

J. H. Tresler

5450

NRL Report 5919
NAREC Reference 27

UNCLASSIFIED

**A PROGRAM FOR THE EXECUTION
OF LGP-30 MACHINE-LANGUAGE CODES
ON THE NAREC COMPUTER**

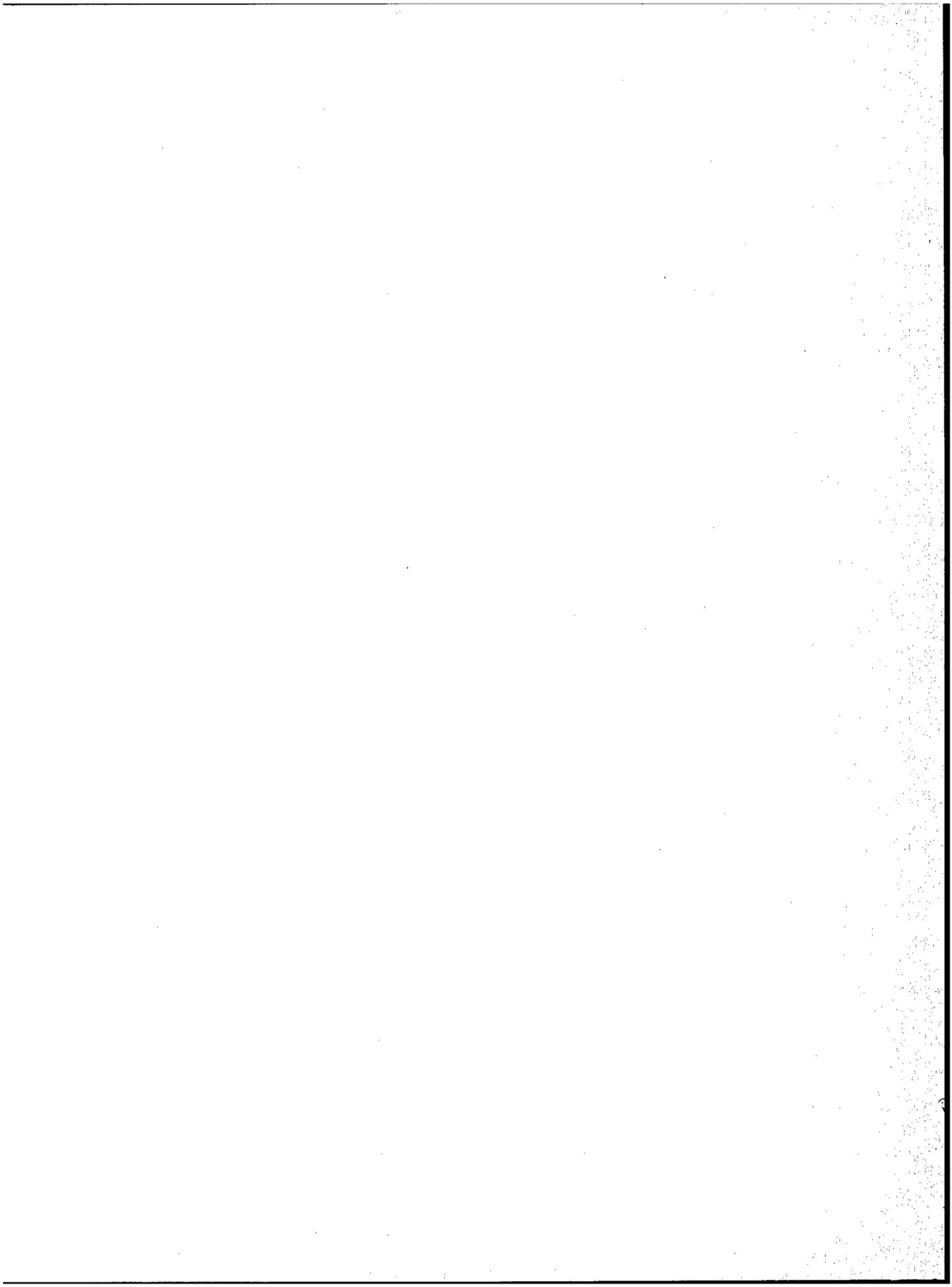
Elizabeth E. Wald and B. Wald

Radio Division

July 1, 1963



U. S. NAVAL RESEARCH LABORATORY
Washington, D.C.



CONTENTS

Abstract	ii
Problem Status	ii
Authorization	ii
INTRODUCTION	1
GENERAL DISCUSSION	1
The Role of Large and Small Computers in a Research Organization	1
Methods of Providing Intermachine Program Compatibility	3
Summary	4
INSTRUCTIONS FOR PERFORMING A SIMULATION	5
General Considerations	5
Simulation of Switches and Pushbuttons	5
Program and Data Inputs	6
Outputs	8
Starts and Stops	8
Sample NAREC Operator's Instruction Sheet	8
DETAILED PROGRAM DESCRIPTION	9
Representation of Memory and Registers	9
Input-Output Codes	12
Simulator Initialization	12
Simulation of LGP-30 Sequencing	12
Simulation of LGP-30 Orders	15
CONCLUSIONS	22
ACKNOWLEDGMENTS	23
Appendix A - Simulator Efficiency	24
Appendix B - Translation Methods	30
Appendix C - Simulated Flexowriter Manual Keyboard Input	36
Appendix D - Coding of Simulator	42
Appendix E - Flow Charts for the NAREC Simulation of LGP-30 Operand Orders	84

ABSTRACT

In order to allow the utilization of digital computer programs written for the LGP-30 computer by an organization which no longer had convenient access to this machine, a program was written for NAREC, the NRL general purpose computer, which causes NAREC to simulate the actions of an LGP-30. Although this program was written out of necessity, the simulation speed gain is sufficiently high to make the simulation of LGP-30 operations more economical than direct execution for many classes of problems. A more cogent argument for the utilization of simulation is to free NRL's LGP-30's from production work and allow them to be used in close support of research.

The simulator utilizes the LGP-30 instructions in an interpretive mode to invoke subroutines or action blocks to perform on the simulated LGP-30 memory, registers, and input-output devices the same actions that would have been performed by an actual LGP-30 executing these instructions. Short-cut procedures are available to increase the utility or speed of common LGP-30 operations. The construction of the simulator is fairly straightforward, except that all characteristics of input-output equipment as well as the computer itself must be carefully simulated.

One year of operating experience with the simulator has demonstrated that while there certainly exist better ways of providing interchangeability between computers, e.g., by not programming in machine-oriented language, it is practical to use the actual machine coding of a problem written for a small computer as the control for a larger computer running an interpretive program.

PROBLEM STATUS

This is an interim report; work on the problem is continuing.

AUTHORIZATION

NRL Problem R06-06
BuShips No. S-1931
Project SF001-08-05, Task 9094
and
NRL Problem R06-07
Projects RF001-08-41-4552
and SF001-08-01, Task 9249

Manuscript submitted January 15, 1963.

A PROGRAM FOR THE EXECUTION OF LGP-30 COMPUTER-LANGUAGE CODES ON THE NAREC COMPUTER

INTRODUCTION

The Radio Division has been utilizing both medium- and small-scale digital computers in the course of its work. The recent loss of a small-scale computer, for which a number of programs existed, prompted an investigation of the feasibility of running these programs on NAREC, NRL's general-purpose, medium-scale, digital computer. It is thought that the results of this investigation may be of some general interest.

The first major section of this report, entitled General Discussion, may be of interest to persons facing similar problems as it explains the course of action pursued to solve our problem. The next section, entitled Instructions for Performing a Simulation, is addressed to those persons who wish to use the results of our work to execute existing LGP-30 programs on NAREC. It is assumed that persons reading this section are quite familiar with LGP-30 programming and operating practices but possess only a casual knowledge of NAREC. Access to the third major section, entitled Detailed Program Description, requires considerable familiarity with both machines. The detailed information is offered so that later modifications may be made to the simulator and so that possible help may be given to others who may be considering the construction of simulators.

GENERAL DISCUSSION

The Role of Large and Small Computers in a Research Organization

The Naval Research Laboratory utilizes a diversity of general-purpose digital computers. The Research Computation Center operates NAREC (a computer of the IBM 704 class) and provides analytical, programming, operating, and maintenance services to make this computer available to scientists throughout the Laboratory. In addition, several divisions and branches operate small digital computers, largely for their own projects but also on a time-available basis for the entire Laboratory.

It has been often observed that the cost-per-operation of a digital computer steadily decreases as the size of the computer increases. For example, NAREC can perform three million additions per dollar, while an LGP-30 is capable of performing about twenty five thousand additions per dollar under typical conditions.*

The natural implication is that the use of small computers is uneconomical when large computers (or even medium computers such as NAREC) are available. Upon closer examination, however, powerful arguments can be raised in defense of the utilization of small computers.

*The assumption is made that the LGP-30 is leased, that it can be supported for six dollars per hour, and that 40 percent of the orders are optimized. It may be argued that the LGP-30 is obsolete, but the NAREC is also no longer modern. At any rate, few will dispute the general principle that for computers within the established state of the art, computing speed increases more rapidly than price.

First, for many problems, only a relatively small number of arithmetic operations are required. Since either a small or a large computer requires comparable time to load the problem, it may be more economical to run the problem on the small computer.

Second, there exists a class of problems which has a large ratio of output to computation. NAREC would be punch-limited in computing speed on such problems and might have only a three-to-one speed advantage over an LGP-30. The addition of a faster punch to NAREC will soon make this argument less relevant. Further development of the simulator program to use a line printer will make it completely irrelevant, but at the current instant there are problems of this kind that may be run more economically on a small computer.

Third, the cost estimates assumed that programming costs would be the same for any machine. While the programming costs could well be expected to be comparable, debugging costs tend to be higher on NAREC since debugging at the console requires equal programmer time in either case and equal machine time at a higher rate in the case of NAREC. Debugging away from the console in the NAREC case raises the programming costs or, if the snapshot system is used, raises the machine time costs.* This may be a significant factor in a one-shot problem where programming and debugging costs overshadow running costs.

Fourth, there exists a class of relatively simple programs which are frequently run to guide the course of an experiment. In these cases convenient access to the computer is the overriding consideration.

Finally, while all scientists should be aware of the ways in which the large computer will help their work, they are human, and some of them will be intimidated by the central installation and may be diffident about using the installation at which experts may scoff at their fumbling. Since they may use the small computer in privacy, they will not have these reservations and it can be argued that it is better that they should use computation inefficiently than not at all. Hopefully, after "cutting their teeth" on the small installation, they will no longer hesitate to use the efficient large installation.

The authors do not presume to pass on the validity of these arguments; they merely wish to point out that small computers do exist at NRL and that their use will probably continue.

One point seems to run through the previous discussion—that many of the advantages of the small computer lie in the area of programming and debugging. Once this program is running on a production basis, the relative computational inefficiency of the small machine begins to predominate. It becomes worthwhile, therefore, to raise the question whether programs written and debugged on the small computer at low cost per hour can be run in production on the large computer at low cost per operation.

In the case of the authors' Branch this question was particularly relevant. A large investment had been made on the writing and debugging of LGP-30 programs, utilizing a machine of another branch which was available to us when not in use by the organization which controlled it. This organization recently moved the computer outside the confines of NRL, and we knew for some time that we were facing its loss. Furthermore, some of these programs, while written as feasibility demonstrations, were being used for production. We have therefore investigated two questions: first, the specific question of how to utilize our existing programs on NAREC, and second, the general question of how to provide running compatibility between the large and small machine for new programs.

*While it is true that the additional aids available to NAREC (or any larger machine), i.e., compilers, assemblers, etc., may decrease programmer time, machine time is higher when reassembly or recompilation is necessary.

Methods of Providing Intermachine Program Compatibility

What we desire is a method whereby programs written and debugged on one machine can be run on a different machine. In considering this problem we must distinguish between the various languages in which programs are written.

Originally most programs were written in machine language, that is, the programmer specifies the actual bits he wants placed in the machine. The programmer's product is loaded into the machine either by a hardware command (the NAREC 32 order) or by a simple loading program (the LGP-30 program input routine).

Since machine languages differ from machine to machine, it is clear that some sort of translation process is necessary. The most direct and desirable process would be a computer program which would translate each order for machine A into an order for machine B. This program would produce as its output a machine-language program for computer B.

Unfortunately this type of translation program is difficult, if not impossible, to write. One serious difficulty occurs when the address coding structures of the two machines differ. It is then difficult for the translator to decide whether an addition order is for computation or for address arithmetic purposes. In the former case, the addend should be translated intact; in the latter case, it must be translated to correspond to the same number of address locations in the address structure of the new machine. Other equally serious difficulties arise when the writing of such a translator is attempted.

If it is not feasible to write a direct machine-language to machine-language translator, it is reasonable to inquire if there are other languages more suitable for translation.

After machine-language programming, the next level of sophistication is assembler-language programming. Under this arrangement the programmer writes in a language which, while oriented to the structure of the machine, is more general than machine language. Since an assembler for a particular machine is oriented to that machine, there is no reason to suppose that assembler language can provide intermachine compatibility.

It is only at the level of compilers that programming languages offer much hope of intermachine compatibility. A compiler is a program that accepts a program written in a problem-oriented language. The best-known compiler source language at this time is probably FORTRAN. If two different machines are equipped with compilers utilizing the same source language, compatibility implies only recompiling the program with perhaps a preliminary transliteration of characters to account for different bit representations of source-language symbols. Thus a program which had been developed and debugged on a small computer could be transliterated and recompiled to produce a machine-language program for a larger computer.

The use of compilers is certainly an attractive procedure, but upon closer examination certain restrictions and reservations appear. In our immediate case the LGP-30 programs had been written in machine language, and the technique would be of no immediate help to us. Even from a longer term viewpoint, at the time of this investigation there existed no compiler for NAREC. Existing compilers of the LGP-30 leave much to be desired. If this technique were to have much general utility, a number of things would have to be done. First a common source language would have to be adopted, and compilers would have to be written for both NAREC and the smaller machines which would accept this source language. A logical choice for this common source language would be ALGOL.

The compilers would have to be well written to allow the programmer the flexibility of the source language and also to utilize the machine efficiently. Furthermore,

programming systems would have to be built around the compilers so that debugging could be done in source language. Programmers would have to discipline themselves to make corrections in the source-language program so that the debugging process will produce a correct tape in compiler language. Unfortunately this implies frequent recompilation during debugging, and compilers on the small machine are likely to be inefficient.

