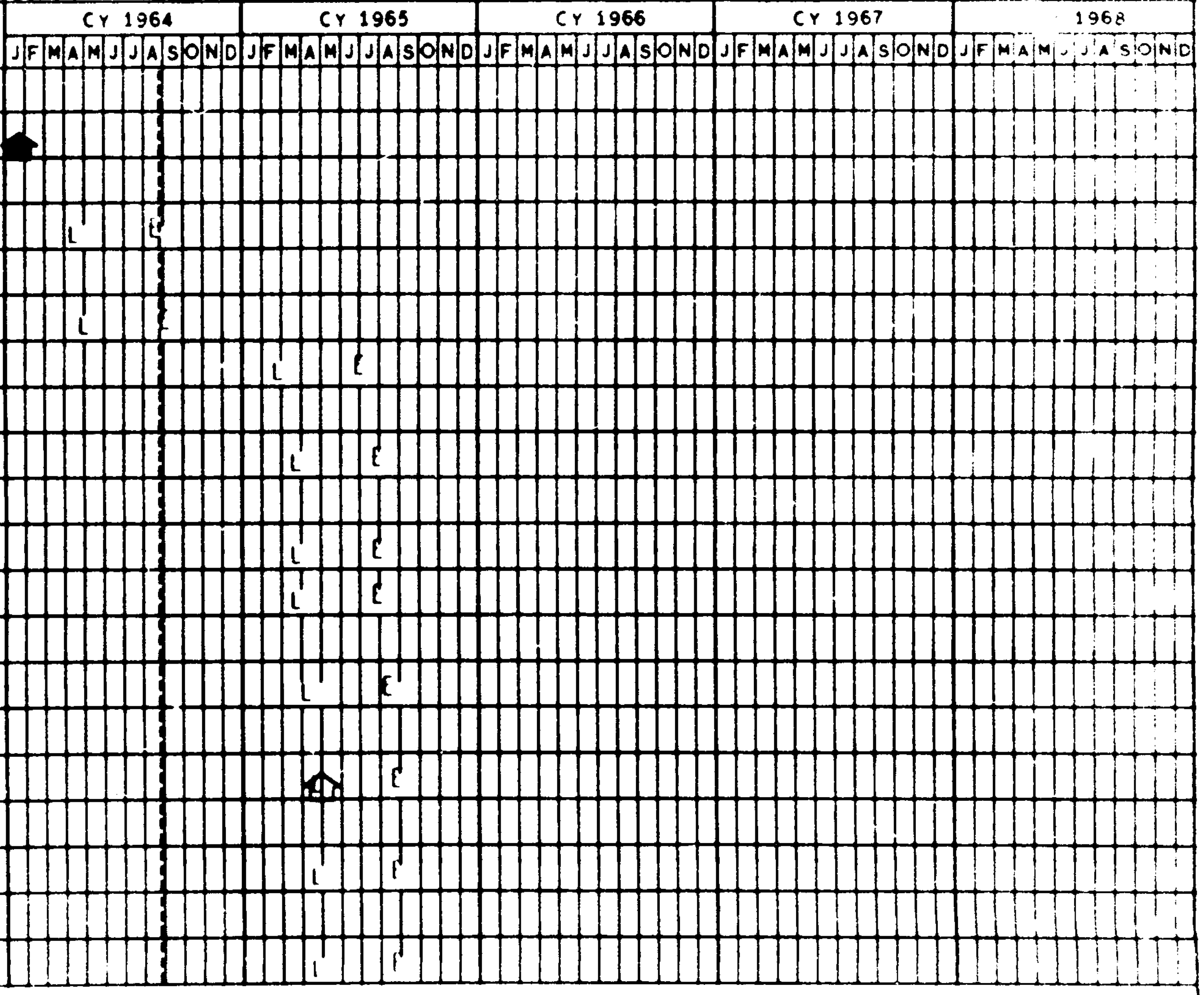
LAST SCHED CHG

STATUS AS OF 8/27/64

CONTR' SCHED

MILESTONE

01 ISSUE PRELIMINARY
02 EQUIPMENTS AND LAYOUT
03 ISSUE FINAL SAT-V
04 EQUIPMENT ARRGMT DRAWING
05 ISSUE MEMO REQUEST FOR
06 SIC ESE DELIVERY
07 OBTAIN&DELIVER SIC ESE
08 SIC ESE&GSE INSTALL/GETS
09 CHECK
10 PREPARE/START INSTALL C/O
11 SIC EQUIP
12 INSTALL SIC ESE
13 PREPARE TO PERFORM
14 INTEGRATED C/O/GSE&SIC
15 START SIC STG
16 OPERATIONAL CHECKOUT
17 INTEG GSE&SIC&MAB C/O
18 CONSTRNT/SIC OPR C/O
19 PERFORM INTEGRATED C/O
20 GSE&SIC&MAB



NOTES

PLANS FOR MECHANIZING CONTRACTOR FINANCIAL MANAGEMENT REPORT

by

K. C. Mulcahy
NASA Headquarters
Washington, D. C.

We have recently completed a revision of the NASA Form 533, Contractor Financial Management Report.

This is an illustration of the format. It contains: costs incurred, forecasts of costs to contract completion, estimated final costs, a distribution of current contract value, estimated completion date, and unfilled orders outstanding as of the report date.

The reporting categories are: subdivisions of the work under contract, elements of cost for subdivisions - that is, direct labor, direct material, overhead, subcontracts and so forth, direct labor hours for each direct labor element of cost, and unfilled orders outstanding forecasts. It is a monthly report - but the forecasts are mandatory only quarterly.

The NASA Form 533 was established after a series of tests and evaluations as the basic report of contractor financial management data to NASA project managers. After approval by the Bureau of the Budget, it was placed in NASA-wide operation in February, 1962. It is more comprehensive than forms used by DoD for the similar purpose of obtaining contractor financial management data, DD Forms 1097 and 1177. The NASA Form 533 is important because something like 90% of NASA's costs are incurred by contractors and NASA's budget is at an annual rate in excess of five billion dollars. When the initial approval expired, NASA recommended some simplifications - reduction in the report columns, eliminations of redundant data from previous reports, and clear definitions.

Since the NASA Form 533 is the basic input of financial data from contractors to the NASA financial management system, the coordination of revisions within NASA required careful consideration. Under the Federal Reports Act, the revisions were submitted to the Bureau of the Budget for approval. A panel of the Advisory Council on Federal Reports was convened to advise the Bureau. This process takes time but both Industry and NASA have reached clearer understandings because of it.