If the above attack were to be adopted, an ALGOL compiler would have to be written for NAREC. There are a considerable number of NAREC programmers who feel that this would be a desirable step in any event, as they would make use of such a compiler for programs written solely for NAREC; indeed, the construction of a compiler based on the NELIAC language (a dialect of ALGOL) commenced about a year after the initiation of this investigation. ACT, the existing LGP-30 compiler, does not implement the ALGOL language, but Dr. Thomas Kurtz of Dartmouth is in the process of constructing such a compiler. An ALGOL-language compiler (ALGO) has recently become available for the Bendix G-15, another small computer.

Although this discussion indicates a possibly desirable course for the future, it did not solve our immediate problem. We were therefore forced to construct an interpreter to allow NAREC to execute LGP-30 machine-language programs. The philosophy employed in the construction of the interpreter is that since the LGP-30 machine-language program reflects the structure of the LGP-30, the interpreter must be a NAREC program that also reflects the structure of the LGP-30. Specifically, the interpreter contains within it, locations that correspond to the program-accessible registers of the LGP-30, and these pseudoregisters in the NAREC program are caused to contain the same bits that their LGP-30 counterparts contain. The interpreter then manipulates the data in these pseudoregisters in the same manner that the LGP-30 would have manipulated its own registers. This implies that each order in the LGP-30 program must cause the interpreter to invoke a subroutine which simulates the action of the LGP-30. An exception to this rule is discussed in Appendix A.

This technique certainly solves the logical problem of how to attain compatibility, but there is some question as to the extent to which it solves the practical problem. The simulation philosophy implies that each order in the LGP-30 program causes the interpreter to execute many NAREC orders. Whether this inefficiency offsets the lower cost per operation on NAREC depends on the precise nature of the program being interpreted. Detailed measures of efficiency can be found in Appendix A.

Summary

We have not found it feasible to write a translator that will directly convert LGP-30 machine-language programs to NAREC programs. Although we feel that a desirable procedure would be to provide intermachine compatibility through the use of a common, compiler source language, this technique is subject to several limitations and offers no help when the source material is an existing LGP-30 machine-language program. We have therefore written a NAREC program which is an interpretive routine that reflects the structure of the LGP-30 and invokes a short subroutine for each LGP-30 order being interpreted.

The operating instructions for this interpreter are given in the next section, and a detailed description of the workings of the interpreter are given in the section entitled Detailed Program Description.

INSTRUCTIONS FOR PERFORMING A SIMULATION

General Considerations

The intent of the simulator program is to make the NAREC computer behave as if it were an LGP-30 computer. From the user's point of view, the actions performed by an LGP-30 may be summarized as follows. First, the machine may or may not be assumed to have information retained in its memory from previous operations. This information usually includes a standard Program Input Routine (PIR) and may include results of computations. Information relevant to the current problem is next fed to the machine in the proper order. This information is imparted by setting switches and pushbuttons, by placing punched paper tapes in one of two readers, and by operating the Flexowriter keyboard. The result is that outputs are produced on one of two devices—the Flexowriter or a separate tape punch. A subsidiary "result" is the contents of the LGP-30 memory, which may be an implicit input to a subsequent operation.

From the point of view of a programmer, NAREC can perform similar functions, but these functions do not have an exact one-to-one correspondence. Techniques have been developed, therefore, to allow analogs of the information inputs to an LGP-30 to be entered in NAREC. Some of this information is contained in a control word, some in transfers of control performed by the NAREC operator, and some in switchbox manipulations performed by the NAREC operator, but the bulk of this information is obtained from the same punched tapes that would have been used to feed an LGP-30.

It is the intent of this section to provide the means whereby an experienced LGP-30 programmer or operator can formulate instructions comprehensible to a NAREC operator that will cause NAREC to process punched tapes and produce outputs identical to those that would have been produced by an LGP-30.

Simulation of Switches and Pushbuttons

A necessary information input to the simulation program is the status of the switches and pushbuttons on the LGP-30 configuration being simulated. Most of these settings are communicated to the simulator by the loading of a control word. If the NAREC operator manually executes the operation LO 1248 and then pushes the Transfer Button, the setting of the NAREC switchbox at the time the Transfer Button is pushed will be sensed and interpreted as a control word. Only the first four hexadecimal characters of the control word are used in the present version; therefore the control word, as communicated to the NAREC operator, will generally be of the hexadecimal form $(n_0n_1n_2n_3 00 0000 00)$.^{*} The significant hexadecimal characters may be constructed by reference to Table 1. It should be remembered that the NAREC terminology for the hexadecimal characters greater than 9 is a,b,c,d,e,f rather than the f,g,j,k,q,w of LGP-30 terminology. The control word specifies the condition of the breakpoint buttons, the Transfer Control Button, the 4-bit/6-bit input button, and the input and output device selector switches. If the Flexowriter is selected as an input device, the control word further specifies the condition of the manual input key on the Flexowriter.

The control word also permits the simulation of modes of operation not directly achievable on an LGP-30, namely, the maintenance of 44 bits of precision in computation, the printing of a monitor trace, and certain variations in Flexowriter operation that will be discussed at a later point.

^{*}An exception is discussed in Appendix A.

Table 1
Control-Word Formulation for Simulating LGP-30
Switches and Pushbuttons on the NAREC

Hexadecimal Character	Bit	0	1
n_0	-3 -2 -1 0	31-Bit Accumulator	44-Bit Accumulator
n_1	1 2 3 4	Breakpoint 32 Up Breakpoint 16 Up Breakpoint 8 Up Breakpoint 4 Up	Breakpoint 32 Down Breakpoint 16 Down Breakpoint 8 Down Breakpoint 4 Down
n_2	5 6 7 8	Transfer Control Up Manual Input Up 4-Bit Input Flexowriter Input	Transfer Control Down Manual Input Down 6-Bit Input Photoreader Input
n_3	9 10 11 12	Flexowriter Output Omit Stop Code Delete Input Copy Omit Trace	Fast Punch Output Print Stop Code Copy Input Execute Trace

The use of the 44-bit precision feature should be considered only after an examination of the effect of N orders which may be used in address arithmetic.

Invocation of the trace feature causes the following information to be printed after the execution of each LGP-30 order: the location of the order just performed, the order itself, and the resulting accumulator contents. The first two items are printed in normal LGP-30 track-sector notation, and the last in LGP-30 hexadecimal notation. If, however, the order just executed was a U or an effective T, the order location will be replaced by a tab. This last feature allows rapid spotting of transfers of control.

The control word may be changed only at the beginning of the simulation or when the simulated program has come to a stop. Setting of the control word will cause a listing of the status of the simulated switches on the output tape.

Activation of the Start Button on the LGP-30 can be simulated by the performance of the operation RO 1000 (i.e., execute instruction in right side of location 1000) by the NAREC operator.

Performance of the inverse N sequence (pushing the One-Operation, Clear-Counter, Normal, and Start Buttons) on the LGP-30 can be simulated by the execution of the operation LO 1007 (i.e., execute instruction in left side of location 1007) by the NAREC operator.

Program and Data Inputs

These inputs may be divided into two general classes: the pre-existing state of the LGP-30 memory, and inputs that are part of the computation.

The most common type of pre-existing memory is a PIR. If the Royal McBee PIR 10.4 is desired, it can be placed in memory by the performance of the operation LO 1068 by the NAREC operator. If some other PIR is desired, its bootstrap must be manually transliterated into NAREC notation by the use of a procedure detailed in Appendix B.

Other pre-existing memory can be handled as an ordinary input, but a shortcut is available. If at the conclusion of a simulation it is desired to save the memory then existing, performance of the operation LO 15c0 by the NAREC operator will produce a tape which is a representation in NAREC language of the entire LGP-30 memory. At subsequent simulations this tape can be reloaded by the performance of the NAREC operation LO 15d0 by the NAREC operator. It should be emphasized that this reloading can be accomplished only through the stated operation and not by the direct execution of a NAREC read order.

It is also possible to store the entire LGP-30 memory in a portion of the NAREC memory not otherwise used by the simulator. Transfer of the LGP-30 memory to NAREC locations 2000-2fff can be accomplished by the execution of the NAREC order LO 15e5, while transfer to 3000-3fff can be accomplished by the execution of the order LO 15eb. Neither of these operations disturbs the simulated LGP-30 memory; instead, an extra copy is created which remains unchanged through subsequent simulations. These copies can be reinserted in the simulated LGP-30 memory by the execution of the order LO 15e8 or LO 15ee, respectively. This feature can be used in the simulation of multiple computer installations or can be used to store a copy of LGP-30 drum after program loading so that in the event that unexpected data garbles the program, an original copy can be reloaded into the LGP-30 pseudomemory in a minimum time.

The remaining inputs consist of data and programs (the latter actually being data for PIR) entered via paper tape or via the simulated Flexowriter keyboard. It is strongly recommended that the use of keyboard inputs be avoided wherever possible as the simulation is awkward and subject to human errors. In the class of LGP-30 programs suitable for simulation, the necessity of keyboard input can usually be eliminated by using v-coded program tapes, by punching any required ; and / codes on other program tapes, by placing the necessary . or + codes on data tapes read by PIR, and, where all else fails, by preparing short tapes of the information that would have been entered by keyboard. Simulation of keyboard input will be discussed later.

Tape inputs are handled through NAREC's photoreader, but by the use of a proper code word the operations of the LGP-30 Flexowriter or photoreader can be simulated. It should be noted that reliable simulation of Flexowriter input requires an adjustment of the timing of the NAREC SCU order. While the Research Computation Center expects that ultimately the necessity of this adjustment will be eliminated by engineering changes, or else that it will be possible to accomplish the change by manipulating a switch that would be installed on the NAREC console, at the time of this writing the adjustment requires the temporary connection of a wire attached to an extra condenser in a multivibrator in the NAREC input section.

In principle, the NAREC photoreader should handle any tape acceptable to an LGP-30 input device. In practice, however, it is advisable to recopy input tapes onto Mylar if they are to be used frequently because NAREC operators prefer not to handle the relatively fragile paper tape. If splices are necessary, they should not be made with glue; a hot splicer should be used instead. Eighteen inches of leader at the beginning and end of the tape will eliminate difficulties with the brakes and clutches on the photoreader, and a suitable label on the tape will make construction of a foolproof instruction sheet possible. It has been found that the simulated LGP-30 photoreader is much more tolerant of pitch variations in paper tape than many real LGP-30 photoreaders and that certain tapes will run properly through the simulator but not through the LGP-30 for which they were made.

Outputs

While LGP-30 outputs may be in the form of either punched paper tape or Flexowriter copy, NAREC has a line printer but no on-line typewriter, and all outputs are on paper tape.* The simulation of LGP-30 punch output presents no problems, irrespective of whether the tape was desired for Flexowriter use, computer input at a later time, or control of other equipment. When LGP-30 Flexowriter output is to be simulated, the simulator produces a paper tape which, when read by an LGP-30 Flexowriter, produces the desired copy.

When the Flexowriter is selected as both the input and the output device in an LGP-30, the Flexowriter's final output, i.e., the piece of paper (or tape if the punch key is on) will contain a copy of the input. This characteristic is simulated by setting bit 11 of the control word to 1. If this bit is set to 0, however, the input copy will be deleted. This would correspond to depressing the Flexowriter punch key only during output and considering the resultant tape, rather than the paper, as the final output.

LGP-30 output programming is subject to certain timing rules which are generally observed by programmers. If these rules have been obeyed, no difficulties will ensue. These rules contain large safety factors because of the great variation from installation to installation of the timing of output devices. Unfortunately, many programmers have discovered this fact and, following the axiom that a programmer invariably takes advantage of a hardware anomaly, have written programs which violate the timing rules but still run satisfactorily at their installation because the safety factor was not necessary in their case. Such programs which violate the published rules may or may not run satisfactorily on the simulator. In certain cases of violated timing rules, the simulator will attempt to guess the programmer's intention. If a guess is made, a note will be printed at the first subsequent effective halt in computation which implies that at least one guess has been made and states the address of the last print order executed on which a guess was made.

Starts and Stops

Table 2 lists the NAREC stops which may occur during the course of a simulation and the information seen in the NAREC control register (i.e., C register). The stops expected at each state of the simulation should be communicated to the NAREC operator so he may verify that he has performed the proper NAREC operations.

Figure 1 diagrams these stops and the possible actions that may be taken after each of them. This figure also indicates other instructions that should be given to the NAREC operator.

Sample NAREC Operator's Instruction Sheet

Figure 2 is a sample NAREC Operator's Instruction Sheet for a hypothetical simulation. The following assumptions have been made:

- a. that PIR 10.4 is satisfactory
- b. that the program consists of two tapes labelled PRO-1 and PRO-2, the first of which is v-coded and ends with '.0000000', and the second of which ends with a . code to the first location of the program

*Later simulator program developments utilize the line printer, thus providing a "hard copy" immediately.

Table 2
NAREC Stops and their Meanings

Contents of NAREC Control Register (C)	Meaning
0001 82	A good normal stop after an effective Z order
0002 82	Normal stop after the transfer of PIR from 1500-15bf to 0000-00bf
0003 82	A normal stop after an arithmetic overflow has occurred
0004 82	A jam input character has been detected
0005 82	An invalid character has been read when the little clock has run out, or programmer is using the Flexowriter input mode and has an I order with no previous P order
0006 82	A normal stop after listing of LGP-30 controls being used
0007 82	(a) Stop after block transfer of drum to location 2000 and up (b) Stop after block transfer of drum to 0000 from 2000 (c) Stop after block transfer of drum to location 3000 and up (d) Stop after block transfer of drum to 0000 from 3000
0008 82	A normal stop after execution of the reloadable dump routine

c. that the program ends with a Z0000 followed by a U to the first location of the program

d. that photoreader input and fast punch output are used

e. that Transfer Control, 6-Bit Input, and Breakpoint 4 should be activated after the program is loaded and before the first data tape is run

f. that the program is to be run on five data tapes

g. that the data tapes may contain out-of-range data (in which case an arithmetic overflow will occur, no further computation is possible on that data tape, and several locations in the program will be garbled)

h. that a tape marked RESTART contains a stop-transfer to the first location of the program.

DETAILED PROGRAM DESCRIPTION

Representation of Memory and Registers

The simulated LGP-30 drum is contained in NAREC locations 0000-00ff. One NAREC location is reserved for each word on the LGP-30 drum. A method of translating LGP-30

decimal track-sector addresses to NAREC hexadecimal addresses and vice versa is given in Appendix B, but it should be noted that the translation is only one of terminology and that the 12-bit fields are the same.

Within each of these NAREC words, the 31 bits of the simulated LGP-30 word are reproduced with binal points aligned as shown in Fig. 3. It should be noted that because of different conventions in blocking off tetrads, the hexadecimal name of a bit pattern in NAREC memory will name a number twice as large as that of the same pattern in the LGP-30 memory.

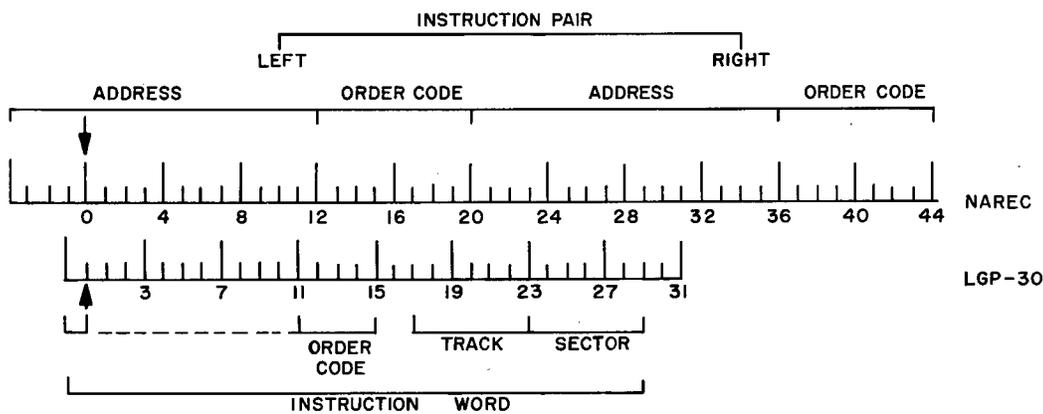


Fig. 3 - NAREC and LGP-30 instruction words

The recirculating LGP-30 registers are also represented in the NAREC memory, but there is no physical analog of the LGP-30 oscilloscope available. Thus the registers of the simulated LGP-30 can be read only by calling them manually to the NAREC core memory display device.

The LGP-30 pseudoaccumulator is located at NAREC address 100c. The representation is the same as used for the general LGP-30 memory, except that the pseudoaccumulator contains one more significant bit.

The program counter is not directly represented in the simulator. Instead, the left-address portion of NAREC location 1001 contains the NAREC address of the order currently being executed rather than that of the next order, as occurs in the LGP-30. Since the address in 1001 is in NAREC form, the method of Appendix B must be used to translate this address to LGP-30 track-sector notation.

The LGP-30 pseudo-instruction-register is located at NAREC address 100d. The representation is the same as used for the general LGP-30 memory and the accumulator.

Three additional registers are maintained in the simulator. The big clock is located at NAREC address 126e and contains a representation of the phase of the single revolution clutch in the output device being simulated. The little clock is located at NAREC address 1269 and contains a representation of the phase of the single revolution clutch in the Flexowriter reader. These registers are necessary for proper simulation of input and output orders. A copy of the control word is maintained at NAREC address 100e.

Input-Output Codes

It should be realized that the LGP-30 relationship between accumulator bits and tape holes differs from that of NAREC. If Friden notation is followed and the bits on punched tape are labelled 6,1,2,3,4,5; the NAREC read order places bits 6,1,2,3,4,5 in that order in the least-significant portion of the U register. In 6-bit configuration the LGP-30 input order stores bits 1,2,3,4,5,6 in that order in the least-significant portion of the pseudo-accumulator after accomplishing a 6-bit left shift; in 4-bit configuration bits 1,2,3,4 are stored in the least-significant portion of the pseudoaccumulator after accomplishing a 4-bit left shift.

A similar problem occurs in output. The NAREC PCU (i.e., print character) order takes the six least-significant bits of the U register and punches them on channels 6,1,2,3,4,5, in that order; the LGP-30 takes six bits from the track address of a P order of the form 1,2,3,4,5,6 and rearranges and punches them on channels 6,1,2,3,4,5.

Since the simulator must read and produce LGP-30 tapes, and since the bit arrangement in the pseudoaccumulator must parallel that of the LGP-30 accumulator, the simulator must accomplish the proper permutations just subsequent to reading and just prior to punching.

Simulator Initialization

Figures E1 and E2 show those subroutines within the simulator that are invoked to communicate information to the simulator concerning the LGP-30 configuration being simulated and to handle the question of pre-existing memory. (Figures E1 through E22 are located in Appendix E.)

Simulation of LGP-30 Sequencing

Figures E3 through E6 show those subroutines which are used to simulate the wired-in sequencing routine of the LGP-30 and certain mechanical variations.

Flow of Control — Figure E3 illustrates the general flow of control. Order sequencing starts at 1000(R) — i.e., the right of location 1000 — which may be reached by the completion of the simulation of a previous order, by the simulated inverse N procedure, or by the execution of the command RO 1000 by the NAREC operator simulating the activation of the start button on the LGP-30.

Since NAREC location 1001 (the pseudo-program-counter) contains the address of the last order executed, it is necessary to increment this location before simulating the next order. Whether this will be done immediately will depend on the contents of 1000(R), which is set by the control word procedure. If the trace is not specified, 1000(R) contains this increment order, and control passes normally to 1001. If the trace is specified, 1000(R) contains a switch to the trace procedure (LO 1498) which prints out information pertaining to the previous order executed, increments location 1001(L), and exits to 1001(L).

At 1000(L) an order classification procedure is initiated which fills the pseudo-instruction-register, shifts the operand address to NAREC right-address position, and transfers control to the proper order simulation subroutine.

The order simulation subroutines normally exit to a clock switch, although the A, S, and D orders may exit to an overflow routine, which causes a printout of the order causing overflow and terminates the simulation. The clock switch states whether any input

or output devices would be operating on the LGP-30 being simulated. If this is the case, a timing routine calculates the time required for execution of the order just simulated and adds this time to the clocks that are running, thus correcting the simulated phases of the single-revolution clutches, and then exits to 1000(R). If no clocks are running, control passes directly to 1000(R).

The P and I orders exit directly to 1000(R) as any necessary clock adjustment is done within the order subroutine. The I order may, however, cause the simulation to cease when an input character is read that would have jammed the LGP-30 Flexowriter.

The Z order may or may not cause simulation to cease, as in the LGP-30, depending on the phase of the single-revolution clutch in the selected output device.

Trace — Figure E4 details the operation of the trace. An examination of the order just executed is made. If the order was not a U or an effective T, the location of the order just executed is printed. The binary left address of NAREC location 1001 is translated to LGP-30 decimal track-sector notation. The instruction just executed is next printed. Suitable coding converts the instruction bits into the proper alphabetic mnemonic, and the address into track-sector notation. Finally, the contents of the pseudoaccumulator are printed in LGP-30 hexadecimal notation.

If the order just executed was a U or an effective T, NAREC location 1001 will not contain the location of that order but will instead contain a location one smaller than the operand of the previous instruction. This scheme allows incrementation of that location to initiate sequencing of the proper next order, but makes it impossible to print out the location of the U or effective T order in a trace routine. Therefore, in this case the printout of the first column is replaced by spaces.

Timing Routine — The timing routine, illustrated in Fig. E5, is invoked when it is necessary to determine how long the order just simulated would have taken on an LGP-30.

The time required for a M, N, or D order may be 73 or 137 sector times, depending on whether the operand sector address is optimally located with reference to the sector at which the order is located. For A,B,C,E,H,R,S,Y and ineffective Z orders, the time is 9 or 73 sectors. Ineffective T orders always require 9 sector times. The time required for a transfer of control is the actual number of sectors traversed, plus 64 if the number is less than 4.

It must be remembered that the sectors on the LGP-30 drum are numbered in an interlaced fashion. We will use underlined quantities to represent sector notation and nonunderlined quantities to represent physical sector position or distance. If we adopt the convention that $00 = \underline{00}$, the drum addresses are interlaced so that $\underline{01}$ is located at 9, $\underline{02}$ at 18, or, in general,

$$K = [9\underline{K}]_{\text{modulo } 64}$$

Multiplying this relationship by 7,

$$[7\underline{K}]_{\text{modulo } 64} = [63\underline{K}]_{\text{modulo } 64} = [-\underline{K}]_{\text{modulo } 64} = 64 - \underline{K};$$

Then,

$$\underline{K} = 64 - [7\underline{K}]_{\text{modulo } 64} = [-7\underline{K}]_{\text{modulo } 64}$$

The question of whether an order located at sector address $\underline{\phi}$ and referring to an operand with sector address $\underline{\rho}$ is optimal can be settled by computing the number of sectors δ between $\underline{\rho}$ and $\underline{\phi + 1}$. If this number δ is greater than 8, $\underline{\rho}$ does not lie

between $\underline{\phi}$ and $\underline{\phi + 1}$ and the order cannot be optimal. If δ is less than 8, the order is optimal provided that δ is large enough to allow reading the next order on the same pass as the fetch or store at $\underline{\rho}$ (or the next successive pass in the case of D, M, and N) and small enough to allow processing the operand at $\underline{\rho}$ on the same pass after reading the order at $\underline{\phi}$.

The limits on δ vary with the different orders. For A,B,C,E,H,R,S,Y, and ineffective Z, the optimal values are 2 through 7; for N, 1 through 7; for M, 3 through 7; and for D, 4 through 7.

Since the quantities $\underline{\phi + 1}$ and $\underline{\rho}$ are available only in sector notation, the test is performed indirectly. First, $\Delta = [(\underline{\phi + 1}) - \underline{\rho}]_{\text{modulo } 64}$ is calculated. Then substituting, $\Delta = [-7\delta]_{\text{modulo } 64}$. Now if $0 < \delta \leq 9$, then $-64 < -7\delta < 0$ and $\Delta = [-7\delta]_{\text{modulo } 64} = 64 - 7\delta$. Thus

$$\Delta - 1 = 63 - 7\delta = 7(9 - \delta),$$

which is evenly divisible by 7. If however, $\delta > 9$, let $\delta = 9 + \delta'$ and $0 < \delta' < 55$. Then, $\Delta = [-7\delta]_{\text{modulo } 64} = [-63 - 7\delta']_{\text{modulo } 64} = [1 - 7\delta']_{\text{modulo } 64}$. Thus

$$\Delta - 1 = [-7\delta']_{\text{modulo } 64} = a64 - 7\delta' \quad \text{where } a = \text{entier} \left\{ \frac{7\delta'}{64} \right\} + 1.$$

Whereas 7 is prime, 64 and 7 are relatively prime integers, and $0 < a < 7$; this assures that $\Delta - 1$ is not integrally divisible by 7. Therefore $\Delta - 1$ will be integrally divisible by 7 only if $\delta < 9$, and in this case the quotient will equal $9 - \delta$. Since the maximum permissible δ is 7, the subroutine considers $\Delta - 15$, which will be 0 for $\delta = 7$ and equal to 7β for $\delta < 7$ where $\beta = 7 - \delta$.

The optimal values of β are 0 to 3 for the D order, 0 to 4 for M, 0 to 5 for most of the other orders, and 0 to 6 for N. Thus, in the timing routine $\Delta - 15$ is computed and a quantity related to $-\beta_{\text{max}}$ for the particular order is loaded in a counter. A loop is invoked which tests the sign of $\Delta - 15$ and, if it is positive, decreases $\Delta - 15$ by 7 and increases the counter by 1. This process may be terminated by a negative $\Delta - 15 - 7\beta$, (which demonstrates that δ is too large), by a zero value in the β counter (which demonstrates a δ too small), or by a zero $\Delta - 15 - 7\beta$, with a negative β counter (which demonstrates an optimal order).

In the case of transfers of control, the time required to execute the order is

$$\begin{aligned} t &= [\rho - \phi - 4]_{\text{modulo } 64} + 4 \\ &= [t' - 4]_{\text{modulo } 64} + 4, \end{aligned}$$

where $t' = [\rho - \phi]_{\text{modulo } 64}$.

Although the indirect scheme previously detailed is employed for the other orders, since it avoids the relatively slow NAREC multiplications and divisions, it is nevertheless a good choice because t' is not a required answer and the decision of whether an order is optimal can always be made in seven or less loops. In transfers of control, however, where t' (the distance between any two arbitrary sectors) must be found, the following more general method is employed.

Given

$$\begin{aligned} t' &= [k_2 - k_1]_{\text{modulo } 64} = \left[\left([-7k_2]_{\text{modulo } 64} \right) - \left([-7k_1]_{\text{modulo } 64} \right) \right]_{\text{modulo } 64} \\ &= \left[-7[k_2 - k_1]_{\text{modulo } 64} \right]_{\text{modulo } 64} = [-7t']_{\text{modulo } 64} \\ &= 64C - 7t', \end{aligned}$$

where C is the least integer that makes t' positive. If we divide t' by 7 and denote the integral quotient by A and the remainder by B , then

$$A = D - t', \text{ where } D \text{ is an integer} = \text{entier} \left\{ \frac{64C}{7} \right\},$$

and

$$B = [64C]_{\text{modulo } 7} = 64C - 7D.$$

Then $9B - A = 9(64C) - 63D - D + t' = t' + 64(9C - D)$, and since C and D are integers,

$$[9B - A]_{\text{modulo } 64} = t'.$$

This technique is coded as a subroutine since it is necessary to compute various times during the execution of the P order. It should be noted that (a) B is calculated indirectly since the NAREC division order (order 70) does not yield a remainder, and (b) the subroutine produces $[t' - 2]_{\text{modulo } 64} + 2$ as each phase in the P order requires at least two sector times.

After the order sequencing and execution time is calculated, an examination is made of the clocks. If only the big clock is running, (i.e., if an output device only is in motion) the answer is added to the big clock. If the big clock is still running, control is transferred to 1000(R), but if the clock has run out, a character is printed and the clock switch is turned off before executing the instruction RO 1000.

If the little clock is on, the tape reader on the Flexowriter is operating. The expiration of this clock indicates that the Flexowriter has read a frame of tape. The action to be taken depends on the character contained on that frame and the state of the big clock.

In addition to this generalized timing routine entered by 13 orders under the clock switch control, the P , I , and Z orders within their detailed simulation invoke the general t' routine to calculate times needed for the control of the clocks and input-output devices.

Overflow Routine — Computation in an LGP-30 ceases when an overflow occurs during the execution of an A , D , or S order. Overflow in NAREC does not cause computation to cease, so in the detailed simulation of each of these orders an examination is made of the pseudoaccumulator and the operand to determine whether overflow would have taken place. If this is the case, these order simulation routines exit to an overflow routine shown in Fig. E6, which prints a note stating that overflow has occurred, the location of the order that caused overflow, the order itself, and the contents of the accumulator before the order is executed. Actually, the order is not executed, and simulation ceases with the memory and pseudoaccumulator representing the state of the LGP-30 before execution of the order that would have caused overflow.