In general, NASA's need for the report is clearly recognized. Much of the discussion by the panel of the Advisory Council on Federal Reports centered on the revision which gave formal recognition in the report instructions of direct labor hours reporting. Since most major contractors have been furnishing direct labor hour data to NASA since inception of the report, and NASA has been utilizing the data extensively, the Bureau approved this formal change.

The other major point weighed by the Bureau was the detail forecasting by quarter. Forecasting is the name of the game, however, and detail is necessary because many people must make related decisions on common bases. Forecasts are expected to fluctuate and to differ from actuals. If it were not so, we would be in a fixed-price rather than a cost reimbursement environment.

NASA's general management has recognized an urgent and immediate need for cost-based planning to resolve the problems of operating within FY 1965 anticipated appropriations. Further, the longer range objectives in the management of the agencys' resources are being based on the accrued costs of cost reimbursement contractors. The NASA Form 533 is essential to both of these aspects of NASA's overall program operating plans. These plans are basic to NASA's budgetary control. The NASA Form 533 is ideally suited for the role of cornerstone in NASA's budget system.

The Bureau of the Budget has levied upon NASA a requirement to report on the uses of the forecasts, the changes made to them, and the means used to improve them. Subject only to this continuing review, they have approved NASA's use of the Form 533 through May 31, 1968.

Coordination within NASA, discussion with the panel of the Advisory Council on Federal Reports, and formal approval by the Bureau of the Budget each resulted in some minor changes. In this climate of change, we have held back on agency-wide standards for mechanizing the NASA Form 533. We were fairly certain some changes would be made, but because of the necessity to negotiate uncertain of the exact changes.

A real need exists for a mechanized analysis of the NASA Form 533. The need is timeliness. Financial data comes in detail which must be developed and analyzed or it cannot be used by NASA project managers in decision making. Only when the time span between receipt and decision is short can there be real assurance that action in response to decision is the effective cause of change. The projects are too large, too complex, and too developmental in nature to wait on hand sorted, checked, and computed financial data.

In order to support analysis of the results of alternative courses of action, and the effects of trade offs, details must be available. The details on financial management come from the NASA Form 533, and only mechanization makes these details available in time. Other advantages-increased accuracy, better controls, more flexibility - will result from having mechanized files of NASA Form 533 available but the need for speed without sacrifice of in depth analysis, in the day-to-day decisions of project managers is the primary reason for mechanizing the Contractor Financial Management Report.

Although an agency-wide program has not been established, NASA centers have mechanized the existing NASA Form 533 to some extent. The need has been too great to wait on formal approval of revisions. The current status of efforts that may be characterized as not impacting on contractors is:

A program in use that compares the current month's report with its predecessor and computes variances. Variances that exceed a fixed criterion are printed out for explanations. The explanations are the basis of analytical reports to project management for action.

A program in final development that will compute variances similarly, but print all variances, make rate computations - that is, dollars per hour, overhead, and burden, and develop ratios on labor mix - that is, high priced labor to low priced labor.

Also, a program is in use that ensures contractors' reports are compatible with in-house accounts and records, and therefore available for current reporting and analytical cost estimating of total costs.

Each of these programs well serves the center management that adopted it for use. They are each more complex than the thumbnail sketches may indicate. File checks and balances are made, fund requirements are forecast, and so forth - none is final - all are continually being developed into more useful tools.

Some needs other than analysis for project managers exist. First, NASA-wide financial summaries and consolidations are prepared. Projects, managed in detail by centers, must sum to the programs controlled by program offices in Headquarters. Programs must be consolidated for presentation to general management. This process could be carried out most efficiently from mechanized files of data.

Secondly, under the present state of the art, wide variations do exist in presentations to management for decision making. This is hazardous where the manager is making a decision that will affect more than one center. In some cases, a particularly cogent point brought out by one presentation cannot be traced to other presentations. Thus, only rough estimates are available for comparison. Lastly, some of NASA's installations need to prepare better analysis, but do not have the capability, or resources to acquire the capability to develop computer programs for financial management. An agency-wide program would assist these installations and the agency by assuring a higher minimum standard than can now be imposed.

These three needs, - summarization - comparability - and more uniform capability - indicate that single minded serving of project managers or even individual centers is not enough.

Our approach to developing an agency-wide standard program is a fairly typical one. This does not make the road any easier though. The first step will be to assess the contractor financial management needs of Headquarters - that is, general management, program management, and the related staffs that program, procure, and administer NASA effort from the agency-wide standpoint. Hopefully, we can develop the real needs of Headquarters for Contractor Financial Management Reporting. We have the ultimate goal of developing a set of data requirements that will serve through a management changeover or reorganization. We expect to produce routine, recurring status reports, highly summarized when things are smooth and special one-time reports to activate solutions when trouble occurs.

Secondly, and probably in parallel with the first, we will assess the center and project managers' needs and their approaches to analysis. Our goal is not to establish an immutable policy regulating how project managers are to manage or analyze, but to determine what they need to know and what they are doing. This is so we can ensure rapid response to requests for data that can be met, and advise on those requests that cannot be readily fulfilled, along with the alternatives that are available.

Thirdly, we will ensure that, as a minimum, those programs that are in operation can meet Headquarters' needs. This ought to be a straightforward comparison of the needs determined in step one and the in-use programs located in step two. The necessary modifications to the in-being programs will be made to the centers' systems to meet Headquarters' needs. It probably is important to emphasize here again that we do not anticipate covering Headquarters with an inch or so of computer prepared runs every month. Detail will be required only to analyze trouble. Routine reports will be brief and to the point. We are after a system that will have available detail when needed, but not recurring proof that we have detail.

Our fourth step will be to develop a basic program for computer analysis of the NASA Form 533 that all NASA installations can use. This program will furnish to project managers analyses that will help them and as by-products installation-wide and NASA-wide summary data that will help installation management and NASA Headquarters.

Finally, we will establish a clearing house for dissemination of variations to the basic program, developed to meet individual needs. Not all variations are to be used by everyone all the time, but are to be available when needed to solve particular problems.