Simulation of LGP-30 Orders

The sequencing routine transfers control to one of sixteen subroutines, one for each LGP-30 order. Upon completion of the subroutine, the simulated drum and the pseudoaccumulator will contain the same information as in the real LGP-30, but there may be variations in the contents of the pseudo-program-counter and in the state of input-output devices and their clocks, as described previously, and in the subroutine descriptions which will now be described.

Z Order (Fig. E7) — The simulator, just as the LGP-30, ignores a Z order if the track bits agree with a depressed breakpoint button on the console (or in the case of the simulator, agreement is checked with bits 1,2,3, and 4 of the control word). If there is a

corresponding breakpoint down, control is switched to the master clock switch; however, if after checking this possibility, no agreement is found, the simulator cannot automatically distinguish between a true stop or a print delay. The LGP-30 is wired so that if the Z order is in fact a delay-after-print, computation ceases until a mechanical device in the output configuration (Flexowriter or fast punch) sends a start signal back to the computer saying that it has finished its output work and that it is safe to continue computation operations without risk of missing the execution of a future output command. The simulator fills this mechanical gap by checking on the program-wound big clock (as distinguished from the little clock which keeps information for the I order) initiated in the P order subroutine. If the big clock is running, it is turned off and the character which is in the printer is recorded; if adding of this detection time makes the clock run out, the character is printed and computation ceases. If the little clock is found running and computation has not yet stopped, control is switched to the timing routine (RO 1200), but if the little clock is not running, the master clock switch is turned off and the next LGP-30 order is processed. If the big clock was not running when the Z order was encountered, computation is stopped. The program may be restarted to pick up the next LGP-30 order after an effective Z by the NAREC operator executing a RO 1000 instruction. (An effective Z will not be traced if that option is used.)

B Order (Fig. E8) — The contents of the pseudoaccumulator are replaced by the B operand and control is transferred to the clock switch.

Y Order (Fig. E9) — The address portion of the pseudoaccumulator is stored in the address portion of memory addressed by the order, and control is transferred to the clock switch. The contents of the pseudoaccumulator and all other parts of memory remain unaltered.

R Order (Fig. E10) — Add two to the pseudo-program-counter, store this as the address portion of the operand of the R order, and go to the clock switch. The simulator adds two, whereas the LGP-30 adds one because in the LGP-30 this counter always reflects the address of the next instruction to be executed rather than the present one.

I Order (Fig. E11) — The most-significant bits of the I-order address are placed in the lowest significant bits of the pseudoaccumulator. The number of bits thus transferred is 4 or 6, depending on the desired input mode. Various paths are then followed, depending on the type of input-output equipment needed for execution of the particular problem.

The big clock is checked. If the clock is running, the order is read and execution times are added. If it is now off, the character from the printer is recorded and the clock switch is turned off; if it is still running, the switch is left on and the I-order analysis proceeds. If the clock is off initially, proceed with the I analysis.

When Flexowriter reader input is used, the little clock, started by the P00xx order, must still be on.* If this specification is not met, an error stop occurs; if it is met, the clock is turned off, and the clock switch is turned off if the big clock is off. If manual input through the Flexowriter is needed, then the little clock is not turned on, but the light flag must be set to the ON position.

After the Flexowriter reader input coding has been proven to satisfy the above condition, the technique of input word analysis merges with that of the fast reader input.

*The little clock could have run out even though it was turned on, and if the character read by its running out was a stop or pseudostop code, the clock remains off, thereby causing a stop when this test is made. This simulates the stopping of the cam on the Flexowriter and thereby stops the line of communication. If the character read when the clock ran out was not one of these special characters, the clock is reset and the I-order test condition is met, but that character is lost.

A single character is read through the NAREC reader and is then checked for the type of code it represents, i.e., a normal code or one of the irregular codes shown in Table 3. If it is a stop code, or some code that looks like a stop code on the LGP-30 input device, the read process is abandoned, the accumulated word is placed in its proper position in the pseudoaccumulator, and control is transferred to the order analysis, which calls for the next chore to be done. If the character is a tape feed, code delete, or some other character generally ignored by the input device, the accumulated word remains the same and another character is read and inspected. If a character would jam the real LGP-30 device, a stop also occurs in the simulated LGP-30. When the character is found to be normal, it is trimmed and/or rearranged according to the 4/6-bit input switch designation and placed at the right side of the accumulated word, and then the whole process of character read is repeated. The effect here is the same as in the real LGP-30. If a short word (less than 8 characters for 4-bit input or 6 characters for 6-bit input) is read and the accumulator has not been previously cleared, the new word will be to the right of the old, displacing an equal number of bits from the left of the old word.

Manual Flexowriter input is somewhat more complicated and, therefore, its use is highly discouraged. All six bits of each character must be placed on the NAREC keyboard in tape code, and the simulator program will rearrange and/or extract a part of each character depending on the 4/6-bit input switch (as designated in the control word) and pack it into the new accumulated word. The NAREC keyboard must be filled from the left-hand side and include the four NAREC bits making up the sign tetrad in order to hold the maximum 8-character LGP-30 word. If 6-bit input is used, the last character must always be a stop code. If a 4-bit input of less than 8 characters is used, a stop code must follow the last information character. When the new word is completely assembled, it is stowed in the pseudoaccumulator and control is returned to the analysis of the next order (RO 1000). Invalid characters, tape feeds, and code deletes are not detected when manual Flexowriter input is used. All information necessary for this type of input is detailed in Appendix C.

D Order (Fig. E12) - Before this order is executed, a check is made on the numerator and denominator. If the magnitude of the numerator is larger or equal to that of the denominator, control is switched to the overflow routine. If in the initial test the denominator is found to be the larger, the contents of the pseudoaccumulator are divided by the

Table 3
Irregular Input Codes

Tape Code	Dec. Code	How Achieved	Output Tape	Flex Input Accum.		Photo Input Accum.	
				4-bit	6-bit	4-bit	6-bit
110101	43	f and z	none	none	none	1010	101011
110111	47	g and z	none	none	none	1011	101111
111001	51	j and z	none	none	none	1100	110011
111011	55	k and z	none	none	none	1101	110111
111101	59	q and z	none	none	none	1110	111011
010100	40	uc and stop	none	start	start	start	start
011000	48	cr and stop	jam	jam	jam	start	start
010110	44	cs and stop	jam	start	start	start	start
011010	52	bs and stop	jam	jam	jam	start	start
001110	28	cs and cr	jam	jam	jam	ignore	ignore
010010	36	lc and stop	jam	start	start	start	start
011100	56	tab and stop	jam	jam	jam	start	start
011110	60	cr,lc, and stop	jam	jam	jam	start	start

D operand and the algebraic quotient is placed in the pseudoaccumulator. Here the programmer has the option of stowing 44 bits plus the sign bit if he so designates in his control word. Program control is then transferred to the clock switch.

N Order (Fig. E13) — The contents of the pseudoaccumulator are multiplied by the N operand. This produces a 62-bit product (or 89-bit product if full precision is used), bit 0 of which is the sign bit. Bits 31-61 of the resultant product are filled into bits 0-31 of the pseudoaccumulator (or bits 31-79 into 0-44 for extended precision). Note that, as in the LGP-30, the sign of the 0th bit of the pseudoaccumulator now merely indicates the most significant magnitude bit of the resultant portion used and is no indication of the sign of the word, although other arithmetic orders will interpret this as a sign bit in further computation. Control is then transferred to the clock switch.

M Order (Fig. E14) — Multiply as in the N order, but place bits 0-31 of the same 62-bit product in bits 0-31 of the pseudoaccumulator (or sign bit and 44 information bits for extended precision). Control is then transferred to the clock switch.

P Order (Fig. E15) — This order is more complex than the others because most of the work must be done in the simulator program on NAREC whereas it would normally be done by the hardware on the actual LGP-30. First, for all types of P order, the simulator program sets the track bits of the order in a separate NAREC memory location for easy accessibility. Then branches radiate according to the condition of the clock switch and the type of input-output devices used.

In general, if the big clock runs out, a character is printed from the printer. It is to be noted here that an output character is not printed immediately when a Pnxxx order is encountered. Instead, upon detection of the effective P order, the character is stowed in a pseudoprinter (NAREC location 126d). It is this character, or this character combined through the logical summation with a character caused by an intervening Pnxxx order when the big clock was still running, that is printed either when the clock runs out or upon the encountering of a Z order. The little clock running out causes a character to be pulled from the input tape, the character thus pulled ignored, and the clock reset to its initial value of minus-six drum revolutions, unless the character read is a stop code or pseudostop code. When such a character is read, the clock remains off, simulating the effect of a like stop motion in the Flexowriter reader mechanism when it detects a stop code.

With these things in mind we now detail each of the individual branches that a P order might take, including how each of the clocks is initialized for a given case of circumstances and what happens if they should run out.

No Clock Is On At Present

A. Input P Order (P00xx)

- a. Photoreader input: Go to the next LGP-30 order (RO 1000).
- b. Flexowriter reader input: Turn on the clock switch, set the little clock to minus-six drum revolutions plus execution time, then go to the next LGP-30 order (RO 1000).
- c. Flexowriter manual input: Turn on the Flexowriter light flag, then go to the next LGP-30 order (RO 1000).

B. Output P Order (Pnnxx): Turn on the clock switch, then if:

a. Fast punch output: Set the big clock to minus-3-1/3 drum revolutions plus execution time, put the character to be output in the printer, then go to the next LGP-30 order (RO 1000).

b. Flexowriter output: If the character is a tab or carriage return, set the big clock to minus-60 drum revolutions, but if it is any other character, set the big clock to minus-7 drum revolutions. Put this character to be printed in the printer, add the execution time for the order to the big clock, and then go to the next LGP-30 order (RO 1000).

One or More Clocks Are Now On

A. Flexowriter input and output with input copy deleted (with this combination it is impossible for both clocks to be on at the same time)

a. The little clock is running: This means that a P00xx has started the clock, no I order has intervened, and no stop code or pseudostop code has been pulled, if the clock ran out, to keep it off permanently—i.e., the Flexowriter has been set into motion and no I order has taken place to process any input information. From this point one of two branches may be taken:

1. If adding the time it took to read the order makes the clock run out, a character is pulled from the input tape. If this is not a stop or pseudostop code, it is ignored and the clock set to minus-6 drum revolutions. The current P order is thereby ignored, as in the case of the actual LGP-30 installation, because the Flexowriter is all ready handling a job. If the character read is indeed a stop code, Flexowriter motion is stopped. The current P order is then analyzed and handled in the same manner as if no clock were on when this order was first encountered.

2. If adding the read time of the current P order did not turn off the clock, the time that it would have taken to execute the P order is added. If the clock is still running, control is switched to the next LGP-30 order; if this latter addition makes the clock run out, a character is read. When the character is normal it is ignored, the clock is reset to minus-6 drum revolutions, and control is sent to bring the next LGP-30 order (RO 1000); but if it is a stop or pseudostop code, the clock switch is turned off. In both of these cases the character read and the current P order are ignored.

b. The big clock is running:

1. If adding of the read time of the current P order turns the clock off, the character from the printer is recorded on the output tape, and the current P order is treated as if no clock were on when it was first encountered.

2. The big clock still on after adding the read time causes control to branch according to the type of P order. If the current P order is an input type (P00xx), the time that it would have taken to execute the order is added to the clock. If it now runs out, the character from the printer is recorded, the clock switch is turned off, and control is switched to the next LGP-30 order; if it does not run out with this addition, the next LGP-30 order is analyzed (RO 1000). If the current P order concerns output (Pnnxx), the time in the clock is studied. The result from this study either causes a decision of a logical summation* of the current P-order character and the last, or the last character

*Logical summation occurs if less than two drum revolutions have passed since the last effective P order for a normal character and if less than 10 drum revolutions have passed in the case for a tab or carriage return when the mode of Flexowriter output is used.

is printed and the current one is placed in the printer and the current P-order's address recorded in a NAREC memory location to be printed out with a note upon the first LGP-30 effective Z order encountered. The execution time of the current P order is then added to the clock. If the clock now runs out, the contents of the printer are recorded, the clock switch is turned off, and the next LGP-30 order is called; if it does not run out, control is sent to the next LGP-30 order (RO 1000).

B. Flexowriter input and output with input copied: This is the same as the combination above, with the only exception of the reading of a character when the little clock runs out. Here, when a character is read it is also copied on the output tape.

C. Photoreader input and fast punch output (with this combination of input-output devices, only the big clock can be running): Add the read time of the P order to the big clock. If the clock is now off, print the character from the printer. The current P order is now processed. If it is a P00xx, control is switched to the next LGP-30 order (RO 1000) after first turning off the clock switch; if the current P order is a Pnxxx, set the big clock to minus-3-1/3 drum revolutions plus the execution time of the order, place the character to be printed into the printer, and go to the next LGP-30 order (RO 1000). If the big clock is still running after adding the read time of the current P order and if this order is a P00xx, add the execution time to the clock; if it is still on, go to the next LGP-30 order (RO 1000), but if it is now off, print the character in the printer, turn off the clock switch, and then go to the next LGP-30 order (RO 1000). If the big clock is still running and the current P order is a Pnxxx, the amount of time in the clock is analyzed. This study results in either a logical summation* of the current P-order character and the last, or the last character is printed and the current one is placed in the printer with the address of the current P order placed in NAREC memory to be printed upon the execution of the first effective Z order. Control is then switched to the next LGP-30 order (RO 1000).

D. Photoreader input and Flexowriter output (with this combination only the big clock may be running)

a. Adding the read time makes the big clock run out: Print the character in the printer. If the current P order is of the form Pnxxx, set the big clock to the proper time, i.e., minus-7 drum revolutions for a normal character or minus-60 drum revolutions for a carriage return or tab, and add the execution time of the P order. Place the character to be printed into the printer and go to the next LGP-30 order (RO 1000). If the current P order is of the form P00xx, turn off the clock switch and then go to the next LGP-30 order (RO 1000).

b. The big clock still on after adding the read time: If this is an input order (P00xx), the execution time of the order is added to the big clock. If it is still running, control is transferred to the next LGP-30 order (RO 1000); if the clock is now off, print the character from the printer, turn off the clock switch, and then go to the next LGP-30 order (RO 1000). If this P order is of the form Pnxxx, check the clock to see if it is in the logical summation area.† If it is, take the logical sum of this character and the one in the printer and place the result in the printer; if not, print the character from the printer, place the one from the current P order in the printer, and place the address of this order in a NAREC memory location to be printed upon execution of the next effective Z order. Then go to the next LGP-30 order (RO 1000).

*Logical summation occurs if less than 29 sectors have passed since the last effective Pnxxx order.

†Logical summation occurs if less than 2 drum revolutions have passed since the last effective Pnxxx order for a normal character or 10 drum revolutions for a tab or carriage return.