The heart, of the matter, as you well recognize, is the step two review of needs and available programs and the end product standard program of step 4. This must be done for the project managers. They all need rapid response. A program that will ensure any and all project managers of minimum analytical data from common equipment, is necessary. It appears that this can be done the same way it was done for PERT-time, we will pick-up contributions from all who have any to offer.

Up to now, no change in the contractors preparation or submission of the report has been discussed. There are some obvious advantages to a more cooperative arrangement. Actually, we have in being arrangements for major contractors to key punch and transmit the report by transceiver (when available) to at least one major center, with tests actively underway at another.

This same center also has a computer to computer hook-up for magnetic tape transmission of at least two major contractors' reports. One of the contractors participating in the tape to tape transmission has a fairly sophisticated computerized system of its own for report preparation.

We hope, when our standard program is available, that new contractual reporting arrangements may prove feasible especially for new starts. That is contractors may use the standard program, augmented by approved file content and update programs to furnish only exception basis reports to NASA.

Another thought that will be explored is to take advantage of the "fixed format" of the NASA Form 533 and use optical scanners in the contractors' plants to create file input. Those who have studied optical scanning see many advantages in optical scanning over conventional key-punching and verifying for inputs to computer files.

NASA is also looking to a simplification of the general approach to time/cost correlations. Essentially to the bringing together of the PERT networks and the data file created from the NASA Form 533. This goal is harder to reach because we must accommodate cost changes that are not in a cause and effect relationship with schedule changes and vice-versa. We know of significant cost variations that have no impact on schedule variations, and believe the opposite may occur too.

On the bright side, however, good programs are already available to those who need them. Programs that can be operated at the NASA Companion Cost level and do not have to be used in the depth of mandatory detail that DOD/PERT/Cost entails. We have, in fact, operated a time/cost correlation computer program on a NASA project that used the PERT and Companion Cost system. We will keep abreast of these programs so that they will be usable by NASA whenever a close continuous time/cost correlation is advantageous to a project manager.

It seems likely that we will allow others to keep the lead in programming time/cost variation correlation, adpating existing programs as needed to specific NASA projects. This is not to say that our centers will not learn by doing. They have and will continue to pursue approaches that will make meaningful presentations to project managers.

NASA's policy is to avoid specifying how or by what system the contractor shall manage his effort, instead NASA requires only reporting in a prescribed format of the summary time and cost data required for total project integration and management by the NASA project manager. The use of PERT, PERT/Cost, or PERT and Companion Cost at the contractor level of management, which backs-up reporting to NASA, is optional to the contractor.

By way of summary then, NASA will continue to take advantage of mechanized analyses of Contractor Financial Management Reports. A standard analysis program will be developed. We will cooperate with contractors on mutually advantageous ways of preparing the reports as economically and expeditiously as possible. We are after the total contractor financial management picture, not pieces of it, but we will not prescribe a system for contractors to use in their own management.

We hope by this effort to emphasize active reports - accurate, controlled, timely, informative, verified, and effective, and to eliminate idle reporting, that is, incomplete, or duplicate, or late, or erroneous reports.

We want the best possible Financial Management Reporting System.

QUESTION AND ANSWER SESSION

The following is an edited transcription of the questions and answers which arose during the informal discussions of the NASA/Industry PERT Computer Program Conference held on July 22 and 23, 1964.

Q. What is the definition of a fragnet and how do you identify them?

A. by Larry Stevens: I use the term fragnet and subnet to be synonymous. There has been some differentiation in the two. A fragnet means a portion or just a fragment of a larger network. A subnet is again just a portion of a larger network. The word subnet has sometimes been used to mean a lower level of network whereas fragnet would mean the same level network but just a portion of a larger network. During all of the presentations today, I believe fragnet and subnet were used interchangeably.

Q. Is NASA PERT "C" still basically the Lockheed program but only modified for input-output purposes?

A. by Larry Stevens: I won't go into a lot of detail on this, but my answer is "no". The systems configuration now is drastically different from the original Lockheed program. The input phase of the original Lockheed program terminated processing of the network each time a data error was found. It had a fixed input format. PERT "C" will allow 3 input formats. The topological ordering technique is entirely different. The Lockheed program did not have the capability of detecting loops. When the computer actually went into a loop, the only way to detect it was when the lights came up bright on the console. Then the loop had to be found by hand. The internal sort routine is the same as was in the original Lockheed program. The Merge Technique is entirely different. The original Lockheed program had a capacity of 5,120 activities. Our current maximum is 30,000 activities with no restrictions on network structure other than 750 starting events. There are a few parameters that could be changed to make the PERT "C" merge program completely open-ended. The expected date calculation on Pass 4 of our current program has calculation options that naturally were not in the original Lockheed. In the report generation phase, there were 4 reports available under the Lockheed version. They were all produced any time a network was run. The report phase required 8 passes to produce these 4 reports. The current program has 6 reports, requiring 6 passes. Each of these reports are optional. So, I think you can readily see that the systems configuration is entirely different from the original Lockheed.

Q. In the PERT "C" program, how many predecessor activities can any one Event have?

A. by Mr. Larry Stevens: As we first distributed PERT "C" to some of our contractors reporting to Manned Spacecraft Center, the program had a maximum of 30 predecessor activities to any one event. This maximum no longer exists. Any number of predecessors may be in existence to any event.

Q. Why do you sequence check in pass one if you sort in pass two?

A. by Larry Stevens: Input data is in sequence by predecessor and successor event number. The topological ordering method requires the activities also in sequence by successor-predecessor. This is a reverse sort from the input and requires a sort pass.

Q. What is the official NASA position on the cost of NASA PERT as a direct contract charge versus an indirect overhead charge?

A. by Mr. Walter Haase: At this time, there is no formally documented NASA position-although one is under preparation. There are a variety of ways in which PERT is being charged in various contracts that we have. We recognize that the government must pay for any systems reporting requirements which are generated by us. However, we don't want to pay twice; for a PERT system which is operated for PERT sake and another system which is operated by the management of the company for the company's management purposes - nor do we wish to pay for "planning, scheduling and management" in a calculated overhead rate and for PERT as a direct charge. This is one of the reasons why we don't specify by what system you shall do your management. If there is going to be a direct charge for management purposes we would expect that there be a appropriate reduction in indirect costs. We have given some detailed instructions to our auditors. I don't know how much trouble they have been causing you but we hope they have been causing some in this area and that they are looking very closely into the methods by which PERT is being charged. The audit people, financial management people, ourselves, and a number of others, are presently engaged in development of policy on how management information systems efforts should be charged. We feel that we are going to be breaking activities associated with information systems into fairly definable packages, and indicating which of these should be direct or indirect. Also, we need to get together with the Department of Defense on this proposed policy. Unless we can agree on some basic policies, it would be sort of ridiculous for NASA and DoD to have incompatible policies in this area. We do feel however, that the major cost associated with the PERT system should be indirect charges rather than direct because of the nature of the activity.

Q. What is NASA's current thoughts or feelings concerning PERT/COST?

A. by Mr. Walter Haase: I think we have touched on this several times today. I have mentioned it-and Ken Mulcahy has mentioned it. We are trying out various PERT/COST computer programs to see how they work with summarized Companion Cost information. Our basic feelings toward the PERT/COST system have not changed from that indicated in the NASA/DoD PERT/COST Guide, and PERT Guide for Management Use. We feel that these are a guide. We believe that the concepts and principles that are in those documents are sound and good.

We advocate contractors to go in the direction of PERT/COST for their own operating management systems, however we recognize that it is going to be a period of time before people can adjust to something like that. There are too many existing systems that are involved. We have not as yet required the use of PERT/COST on the part of the contractor and, as far as I know at least, we don't anticipate it in the next week or two.

Q. Would NASA be interested in considering the desirability and feasibility of incorporating into their program a routine which produces an output of such nature that the PERT analyst can determine at a glance the effect on the slack of all pass in the network of the change in time estimate for any activity? Who should be contacted? I have programmed this routine in COBOL and would like to share its advantages with others. I have a source decking program listing with me.

A. by Walter Haase: We are, of course, always interested in any developments and ideas and the people to talk to are the people who are here today, and this is submitted by Mr. Thompson of General Precision. Maybe we can arrange to have you talk on the subject about 5 minutes tomorrow during the informal discussion.