E. Flexowriter input and fast punch output: (both clocks may be on)

a. The little clock is not on: Add the read time to the big clock; if it is still on, treat this order in the same manner as if this condition were present with Photoreader input and fast punch output; if it is now off, print the character from the printer and test for the type of P order. If it is an input P00xx order, set the little clock to minus-6 drum revolutions plus execution time, and go to the next LGP-30 order (RO 1000); if it is an output (Pnxxx) order, set the big clock to minus-3-1/3 drum revolutions plus execution time, place the character in the printer, and go to the next LGP-30 order (RO 1000).

b. The little clock is on: Add the read time.

1. If the clock is now off, read one frame of tape. If the character read is not a stop or pseudostop code, reset the little clock to minus-6 drum revolutions plus the execution time of the order, and go to section E.b.2., 3rd sentence, for explanation of the handling of the big clock; if the character read is a stop or pseudostop code, leave the clock off and check to see if the big clock is on. If it is indeed running, treat this order in the same manner as if Photoreader input and fast punch output were used; if the clock is off, however, check the type of P order. If it is a P00xx order, set the little clock to minus-6 drum revolutions plus the execution time of the order, and switch control to the next LGP-30 order; if it is a Pnxxx order, set the big clock to minus-3-1/3 drum revolutions plus the execution time, place the character in the printer, and go to the next LGP-30 order (RO 1000).

2. If the little clock is still running after the addition of the read time, add the execution time. If the clock is now off, go to E.b.1. for further explanation. If the clock is still on, check the big clock. If it is not on and the current P order is of the form P00xx, go to the next LGP-30 order (RO 1000) because the Flexowriter is already handling a job; if this P order is of the form Pnxxx, set the big clock to minus-3-1/3 drum revolutions plus execution time, set the new character in the printer, and transfer control to the next LGP-30 order (RO 1000). When the big clock is on, add the read time of the current P order and check to see if it is still on. If it is not, print the character from the printer and check for the type of the current P order. If it is of the form P00xx, go to the next LGP-30 order (RO 1000); if the order is of the form Pnxxx, set the big clock to minus-3-1/3 drum revolutions plus the execution time, place the new character in the printer, and transfer control to the next LGP-30 order (RO 1000). If adding the read time to the big clock does not turn it off, the type of order is examined. If it is a P00xx order, the execution time is added to the big clock. When it is still on, control transfers to the next LGP-30 order, but if it is turned off by the addition, the character in the printer is recorded and then control is transferred to the next LGP-30 order (RO 1000). If we detect a Pnxxx order with the big clock still running after adding the read time, we check for the possibility of a logical summation.* The decision is then made whether to logically sum this character with the one in the printer and stow the result in the printer, or to print out the old character, stow the new one in the printer, and stow the location of this latter P order to be printed out upon execution of the next effective Z order. The execution time of this P order is then added to the clock. If it is still on, transfer control to the next LGP-30 order (RO 1000); if not, print the character in the printer and then go to the next LGP-30 order (RO 1000).

E Order (Fig. E16) — Mask the contents of the pseudoaccumulator by the E operand, i.e., keep all the bits in the pseudoaccumulator that match those in the E operand and delete all others. Then go to the clock switch (LO 1200).

*Logical summation occurs if less than 29 sectors have passed since the last effective Pnxxx order.

U Order (Fig. E17) — Place the operand address of the U order, minus one, into the pseudoprogram counter and proceed to simulate the LGP-30 sequentially from there. The reason that a one must be subtracted is that before executing the next order the present U order must be timed if any clock is found running, and this timing routine automatically increments the program counter, i.e., if we did not subtract the one, the next LGP-30 order executed would be the one in the U operand address plus one location. The address of this order is then stowed with a minus sign in NAREC memory to tell the timing routine that the current order to be timed was a transfer of control order. Then control is transferred to the clock switch (LO 1200).

T Order (Fig. E18) — If this order is detected and a bit is found in the sign position of the pseudoaccumulator, or a bit in the sign position of the pseudoprogram register with a bit in the fifth position of the control word, proceed with analysis as in the U order; otherwise ignore this order. Then go to the clock switch (LO 1200).

H Order (Fig. E19) — Store the contents of the pseudoaccumulator in the pseudo-LGP-30 memory location designated by the H address. Normally, 31 bits are thus stored, but if the programmer wishes he may store all 45 bits available into the memory location by placing a bit in the sign position of the control word. The contents of the pseudoaccumulator are unaltered by this operation. Then control is transferred to the clock switch (LO 1200).

C Order (Fig. E20) — Store the contents of the pseudoaccumulator in the pseudo-LGP-30 memory location designated by the H address and set the pseudoaccumulator to zero. Normally, 31 bits are thus stored, but if the programmer wishes he may store all 45 bits available into the memory location by placing a bit in the sign position of the control word. Then go to the clock switch (LO 1200).

A Order (Fig. E21) — If the absolute sum of the A operand and the pseudoaccumulator is greater than or equal to one, the simulator transfers control to the overflow routine. If there is no overflow, the A operand is added to the contents of the pseudoaccumulator, the result is stowed in the pseudoaccumulator, and control is transferred to the clock switch (LO 1200).

S Order (Fig. E22) — When the signs of the two quantities involved are not identical, compute the sum of their absolute values. If this results in a number greater than or equal to one, the simulator transfers control to the overflow routine. If the signs of the two quantities are identical, or if they are not but no overflow can occur, the S operand is subtracted from the contents of the pseudoaccumulator, the result is placed in the pseudoaccumulator, and control is transferred to the clock switch (LO 1200).

CONCLUSIONS

It has been found possible to construct a NAREC program that permits the execution of LGP-30 machine-language programs at a greater speed than on the actual LGP-30.

While this program was written to allow the use of programs written and debugged on the LGP-30 by an organization now lacking the active use of this computer, other organizations having access to NAREC may find it desirable to use this system.

The factors tending to favor the use of the system include:

1. Problems monopolizing an LGP-30 and preventing its use in its legitimate role in close-support guidance of research
2. Desire to use the LGP-30 operator-programmer in more productive tasks

3. Programs which are computation-limited and could therefore run much faster on NAREC

4. Breakdown of the LGP-30

5. Programs that are so complex that it is necessary to secure additional LGP-30 computers or pay overtime charges

6. Removal of an LGP-30 computer from a site where great programming effort exists for that computer.

The factors tending to discourage the use of this system would include:

1. Incompletely debugged programs

2. Input-output-limited programs which would not run very much faster on NAREC when the LGP-30 is not being used for other work

3. Short programs which are run concurrently with the performance of experiments and which continuously guide the course of the experiment.

In recent months the simulator program has been actively used on a variety of programs with extremely satisfactory results, both to the Radio Division and to several other groups within the Laboratory. These programs include simulations of a major two-computer system, an Act I compilation and subsequent computation, a research project of the Sound Division, and various short-run problems. Over 300 hours of NAREC computer time have been used to date.

ACKNOWLEDGMENTS

The authors wish to express their appreciation to Mr. Ron Bachelor of the Royal McBee Corporation for his aid in defining timing characteristics within the hardware of the LGP-30 and to the entire staff of the Research Computer Center at NRL for their patience and cooperation during the programming and debugging of the simulator program.

Appendix A

SIMULATOR EFFICIENCY

The simulation ratio, or the speed gain of a simulation, is dependent on the time required for the execution of a particular code on both the LGP-30 and the simulator. The first quantity is determined by the coding and is calculable, provided latency is taken into account, but the second depends on a number of variables. NAREC is an asynchronous machine, and the time required to execute a particular NAREC order is determined by activator timings that are subject to month-to-month adjustments and relations between the time of occurrence of completion signals and the phase of the core memory. Furthermore, even if activator times are held constant, the execution times of various NAREC orders, particularly multiplication, are variable and dependent on the bit structure of the NAREC operand. Finally, the path taken by the simulator itself may vary with the LGP-30 operand.

Nevertheless a method of predicting simulation ratio is desirable in analyzing whether or not it is worthwhile to simulate a particular program. Table A1 presents information which reflects simulation speed for typical operands with NAREC running at April-1962 activator times. These activator times, while not the fastest that have ever been maintained, probably represent the fastest times that can be safely predicted to be maintained in the future. LGP-30 speeds and simulation ratios are calculated on the basis of 3600-rpm drum rotation. The quantities in parentheses in Table A1 are sensitive to the bit structure of the operand.

The greatest simulation ratios occur for nonoptimal orders and for multiplication orders. Thus a program could exhibit a simulation ratio of as much as 35, while a program consisting of optimal short orders could exhibit a simulation ratio of as little as 3.5. In an attempt to illustrate realistic cases, Table A2 was calculated under the assumption that the program executed an equal number of M, N, and D orders, and a suitable mix of short orders with an average simulation time of 635 μ sec. The entry parameters in Table A2 are the multiplication load, which is defined as the number of M, N, and D executions divided by the total number of order executions, and the fraction of the executions which are optimal, assuming that this fraction is the same for both long and short orders.

It may be seen that for 10-percent optimization (which would occur accidentally if no attempt at optimization were made) the simulation ratio is about 26. As the optimization increases, the simulation ratio decreases and shows sensitivity to multiplication load, decreasing for decreasing multiplication load.

In practice, programs tend to consist of combinations of highly optimized standard subroutines and unoptimized special coding. The Royal-McBee Sine Subroutine 14.0, in which all orders except transfers of control are optimal, exhibits a simulation ratio of 9. A typical program might spend half its time in such highly optimized subroutines and half its time in nonoptimal private coding. Note that this assumption implies that most of the orders executed are in the subroutines, as the subroutines execute more orders per second than the private coding. In this case the simulation ratio would be $1/[(0.5/9) + (0.5/26)] = 12$. Note that the ratios 9 and 26 cannot be simply averaged, as this would imply that an equal number of orders were executed in the subroutines and the private coding.

Table A1
Simulation Speed for LGP-30 Orders

Order	Simulator Time (μsec)	Nonoptimal LGP-30 Order		Optimal LGP-30 Order	
		Time (μsec)	Simulation Ratio	Time (μsec)	Simulation Ratio
B	533	19010	35.7	2344	4.4
Y	724	19010	26.3	2344	3.2
R	876	19010	21.7	2344	2.7
D	1352	35677	26.4	19010	14.1
N	(1605)	35677	(22.2)	19010	(11.8)
M	(1049)	35677	(34.0)	19010	(18.1)
E	549	19010	34.6	2344	4.3
U*	838	17448	20.8	1042	1.2
T†	498	2344	4.7	2344	4.7
H	567	19010	33.5	2344	4.1
C	662	19010	28.7	2344	3.5
A	(604)	19010	31.5	2344	3.9
S	(723)	19010	26.3	2344	3.2

*Or effective T order. The times stated represent the maximum and minimum possible times; intermediate values are more likely.

†Ineffective T order.

Note: Times and ratios in parentheses are subject to variation.

Table A2
Simulation Ratio as a Function of Multiplication Load and Degree of Optimization

Multiplication Load	Simulation Ratio for Various Ratios of Optimal Executions to Total Executions				
	0.1	0.3	0.5	0.7	0.9
0.0	27.3	22.1	16.8	11.6	6.3
0.2	26.7	22.4	18.1	13.8	9.5
0.4	26.2	22.6	19.0	15.3	11.7
0.6	25.9	22.8	19.6	16.4	13.3

The simulation ratio in input-output is a complicated matter, dependent on the input-output subroutines being simulated, the devices being simulated, and interrelations between the subroutine, the simulator coding, and the NAREC input-output devices. In general, a routine that does not slow down the LGP-30 photoreader will exhibit a simulation ratio of a little better than 2 when reading tape, while a Flexowriter input process will show a substantial simulation ratio. Simulator output generally proceeds at about 40 frames per second if the program would have been punch- or Flexowriter-limited on the LGP-30, representing simulation ratios of 2 and 4, respectively. In practice, output routines utilizing the LGP-30 punch do not use it with maximum efficiency and tend to exhibit simulation ratios of nearly 4. In general, simulation ratios are less for input-output operations than for calculations, and the overall simulation ratio decreases as input-output load increases. Table A3 is an attempt to predict simulation ratio for various combinations of subroutine utilization and input-output loading. For example, the simulation of ACT I compilations, which are highly optimized and have considerable input-output, exhibits a simulation ratio of 7.4 when the second printing is omitted and 6.1 when this printing is included.

Table A3
Simulation Ratio as a Function of Input-Output Load and the Degree of Utilization of Highly Optimized Subroutines

Input-Output Load	Simulation Ratio for Various Degrees of Utilization of Highly Optimized Subroutines		
	Negligible	Moderate	High
Negligible	26	12	9
Moderate	10	8	5
High	6	5	4

An extension, currently under way, is to translate common Royal McBee subroutines into NAREC language. With suitable modifications to the sequencing simulation, these subroutines could be directly executed in NAREC language rather than by simulation. Preliminary studies indicated that high effective simulation ratios would be possible through this technique.

At the time of this writing this scheme has been implemented for a limited number of subroutines. This implementation is contained in NAREC tape No. 4321A only. If this tape is used and the programmer does not desire to use this feature, the right-address position of the control word must contain zero. If this position contains some nonzero number n , the simulator expects to receive n subsequent subsidiary control words. Each of these words is placed on the switchbox in the form XXXX YY YY00 00, where XXXX is the commonly used identification number of the LGP-30 routine and YYYY is the decimal track-sector notation of the first location of that subroutine. The subroutines, identification numbers, and simulation ratios are listed in Table A4.

The technique used, shown in Figs. A1 and A2, is to combine information known to the simulator about the subroutine together with the specification of its starting location to generate a table of operand addresses for R orders, which may be the beginning of a subroutine call, and the operand addresses for U orders following these R orders which constitute the call. It should be noted that more than one R order is possible for some subroutines and that, for a given R address, more than one U address is meaningful.