Q. Will you issue any documentation describing PERT "C" system logic? Will you be issuing an Operator's Manual?

A. by Walter Haase: This was directed to me but I think perhaps that Larry or John should answer this.

A. by John Leonard: We have plans for developing flow charts of our systems logic, which are much more detailed than what we have in the PERT "C" Manual. We do not have any plans at this time for developing an Operator's Manual as such, but personally I can definitely see a need for one. I think at the next NASA/PERT Coordination Meeting at which all Centers will be represented, we will probably make a proposal that this Operator's book be developed and made available to users of the system.

Q. To Homer Smith: Homer, what PERT program are you currently using in your PERT operations, and the second question: what is your PERT production volume?

A. by Homer Smith: I wish that question hadn't been asked because we are currently running two PERT programs; no, three--the third one is currently "by remote". One division of Boeing is using the NASA "B" program and is reporting to one of the NASA Centers by sending in the deck. The rest of our PERT processing within Boeing is being done on an in-house program that we wrote several years ago. It was written with a specific purpose in mind--networks of less than 5,000 activities and minimum number of reports. We use this on all projects where the PERT requirements are such that we can get away with using this program. For the rest of our contracts, we are running the Air Force system.

John?

Q. What's the mailing address for sending tapes to obtain PERT "C"?

A. by Mr. John Leonard: Here is the address, if you would like to write it down: Data Systems Development Branch, Computation and Analysis Division, Manned Spacecraft Center, Houston, Texas, and you might make it to my attention. The question was also asked: Where to send tapes for Fortran IV? I believe this was answered earlier. It is being distributed through the computer user groups, such as SHARE.

Q. Are the selected ending events edited for existence in the PERT network?

A. by Mr. Larry Stevens: Ending events are edited for existance on end runs. If an event does not exist, that end run is skipped and the next end run is processed.

Q. Is more than one input format available?

A. by Larry Stevens: The NASA PERT "C" Manual includes only one input format and that is the NASA-wide format. The NASA PERT "B" Manual lists two other available formats; one called the MSC format and the other a Langley format. The PERT "C" Program will accept all formats that were acceptable under PERT "B".

Q. Do the tables in Pass 4 for posting an expected date have a capacity limit? Is there a limit on the width of a network?

A. If you had a network with 10,000 parallel paths of 3 activities each, it would not process on the PERT "C" Program in its present configuration. If you desire to have it processed, a slight modification could be made to the program and it could be processed. However, as distributed, PERT "C" has a capacity of approximately 9,250 parallel paths at any one time.

Q. Will PERT "C" loop diagnostics print online, if desired, rather than on the report tape?

A. by Mr. Larry Stevens: All diagnostics--the Pass 1 input diagnostics, the loop detection diagnostics, and the Pass 4 diagnostics which are constrained activities and incomplete activities with no predecessors--all of these diagnostics are produced on the A3 output tape. If desired, they may also be printed online by simply depressing sense switch 5 on the 7094 console. If you do not like to use sense switches, it would be a one-card change in 3 affected passes to force the diagnostics to always print online.

Q. What input errors specifically will cause termination of a PERT "C" run, and are all errors detected before a run is terminated?

A. by Larry Stevens: Input errors such as date, sequence, and transaction code errors, will terminate the run. However, the run is not terminated until after the Pass 4 or network continuity errors are also checked. So, first, Pass 1 errors will terminate a run. These are the input errors, such as sequence errors, duplicate errors, predecessor-successor equal, invalid schedule date, invalid completion date. Second, a loop will terminate a run. Third, network continuity errors detected in Pass 4 may terminate a computer run. The normal process is to terminate the run, however, if sense switch 4 is on the run will continue. The constraining activity producing "constrained but complete activities" will be ignored and "incomplete activities with no predecessors" are made complete activities with the "date of the report" as the "completion date". On option you can either continue on with the run, ignoring Pass 4 continuity error, or terminate the run, examine the diagnostics, and if they have little effect on the final results, you can restart at the beginning of Pass 5 and continue the run. Of course, the normal procedure is to correct any input errors and start the run over.

Q. In regard to the suppresion of completed activities, are they retained on the master file and considered in calculations?

A. by Mr. Larry Stevens: There are two methods of completing deleted activities; the first is prior to any processing on PERT "C". This requires an additional Fortran Program which removes the completed activities from the input data. The other method is deleting completed activities during processing time in the Pass 4 calculation. As soon as the date has been picked up from the completed activities for its successor event in Pass 4, the activity is removed from any further processing. It is not carried through the latest allowable date calculation or any of the subsequent sort passes but is removed as soon as it is no longer needed for date calculations. This method has no affect on the input file.

Q. This is a very similar question. When completed activity cards are submitted as a part of an update, do these activities appear in the output report if the option is taken to remove all completed activities?

A. by Larry Stevens: If you decide to delete completed activities at execution time, all completed activities are deleted--no matter when they were entered into the network.

Q. When is PERT "C" not tape bound?

A. by Mr. Larry Stevens: We are tape bound during all but three phases of the program. Those phases are: the topological ordering phase, the expected date calculation, and the latest allowable date calculation.

Q. Isn't sample output available on the 1401 graphic scheduled display?

A. by Mr. Larry Stevens: The output is available here on the rostrum if you would like to have a sample. It is the more or less standard graphical output. Instead of the expected,latest allowable and scheduled date, it prints E, S, and L on a chart underneath the date; the calendar is printed at the top of the page.

Q. Does the Fortran tape update package containing all of the features of the 1401 update package?

A. by Mr. Larry Stevens: The Fortran update in its present configuration is strictly an update. We hope to incorporate many of the 1401 update features into it as soon as possible. In its present form it takes a blocked input tape, 16 activities per record, updates that information with the changes required and prints a list on the changes only.

Q. What 1401 machine configuration is required to use the 1401 update package?

A. by Mr. Larry Stevens: We have two packages available; one uses an unblocked tape for a 4k 1401. The program that we use in-house uses a block input tape of 16 activities per record. This requires an 8k 1401 with more record feature and indexing. Both programs require two-tape units, and a third tape unit is optional.

Q. Can the update program be run on a Honeywell 8200 12k?

A. by Mr. Larry Stevens: I understand that Honeywell has available a program for converting a machine language 1401 program. These 1401 programs, incidentally, are written in SPS. Therefore, if my information is correct, the programs could be converted and run on the Honeywell equipment.

Q. What type of sort and merge technique is used in PERT "C"?

A. by Mr. John Leonard: We had about 10 minutes set up with some slides to describe our sort and merge technique but we are just not going to have the time. I'll try to give you a very quick broad, brush description. The technique uses a variable amount of tape units, depending on the size of your input, from no merge or scratch tapes up to a total of 6, depending on the size of your network. With very little modification this routine could be made an open end sort and merge technique. There are no "end of tape" tests on the merge tapes. Therefore, the maximum size file that could be sorted would be limited to the capacity of a tape. I would like to point out one thing--a lot of the NASA centers are considering going to a moonlight or direct couple system and we believe that, with a minimum of reprogramming of the sort and merge package, that we can tremendously cut down the running time of PERT "C" on such a system. Instead of utilizing six-tape files for merging, we will take advantage of the almost unlimited files available on the disc. So, as the Fortran Program can be made to run faster with a larger core or disc, certainly the efficiency of PERT "C" can also be improved by utilizing this hardware. We are currently working on modifications for the Direct Couple System.

Q. I have a question as a user of the computer program. I have a question which may end up as a criticism on your PERT "C" Program. And so, let me start this way. One of your first passes will sort the program--I mean, you have to pre-sort your event numbers, right? 14 columns? Right?

A. by Mr. John Leonard: That's true.

Q. This is the same as the Old Lockheed Program?

A. Yes.

Q. Now, my criticism is: "Why in the devil did you do that?" One of the biggest things, as a user, that I have had problems with on any of these other programs--is that I have to pre-sort these event numbers.

A. by Mr. John Leonard: As we develop the Fortran update package, this is definitely one of the things we want to do first. One of the first modifications that we will make is provide the capability to read in the data; check it for sequence as it is read, and if it is out of sequence, go into 90 sort; sequence it, then go directly into PERT "C" Program all as one run on the computer.