Table A4
 Identification, Relative Execution Time, and Simulation Ratio of Subroutines

LGP-30 Name	Identi- fication (XXXX)	Relative Ex. Time			Simulation Ratio	
		LGP-30	Normal Sim.	Direct Sim.	Normal	Direct
Sine-Cosine 14.0	0140	33.4	3.71	1.00	9.0	33.4
Log 18.0	0180	81.8	13.2	1.00	6.2	81.8
Square Root 15.1	0151	281	36.9	1.00	7.6	281
Arc Tan 16.0	0160	38.4	3.4	1.00	11.3	38.4
Arc Sin - Arc Cos 20.0	0200	533	74.8	1.00	7.1	533
Exponentiation 17.0	0170	25.9	2.64	1.00	9.8	25.9

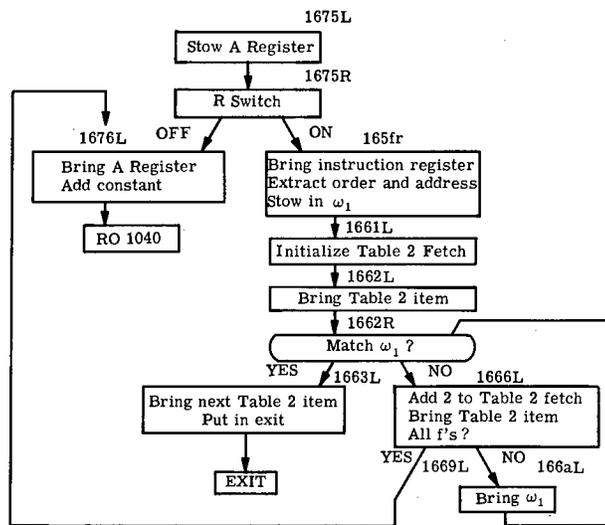


Fig. A1 - Extension of R order code sheets and flowchart

In the simulation of each R order the operand address is compared against the table. If the address corresponds to a table entry, the next order is examined. If it is not a proper subroutine entry, the R and subsequent orders are simulated in normal fashion; if it is a subroutine entry, control is transferred to a procedure which accomplishes the same function as the subroutine.

The following points should be carefully noted:

1. The specification of a direct subroutine simulation does not affect the simulated LGP-30 memory; the subroutine will be at the specified location only if it is placed there in the normal fashion.
2. It is not necessary to place the subroutine in the implied location; indeed, it is possible to use this space on the pseudomemory for other coding or data, though such a program would not properly run on the physical LGP-30.
3. A subroutine directly simulated has no effect on the working storages (internal or track 63) of the subroutine.

Appendix B

TRANSLATION METHODS

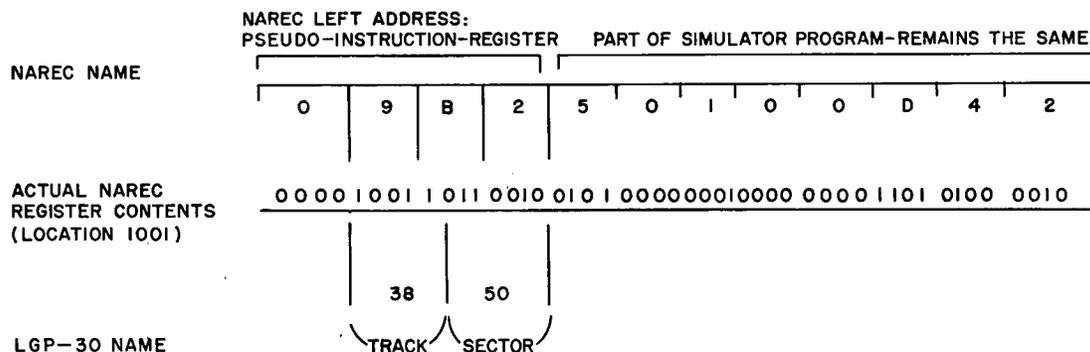
All discussions that follow assume that we are working in 4-bit rather than 6-bit syllables. If 6-bit is needed, a bit-by-bit inspection is necessary.

Translation of the Pseudo-Program-Counter

The simulator program maintains the LGP-30 pseudo-program-counter in the left-address portion (bits -3 through 12) of NAREC location 1001. Bits 1-6 are used to hold the track portion and bits 7-12 are used to hold the sector portion of the LGP-30 address, as shown in Fig. B1. Thus, since the maximum LGP-30 memory location is 6363 or 4096 words, the maximum address that this pseudocounter can hold is Off in hexadecimal.

To take an example, if a program stops and we wish to see the location of the stop, call location 1001 to the NAREC display device and read the contents. If this should read 09b250100d42 in NAREC hexadecimal, as shown in Fig. B1, we take the left 16 bits, break bits 1-12 into two groups, and find that we have the LGP-30 track-sector address of 3850. Conversely, if we wish to start a program with location 3850 and do not wish to transmit this by paper tape or manual keyboard input, the LGP-30 address 3850 can be translated to the NAREC form of the address, 09b2, this number being filled in manually in bits 1-12 of location 1001 by the NAREC operator (who must be sure to reproduce the rest of the word in 1001), and the programmer may then instruct the NAREC operator to execute a LO 1001 instruction on the NAREC console, which will pick up simulation starting with the operand of LGP-30 location 3850.

To save time, the relationship between the LGP-30 track-sector notation and the NAREC notation for these 12 bits is summarized in Table B1. For the example above, the NAREC name of LGP-30 location is found by adding the track notation of 38 from the left column, 0980, and hexadecimally adding the sector notation for 50 from the right column, 0032, to obtain the desired result 09b2. Translation from NAREC notation to



LGP-30 notation is achieved by dividing the address 09b2 into the portions 0980 and 0032 and finding these values to be the LGP-30 notation 38 and 50, respectively, to make up the LGP-30 address 3850.

Table B1
Program-Counter Translation

LGP-30 Track-Sector Notation	NAREC Hexadecimal Notation		LGP-30 Track-Sector Notation	NAREC Hexadecimal Notation	
00	0000	0000	32	0800	0020
01	0040	0001	33	0840	0021
02	0080	0002	34	0880	0022
03	00c0	0003	35	08c0	0023
04	0100	0004	36	0900	0024
05	0140	0005	37	0940	0025
06	0180	0006	38	0980	0026
07	01c0	0007	39	09c0	0027
08	0200	0008	40	0a00	0028
09	0240	0009	41	0a40	0029
10	0280	000a	42	0a80	002a
11	02c0	000b	43	0ac0	002b
12	0300	000c	44	0b00	002c
13	0340	000d	45	0b40	002d
14	0380	000e	46	0b80	002e
15	03c0	000f	47	0bc0	002f
16	0400	0010	48	0c00	0030
17	0440	0011	49	0c40	0031
18	0480	0012	50	0c80	0032
19	04c0	0013	51	0cc0	0033
20	0500	0014	52	0d00	0034
21	0540	0015	53	0d40	0035
22	0580	0016	54	0d80	0036
23	05c0	0017	55	0dc0	0037
24	0600	0018	56	0e00	0038
25	0640	0019	57	0e40	0039
26	0680	001a	58	0e80	003a
27	06c0	001b	59	0ec0	003b
28	0700	001c	60	0f00	003c
29	0740	001d	61	0f40	003d
30	0780	001e	62	0f80	003e
31	07c0	001f	63	0fc0	003f

Translation of the Pseudo-Instruction-Register

The pseudo-instruction-register is kept in NAREC location 100d and reflects the same information as does the actual LGP-30 instruction register, but the names used are different. Figure B2 shows the register with an example instruction. Note that the bits occupy the same actual bit names, i.e., bit 1 of NAREC is the same name as bit 1 of the LGP-30. However, the difference shows up in the way that these bits are grouped to form the hexadecimal name of the same bit-by-bit representation.

If the programmer wishes to represent the order S 4961, he could manually write out the bit-by-bit representation as shown in the figure and then mentally divide the bits into groups of four in the NAREC manner in order to get the NAREC equivalent of the desired word. However, this is a tedious task and a short-cut method seems desirable. If the

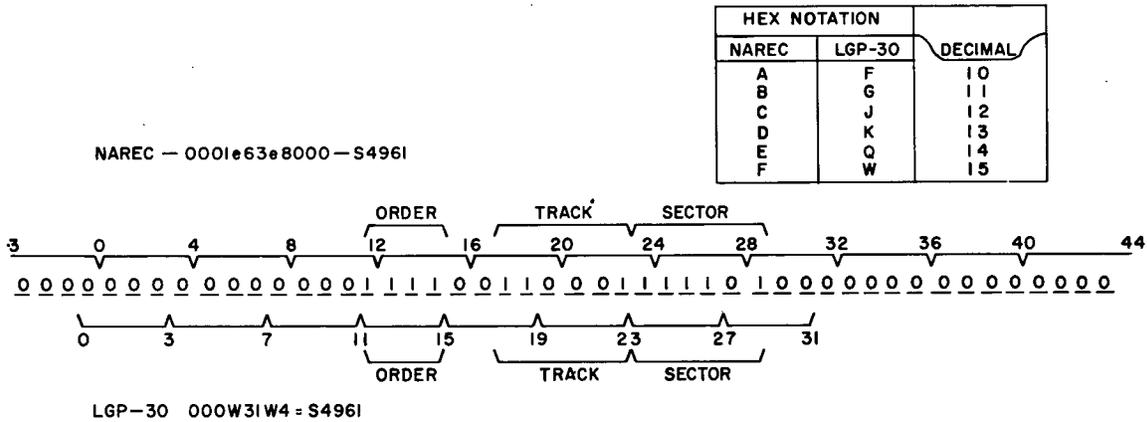


Fig. B2 - Bit placement in the pseudo-instruction-register

desired word to be translated is in the LGP-30 hexadecimal form, for example 000w31w4, which is the equivalent to S 4961, then it appears that if this number were multiplied hexadecimally by two and placed correctly within the NAREC word we would have the correct result. As Fig. B2 shows, the last three NAREC hexadecimal characters must be zero, and a zero must be placed at the left of the word to allow for the sign. Performing the required multiplication for the above example then gives the result

$$\begin{array}{r}
 0[000w\ 31w4]000 \\
 \hline
 2 \\
 \hline
 0\ 001e\ 63e8\ 000 = 0001\ e6\ 3e80\ 00.
 \end{array}$$

If the LGP-30 hexadecimal representation is not available to use for the translation, the programmer may use the second part of Table B2. The NAREC equivalent is here obtained by hexadecimally adding the NAREC name for the individual order code, track, and sector. Thus in the above example, the result would be

$$\begin{array}{r}
 0001\ e0\ 0000\ 00 \\
 0000\ 06\ 2000\ 00 \\
 0000\ 00\ 1e80\ 00 \\
 \hline
 0001\ e6\ 3e80\ 00.
 \end{array}$$

When the LGP-30 instruction contains a counter to the left of the address, the translation may be done only by the bit-by-bit inspection method or by the LGP-30 hexadecimal to NAREC hexadecimal method.

Translation from the contents of NAREC location 100d to its equivalent LGP-30 name is done by breaking the word into the three groups of bits, i.e., order bits 11-15, track bits 17-23, and sector bits 25-29, and looking these up in Table B2 to get the desired result.

Translation of the Pseudoaccumulator

The method used to translate the instruction register for NAREC location 100a is the same as the hexadecimal-to-hexadecimal translation described in the previous section.

Table B2
Pseudo-Instruction-Register Translation for Hex-to-Hex
Notation and for Track-Sector-to-Hex Notation

Hexadecimal Notation		Hexadecimal Notation	
LGP-30 Order	NAREC Code	LGP-30 Order	NAREC Code
Z	0000 00 0000 00	E	0001 20 0000 00
B	0000 20 0000 00	U	0001 40 0000 00
Y	0000 40 0000 00	T	0001 60 0000 00
R	0000 60 0000 00	800 T	1001 60 0000 00
I	0000 80 0000 00	H	0001 80 0000 00
D	0000 a0 0000 00	C	0001 a0 0000 00
N	0000 c0 0000 00	A	0001 c0 0000 00
M	0000 e0 0000 00	S	0001 e0 0000 00
P	0001 00 0000 00		

LGP-30 Track-Sector Notation	NAREC Hexadecimal Notation	
	Equivalent for Track	Equivalent for Sector
00	0000 00 0000 00	0000 00 0000 00
01	0000 00 2000 00	0000 00 0080 00
02	0000 00 4000 00	0000 00 0100 00
03	0000 00 6000 00	0000 00 0180 00
04	0000 00 8000 00	0000 00 0200 00
05	0000 00 a000 00	0000 00 0280 00
06	0000 00 c000 00	0000 00 0300 00
07	0000 00 e000 00	0000 00 0380 00
08	0000 01 0000 00	0000 00 0400 00
09	0000 01 2000 00	0000 00 0480 00
10	0000 01 4000 00	0000 00 0500 00
11	0000 01 6000 00	0000 00 0580 00
12	0000 01 8000 00	0000 00 0600 00
13	0000 01 a000 00	0000 00 0680 00
14	0000 01 c000 00	0000 00 0700 00
15	0000 01 e000 00	0000 00 0780 00
16	0000 02 0000 00	0000 00 0800 00
17	0000 02 2000 00	0000 00 0880 00
18	0000 02 4000 00	0000 00 0900 00
19	0000 02 6000 00	0000 00 0980 00
20	0000 02 8000 00	0000 00 0a00 00
21	0000 02 a000 00	0000 00 0a80 00
22	0000 02 c000 00	0000 00 0b00 00
23	0000 02 e000 00	0000 00 0b80 00
24	0000 03 0000 00	0000 00 0c00 00
25	0000 03 2000 00	0000 00 0c80 00
26	0000 03 4000 00	0000 00 0d00 00
27	0000 03 6000 00	0000 00 0d80 00
28	0000 03 8000 00	0000 00 0e00 00
29	0000 03 a000 00	0000 00 0e80 00
30	0000 03 c000 00	0000 00 0f00 00
31	0000 03 e000 00	0000 00 0f80 00

Table continues

Table B2—Continued

LGP-30 Track-Sector Notation	NAREC Hexadecimal Notation	
	Equivalent for Track	Equivalent for Sector
32	0000 04 0000 00	0000 00 1000 00
33	0000 04 2000 00	0000 00 1080 00
34	0000 04 4000 00	0000 00 1100 00
35	0000 04 6000 00	0000 00 1180 00
36	0000 04 8000 00	0000 00 1200 00
37	0000 04 a000 00	0000 00 1280 00
38	0000 04 c000 00	0000 00 1300 00
39	0000 04 e000 00	0000 00 1380 00
40	0000 05 0000 00	0000 00 1400 00
41	0000 05 2000 00	0000 00 1480 00
42	0000 05 4000 00	0000 00 1500 00
43	0000 05 6000 00	0000 00 1580 00
44	0000 05 8000 00	0000 00 1600 00
45	0000 05 a000 00	0000 00 1680 00
46	0000 05 c000 00	0000 00 1700 00
47	0000 05 e000 00	0000 00 1780 00
48	0000 06 0000 00	0000 00 1800 00
49	0000 06 2000 00	0000 00 1880 00
50	0000 06 4000 00	0000 00 1900 00
51	0000 06 6000 00	0000 00 1980 00
52	0000 06 8000 00	0000 00 1a00 00
53	0000 06 a000 00	0000 00 1a80 00
54	0000 06 c000 00	0000 00 1b00 00
55	0000 06 e000 00	0000 00 1b80 00
56	0000 07 0000 00	0000 00 1c00 00
57	0000 07 2000 00	0000 00 1c80 00
58	0000 07 4000 00	0000 00 1d00 00
59	0000 07 6000 00	0000 00 1d80 00
60	0000 07 8000 00	0000 00 1e00 00
61	0000 07 a000 00	0000 00 1e80 00
62	0000 07 c000 00	0000 00 1f00 00
63	0000 07 e000 00	0000 00 1f80 00

The LGP-30 hexadecimal word is multiplied by two, and the proper number of zeros is added. Two more examples are here provided for the programmer.

$$\frac{0[\text{wwwq} \text{ wwwq}]000}{2} = 1\text{fff ff ffc} 000 = 1\text{fff ff ffc} 00, \quad \frac{0[\text{j679 24w1}]000}{2} = 1\text{8cf2 49e2} 000 = 1\text{8cf 24 9e20} 00.$$

Translation of Bootstrap

In normal practice, the PIR 10.4 is placed into the LGP-30 pseudo-drum-memory by the execution of the instruction LO 1068 by the NAREC operator. As this subroutine has already been stored in NAREC memory in a transliterated form there is no need for use of the bootstrap program. All that need be done is to transfer the transformed version of PIR into its proper locations. However, if the programmer wishes to use another type

of program input routine that is written in LGP-30 language, he must first translate a bootstrap program into NAREC language.

When an LGP-30 operator uses the "manual input--fill instruction--execute" push-button sequence to load a bootstrap, he is loading by use of wired functions independent of input routines. In simulation, a bootstrap is also loaded by a wired function, i.e., the NAREC 32 order. The essential difference is that the bootstrap must be translated into NAREC language (as just described in the section on translation of the instruction register) and prepared on a NAREC Flexowriter. The translation of the normally used bootstrap for PIR 10.4 is given in Table B3.

To load the PIR into the pseudo-drum-memory, the beginning location of the bootstrap routine is translated by the methods just described in the section entitled translation of the program counter. This number is then loaded into the program counter (left address of NAREC location 1001) and also given to the NAREC operator with the instructions to read the translated bootstrap tape into that location and then to perform L0 1001 with the LGP-30-coded PIR tape on the reader. In the example shown, we wish to start filling the bootstrap program into the NAREC equivalent of 6300. To do this we instruct the NAREC operator to perform a Rd 0fc0 order with the bootstrap tape (NAREC coded) on the reader. To execute this routine, instruct the operator to place the NAREC word 0fc0 50 100d 42 into location 1001 and then execute the order L0 1001.

Table B3
Coding of the Bootstrap Program

NAREC	LGP-30
(cr)	000C3w00'
nmn	000P 0000'
(cr)	000C3w04'
(tab) 0001 00 0000 00 (tab)(cr)	000I 0000'
(tab) 0000 80 0000 00 (tab)(cr)	000C3w08'
(tab) 0001 a7 e280 00 (tab)(cr)	000C3w14'
(tab) 0001 00 0000 00 (tab)(cr)	000C3w0J'
(tab) 0000 80 0000 00 (tab)(cr)	000P 0000'
mmmm	000C3w10'
(cr)	000I 0000'
	000U3w00'
	000Z 0000'
Loaded into NAREC location 0fc0 which is the LGP-30 equivalent 6300 or 3w00	

Appendix C

SIMULATED FLEXOWRITER MANUAL KEYBOARD INPUT

When the programmer chooses the option of Flexowriter manual keyboard input and an I order is detected by the simulator, NAREC comes to a switchbox stop if the light flag is up when this order is analyzed. This means that the computer needs more information that can be conveyed through the loading of a word in the NAREC switchbox. This switchbox, located on the NAREC console, consists of 48 toggle switches representing each individual bit of a NAREC word. Since the maximum LGP-30 input word will have eight characters, the NAREC word will be able to represent each of these in a six-bit form. The simulator processes the word thus transmitted by a character-by-character scan. Each 6-bit character is inspected, starting at the left-hand side of the NAREC word, and either four or six bits of the character, depending on the control word designation, are shifted into the least-significant bits of the pseudoaccumulator. This process is terminated when either a stop code is detected or when eight characters have been processed.

The simulation procedure differs from that of the actual LGP-30 in that the programmer must manually provide all six bits of each character needed, rather than just simply pressing a button on a Flexowriter. Each character must be in tape code format. When less than eight characters are used, the simulation of pressing the start-compute button is accomplished by loading a stop code, also in tape code format, immediately to the right of the last information character. When the full eight information characters are used, the simulator automatically performs this task. The simulator then does the work of the LGP-30 circuits by rearranging the bits, if 6-bit input is required, or by extracting the proper 4 bits, if that type of input is designated. The simulator program does not check for characters to be ignored or invalid characters for this type of input, i.e., all possible combinations of the 6-bit character code, except the stop code, are acceptable to the simulator and will be entered in rearranged form into the pseudoaccumulator; only the noncontrol character codes connected to the Flexowriter keys can enter the computer by this method on the actual LGP-30. The only exception to the rule for the simulator is the tab, where four or six zeros replace the original character to be shifted into the pseudoaccumulator.

Following are several examples of different types of LGP-30 words and how they are translated into the NAREC word for the switchbox. After the NAREC word has been placed into the switchbox, instruct the NAREC operator to push the NAREC Transfer Buttons.

Example 1

Figure C1 shows a particular bit pattern that might be used. Note that the bits between each pair of arrows represent a character, the bit to the far right in each case containing a one, and the most-significant bit a zero, except for the last character which represents a stop code. This is true of all number input codes. The other bits of each character are filled in the usual bit-by-bit fashion which is summarized in Table C1. The NAREC word b5304d7c1400, which would be given to the NAREC operator for the switchbox as shown in Fig. C1, could represent one of two words on the LGP-30. It could be the NAREC hexadecimal notation for either the short 4-bit input word 6906w0', which enters the pseudoaccumulator as the NAREC equivalent of xx6906w0, where xx represents

the two hexadecimal digits that were at the far right of the previous contents of the pseudoaccumulator, or the 6-bit input word 6906w0', which enters the pseudoaccumulator as the NAREC equivalent of the LGP-30 word 6e609fw8.

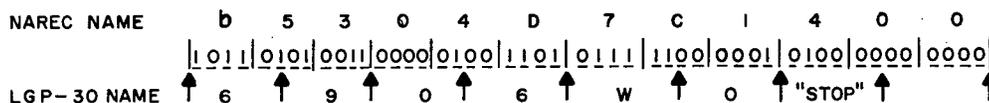


Fig. C1 - Input of a data word by Flexowriter

Table C1
Bit-by-Bit Representation of Characters to Make Up
Switchbox Word for Flexowriter Manual Input

Representations					
Numerical	Commands	Controls		Balance of Keyboard	Signs
)0 000001	Zz 100000	Lower Case	000010	;: 100111	= + 100101
L1 000011	Bb 100010	Upper Case	000100	?/ 101001	- - 100011
"2 000101	Yy 100100	Color Shift	000110] 101011	
#3 000111	Rr 100110	Carriage Return	001000	[, 101101	
\$4 001001	Ii 101000	Back Space	001010	Vv 101111	
%5 001011	Dd 101010	Tab	001100	Oo 110001	
_6 001101	Nn 101100	Cond. Stop (')	010000	Xx 110011	
&7 001111	Mm 101110	Start Read	000000		
'8 010001	Pp 110000	Space	100001		
(9 010011	Ee 110010	Code Delete	111111		
Ff 010101	Uu 110100				
Gg 010111	Tt 110110				
Jj 011001	Hh 111000				
Kk 011011	Cc 111010				
Qq 011101	Aa 111100				
Ww 011111	Ss 111110				

Example 2

Control words recognized by the PIR will probably be the most commonly needed by this type of input. For this reason each of these words is discussed below, together with short-cut methods for translating each for the NAREC operator.

1. Figure C2 shows the bit configuration for a comma code when the word is ,000000'. The number of hexadecimal words that are to be loaded will have to be filled in the bit-by-bit manner using the numbers shown in Table C1. The skeleton word in NAREC hexadecimal is b41041041041; for a one-word fill, the NAREC word is b41041041043; for a 49-word fill, the NAREC word is b410410410e3.

2. The start-fill (;) code shown in Fig. C3(a) represents the start fill into location zero, given to the NAREC operator as the hexadecimal word 9c1041041041. In general

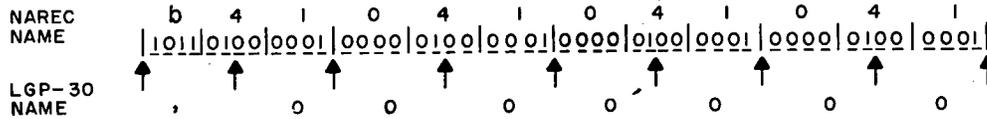
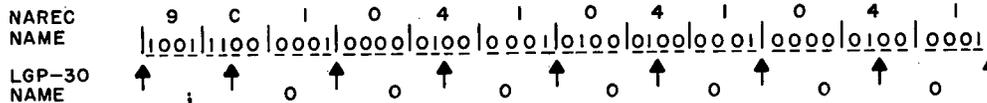
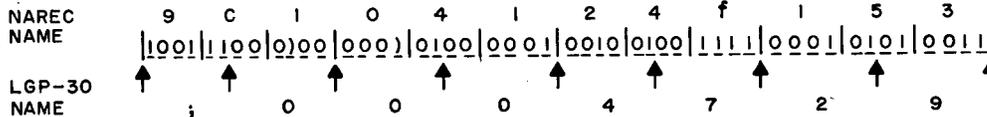


Fig. C2 - Input of comma control word

practice this code will be used mostly to load information into locations different from zero. This may be done either by the bit-by-bit method using Table C1 or by looking up the track-sector address in Table C2 and getting the equivalent NAREC notation. For example, the basic NAREC code word will always be 9c1041xxxxxx, where xxxxxx represents the decimal track-sector of the first location to be filled. Suppose we wish to start stowing information into location 4729 by a bit-by-bit method, the resulting NAREC word is 9c104124f153. By looking at Table C2 we find that the desired address is again 24f153, and with the constant part of the start-fill instruction combined, the same entire word results (, 9c104124f153). To start filling into location 1010, the NAREC result would be 9c10410c10c1.



(a)



(b)

Fig. C3 - (a) Input of start-fill code for location 0000 and
(b) Input of start-fill code for location 4729

Table C2
6-Bit Representation of Track and Sector for NAREC Switchbox Word

LGP-30 Track-Sector Notation	NAREC Notation for ;/.+	NAREC Notation for v	LGP-30 Track-Sector Notation	NAREC Notation for ;/.+	NAREC Notation for v
00	041	041	16	0cd	0c1
01	043	043	17	0cf	0c3
02	045	045	18	0d1	0c5
03	047	047	19	0d3	0c7
04	049	049	20	141	0c9
05	04b	04b	21	143	0cb
06	04d	04d	22	145	0cd
07	04f	04f	23	147	0cf
08	051	051	24	149	0d1
09	053	053	25	14b	0d3
10	0c1	055	26	14d	0d5
11	0c3	057	27	14f	0d7
12	0c5	059	28	151	0d9
13	0c7	05b	29	153	0da
14	0c9	05d	30	1c1	0dd
15	0cb	05f	31	1c3	0df

Table continues

Table C2—Continued

LGP-30 Track-Sector Notation	NAREC Notation for ;/ .+	NAREC Notation for v	LGP-30 Track-Sector Notation	NAREC Notation for ;/ .+	NAREC Notation for v
32	1c5	141	48	251	1c1
33	1c7	143	49	253	1c3
34	1c9	145	50	2c1	1c5
35	1cb	147	51	2c3	1c7
36	1cd	149	52	2c5	1c9
37	1cf	14b	53	2c7	1cb
38	1d1	14d	54	2c9	1cd
39	1d3	14f	55	2cb	1cf
40	241	151	56	2cd	1d1
41	243	153	57	2cf	1d3
42	245	155	58	2d1	1d5
43	247	157	59	2d3	1d7
44	249	159	60	341	1d9
45	24b	15b	61	343	1db
46	24d	15d	62	345	1dd
47	24f	15f	63	347	1df

3. The set-modifier control word (/) may be translated either by the bit-by-bit method using Table C1 or by using Table C2 for the address portion. Here, the basic NAREC word as shown in Fig. C4 is a41041041041. If the set modifier were to be 2347, we take the basic form 241041xxxxxx and replace the x's by items from Table C2 to get the desired result a4104114724f.

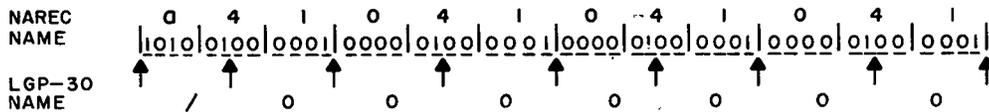


Fig. C4 - Input of the set-modifier control word to location 0000

4. The basic NAREC word for the stop-transfer (.) control word for location zero, as shown on a bit-by-bit basis in Fig. C5 is ac1041041041, or using the basic form ac1041xxxxxx, we find an LGP-30 .0003219' to be the NAREC input word ac10411c50d3.