Q. No pre-sorting then?

A. by Mr. John Leonard: Right, no pre-sorting.

Q. Your PERT "C" program does--I was thinking of the old Lockheed Program--every time you came to something out of order, it would stop; if I had 10 errors, it would stop 10 times before I found them--you know what I mean? You now view them all and give me one list of all sequence errors, isn't that right--and then terminate the run?

A. by Mr. John Leonard: Larry, can you answer that? Does PERT "C" catch all sequence errors in Pass 1?

A. by Mr. Larry Stevens: Not only catches all sequence errors but continues editing for other types of input errors. All errors are found in one pass unless a loop exists which prevents network continuity checks.

Q. What is the running time of the NASA PERT 1410 program developed by the John F. Kennedy Space Center?

A. by Mr. Al Sheffield:

1410 time for PERT run

1. 600 activities
2. Runs requested
   2-A Successor - predecessor event
   2-B Paths of criticality
   2-C Expected date
   2-D Latest allowable date
   2-E Organization
   2-F Slack sort of start dates
3. Run time - 20 minutes

Q. What topological technique do you use in the 1410 program?

A. by Mr. Al Sheffield: Topological ranking is done during pass two. The records enter the system in predecessor sequence. As the records are read they are assigned a number from 1 to xxxx (number of activities.) The tape is rewound. The successor rank field is compared to predecessor rank field and the successor field must exceed the predecessor rank field by a value of at least one. This procedure is completed until no successor rank field is incremented. The records are then sorted on the successor rank field.

Q. What editing procedure do you use in your 1410 program?

A. by Mr. Al Sheffield: PERT Edit for 1410.

There are three phases to the new PERT Edit for the 1410. Phase one checks each input record for valid characters, sequence checks the input and checks for Duplicate Records. Below is a list of error messages and their meanings that are used in Phase I.

1. Delete this record — A "5" card was merged in during EAM run. Pull this card.

2. Col 1 Not 1, 2, or 3

3.	Pred. all zeros	
4.	Pred. Blnk/alpha	
5.	Succ. all zeros	
6.	Succ. Blank/alpha	
7.	Est. time in error	Not all numeric
8.	Resource in error	Not all numeric
9.	Day is in error	
10.	Year below 62	
11.	Comp Acty blnk dte	Every com activity must have a date in col. 26-31
12.	C Dte after R Dte	Comp date after run date
13.	Deck out sequence	
14.	Month is in error	
15.	Duplicate Records	

Phase 2 is a check for constrained activities; no predecessors and dangling successors. A constrained activity is one that has a non-complete predecessor of a complete activity.

No predecessor means there are no other activities coming into this activity. To be a legitimate starting activity, it must be a completed activity.

Dangling successor means this activity does not lead to another activity. Only those terminal activities designated by the customer are legitimate terminal events numbers.

Phase 3 is a listing of the Input in Successor Sequence.

There are no halts in this program. All error messages are put on Error tape and editing is continued.

Q. In the Fortran IV Program, how is the following situation handled--a terminal event which has three or more predecessors each with different schedule dates--will the program use the latest schedule date?

A. by Mr. Ross Bainbridge: We have made note to our users of this situation in the hope that only one scheduled date be used. The program uses the first scheduled date encountered. The situation here is that we assign the scheduled date to an event. As long as one activity defines the event scheduled date, then this date is used. Of course, if no scheduled date is found, then the expected time is used as the scheduled date.

Q. Are there any plans to form a PERT-COST program in FORTRAN or a supplemented FORTRAN?

A. by Mr. Ross Bainbridge: At Lewis as well as other NASA installations cost programming is being done. None of this is being done at present to produce a standard NASA Cost or Comparison Cost system. If the work were to be done at Lewis on a cost program, then the program would almost certainly be done in FORTRAN IV. This program would also most likely be a new independent program using output from the time program. One of the features of such a cost program would be a powerful report generator.

Dan Hirsch of our Lewis PERT Office has done some cost work on the Lewis-Goddard-NASA PERT TIME I program. For further information as to the results of his work we feel that contact should be made directly with him.

Q. In the Fortran IV program, if a summary report is requested, are the other subnet reports also available during that computer report pass?

A. by Elizabeth Ryan: Yes, as many reports as requested can be obtained.