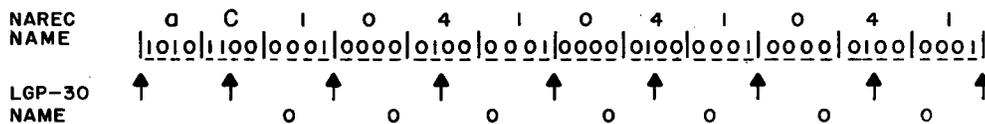
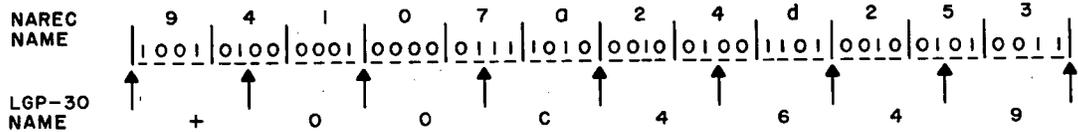
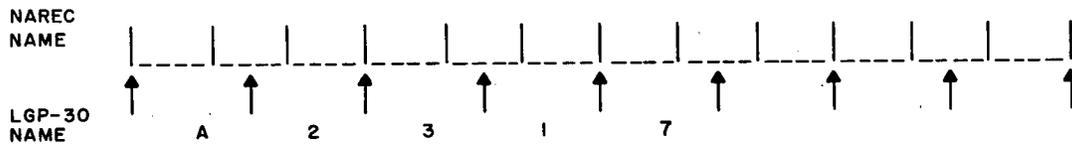


Fig. C5 - Input of the transfer control word to location 0000

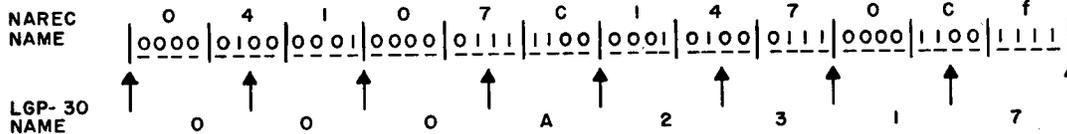
5. The command control word (+) has the basic NAREC word 9410yyxxxxxx, where yy is the order code. Table C3 gives the possible order codes, leaving the x's (track-sector) to be filled in by the programmer by using the information in Table C2. As an example of this use of the Flexowriter input, suppose it is desired that the instruction A2317 be placed into the memory location 4649. We would enter the PIR at the normal entry by executing the NAREC Order L0 1007. When the computer comes to a call for information from the NAREC switchbox, enter the NAREC equivalent of +000C4649, which is 94107a24d253 as shown in Fig. C6(a). As before, this word may be obtained either through the bit-by-bit method or by using the basic word (Table C3 for the yy portion and Table C2 for the xxxxxx portion). When the simulator calls for a second word from the switchbox a word is loaded as shown in Fig. C6(a) or C6(b). The first of these figures gives the word if the pseudo has been cleared previous to the I order, and the second figure gives the word if the pseudoaccumulator has not been previously cleared.



(a)



(b)



(c)

Fig. C6 - (a) Input of command control word, (b) input of order, and (c) input of order with leading zeros

6. The last of the control words for the PIR is the hexadecimal (v) fill. This is different from the others in that the starting address and number of words to be filled must both be in hexadecimal notation. For this reason, Table C2 has this notation separately in the right column. Figure C7 shows the basic form of this word. The V code at the left will always remain constant when using this control word. The middle 18 bits must be done in the bit-by-bit manner, remembering that the least-significant bit of each 6-bit character must always be a one and the most-significant bit a zero. The value of each of

Table C3
Filling the Order Code yy of the NAREC Command Control Word 9410yy xxxxxx

Order to PIR	Possible Order Codes (yy)	Order to PIR	Possible Order Codes (yy)
+00A	94107c xxxxxx	+00N	94106c xxxxxx
+00B	941062 xxxxxx	+00P	941070 xxxxxx
+00C	94107a xxxxxx	+00R	941066 xxxxxx
+00D	94106a xxxxxx	+00S	94107e xxxxxx
+00E	941072 xxxxxx	+00T	941076 xxxxxx
+00H	941078 xxxxxx	+00U	941074 xxxxxx
+00I	941068 xxxxxx	+00Y	941064 xxxxxx
+00M	94106e xxxxxx	+00Z	941060 xxxxxx

these bits is indicated in the figure for the convenience of the programmer. The right part of the NAREC word is the hexadecimal track and sector of the word, which as mentioned above may be obtained from the right column of Table C2. To take an example, suppose a program consists of 124 words in LGP-30 hexadecimal code to be loaded into memory starting at location 3147. The resulting code word, as seen in Fig. C8, is bc13d90df15f.

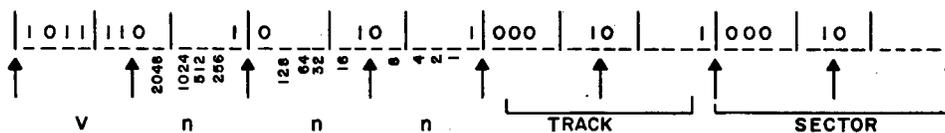


Fig. C7 - Basic form of hexadecimal fill control word

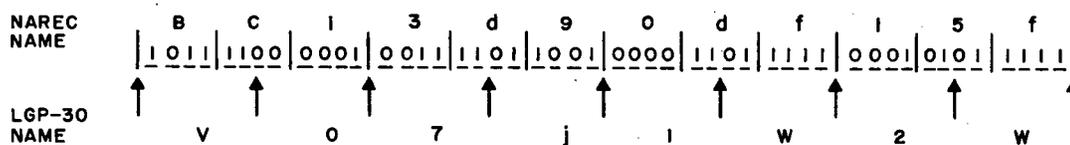


Fig. C8 - Input of hexadecimal fill control word

NOTE 1: 1015 1110 112f 1161 1242 1296 12f3 12e9
1099 1119 1137 11e4 1245 12a1 1301 1200
109d 111e 113a 1233 1292 12B7 130B 1499
10cd 1122 1150 123d 1294 12eB 131a

If use new coding add:

1603 16c1 16ec 1723 1755 17B6 17cB 1826

NOTE 2: 143e 14d0 12a8 1203 10B5 1009
1499 12cc 12c6 10B6 1044

If use new coding add:

1700 1746 1604 1780 17B6 1820
1704 1749 16a6 1784 17c0 1824
1740 17a0 16e0 179c 17c4
1744 17a3 16e4 17ae 17c6

NOTE 3: 1140 1102 1180 10f4 10c0 1062 1076 1034
1155 1104 10e0 10f6 10d1 1064 1081 1053
1161 1105 10e1 10a1 10d4 1067 1007 1051
14a6 1107 10f2 10a2 14B5 1071 1022 1059

If use new coding add:

1755 1744 16a4 16e4 17c4 1824
1704 1747 16a7 16eB 17c7 1826
1722 1749 16c1 1792 17cB

NOTE 4: 1207 12cd 1001 1050 10c1
12c4 14a4 1004 10B0 1201

If use new coding add:

165f

NOTE 5: 137f 124c 1412 1356 1433 148f
1011 140e 1348 142B 1436 14ce
138c 1343 1415 1369 1439

NAVAL RESEARCH LABORATORY

CODING SHEET PREC-NR-38-438 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPF NO. 4321 PROGRAMMER B. and E. WALD SHEET NO. 1

STOR. AGT. LOC. TION	ORDER PAIR		CODED ORDER PAIR		REMARKS		
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.			
1000	(A) A	14e0	10	0000	00 (14e0) FETCH NEXT ORDER AND/OR TRAILE LAST ORDER		
100B-1	(A) A	100d	50	100d	42 FETCH ORDER, STOW IN P		
2	(A) U	UA	102f	26	0000	40 EXTRACT L&P-30 COMMAND	
3	A L	Ad	0007	30	102e	54 VONNATE ENTRY - ORDER	
4	LAR 100b	100d A	1006	21	100d	50 PLACE OPERAND ADDRESS IN	
5	AR 7	(A) U	0007	31	102d	26 NAREC RT. ADDR. A & U	
6	UA	LO	0000	40	0000	10 (100d) ENTRY TO ORDER EXECUTION	
7	XA	A d	0000	80	100c	42 SET "0" IN α , β , CLOCKS	
8	A B	A B	126e	42	1269	42	
9	LAL d	RA	1001	20	102c	50	
a	A B	RA	1200	20	141d	50	
b	A B	LO	14e0	42	1001	10	
c	()	()	0000	00	0000	00 SEE NOTE # 3	
d	()	()	0000	00	0000	00 SEE NOTE # 4	
e	(CONTROL WORD)	()	0000	00	0000	00 SEE NOTE # 5	
f	()	()	0000	00	0000	00 (14e0)(140d)	
1006	A 101f	LO 101f	101f	42	0000	10 (14e0)(140d) STOW ADDR. - INVP = NO BKPT	
1010-1	100e A	LAR 101e	100e	50	101e	21 PLACE BPT. BITS IN POSITION	
2	XU	(A) U	0000	71	102B	26 CORRESP. TO TRAPK BITS	
3	UA	(101f) U	0000	40	101f	26 MULTIBLY EXTRACT	
4	X A	CUL	0000	80	1200	14	
1010-5	A	CL	1000	126e	50	1000	12 IF CLOCK OFF: STOP
6	1016 A	LO 12cB	1016	50	12cB	10 CLOCK ON: ADD READ TIME	
7	1016 A	CR 101a	126e	54	101a	13 MAKE CLOCK +, TURN OFF	
8	101B A	LO 1280	101B	50	1280	10 CLOCK SWITCH	
1017-9	LO 1179	CL 12a3	1179	10	12a3	12 TEST IF CLOCK ON	
1017-a	RO 1200	A B	1200	11	126e	42	
b	101B A	LO 1280	101B	50	1280	10 PRINT CHARACTER IF VALID	
c	STOP	()	0001	82	()	()	
d	()	()	1000	11	0000	80 (128c)(130a)(137d)(102a)	
e	()	()	0000	00	0000	00 (102b)(111a)(113d)(116c)	
f	()	()	0000	00	0000	00 (1011)(1045)	
g	()	()	0000	00	0000	00 (1010)(103)(1033)(1035)(1043)	

CONTROL WORD ENTRY = LO 1248
INVERSE N = LO 1007
LOAD PIR = LO 1068

CODING SHEET PREC-NR-38-438 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPF NO. 4321 PROGRAMMER B. and E. WALD SHEET NO. 2

STOR. AGT. LOC. TION	ORDER PAIR		CODED ORDER PAIR		REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.		
1006	(A) A	A 1021	1029	54	1021	42 (1020) BRING OPERAND
1	()	()	0000	00	0000	00
2	A d	LO	100c	42	1200	10 STOW IN α
3	()	()	()	()	()	()
4	2 @ 12	()	0002	00	0000	00 (1044)
5	ADDRESS - NORMAL POSITION	()	0000	07	ff80	00 (1034)(1046)
6	ALL BPT. ADDRESS	()	ffff	f8	00ff	ff (1033)(1043)
7	42-31:11	50-e-42	0000	31	0000	0e (1031)(1041)
8	SET UP STOW, FETCH OPERAND	()	1036	42	0000	50 (1030)(1040)
9	BRING "B" OPERAND	()	1021	11	0000	50 (1020)
a	()	()	()	()	()	()
b	()	()	0000	00	0f00	00 (1012)
c	()	()	1000	00	0000	00 (10a9)(11e3)(11e5)
d	()	()	0000	00	0fff	00 (1005)
e	()	()	1010	00	0000	00 (1003)
1006	L&P-30 COMMAND BITS	()	0001	80	0000	00 (1002)(14a4)
1030	(A) A	A 1032	1028	54	1032	42 SET STOW STOWER, OPERAND FETCH
1	(A) A	LO 1032	1027	55	1032	10 (1030)
1031-2	(1036 42)	XXXX 50	0000	00	0000	00 STOW STOWER, FETCH OPERAND
3	(A) U	U 101f	1026	26	101f	43 ERASE OPERAND ADDRESS
4	α A	(A) U	100c	50	1025	26 MERGE WITH α ADDRESS
1046-5	UA	101f Ad	0000	40	101f	54
6	(1036 11)	XXXX 42	0000	00	0000	00 STOW (1032)
7	LO	CLOCK SWITCH	1200	10	()	()
8	()	()	()	()	()	()
9	()	()	()	()	()	()
1400-10	1098 A	CL 103f	1098	50	103f	12 SET UP FOR SWITCH IN
103f-10	(A) A	A 1098	122e	50	1098	42 "P" ORDER FOR WHEN
c	LO 1401	()	1401	10	()	() SOME CLOCK IS RUNNING
d	()	()	()	()	()	()
e	()	()	()	()	()	()
103a-10	(A) A	RO 103B	11de	50	103B	11

NAVAL RESEARCH LABORATORY

UNCLASSIFIED

CODING SHEET
 PREC-REL-36-606 (REV. 2-58)
 DATE 4/18/62 PROBLEM NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 3

STOP- AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
R→1740	(101675) A	1042	1675	10	1042	42	SET STOW STOWER, OPERAND FETCH
1	-RAD	LO 1042	1027	55	1042	10	
1041→2	(1026 42 XXXX 50)		0000	00	0000	00	(1040) STOW STOWER FETCH OPERAND
3	(*A)U	U 101F	1026	26	101F	43	ERASE OPERAND ADDRESS
4	0 A	-RAD	1001	50	1024	54	MERGE WITH (0+2)
5	LAR 101e	AL 7	101e	21	0007	30	
6	(*A)U	LO 1035	1025	26	1035	10	
7	EXTRACT	TRACK	0000	07	0000	00	(1050)
8	TEMP		0000	00	0000	00	(1148)(1157)(1159)(1153)
9	STOP CODE (TAPE)		0000	00	0000	10	(1143)(1144)(1162)
a	PHOTO IGNORE LIST	11a d	00	0000	00		(1141)
b	TEMP		0000	00	0000	00	(116a)(116b)(1171)(1181)(1501) (1141)(1154)(1160)(1167)(1168)
c	1044		0000	00	0000	01	
d	TRIM TO 32 BITS		1fff	ff	ffe0	00	(1058)
e	TEMP		0000	00	0000	00	(1053)(1054)(1058)
f	TEMP (TRACK)		0000	00	0000	00	(1051)(1052)(1055)
I 1750	0 A	(*A)U	100d	50	1047	26	EXTRACT TRACK FROM P
1	U 104f	LO (1022)	104f	43	0000	10	(135a)(1365)
1051→2	TRACK A	AR 10	104f	50	000a	31	PLACE 4 TRACK BITS AT
3	A TEMP	dU	104e	42	100c	24	LOWER END OF d
4	UL 4	RO 1057	0004	34	1057	11	
1051→5	TRACK A	AR 8	104f	50	0008	31	PUT 6 TRACK BITS AT
6	A TEMP	dU	104e	42	100c	24	LOWER END OF d
1054→7	UL 6	UA	0006	34	0000	40	
8	TEMP Ad	(*A)U	104e	54	104d	26	TRIM WORD
9	U d	LO 105a	100c	43	105a	10	
1059→a	105a A	LO PRE TIME	105a	50	12c B	10	COMPUTE PRE, AFT TIME
b	105B A	LO AFT TIME	105B	50	12c 3	10	
c	BLOCK A	CL 11c 3	126e	50	11c 3	12	
d	PRE AR	A PRE	1274	54	126e	42	ADD PRE TIME - STILL ON?
e	CL 11d c	AFT AR	11d c	12	1273	54	
f	A 126e	LO 1083	126e	42	1083	10	

CODING SHEET
 PREC-REL-36-606 (REV. 2-58)
 DATE 4/18/62 PROBLEM NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 4

STOP- AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
D 1060	-RAD	A 1061	106f	54	1061	42	SET UP FETCH OPERAND (1040)(1043)(1044)(1075)
1	(0000 71 XXXX 50)		0000	00	0000	00	BRING OPERAND (1075)
2	A 1061	101 A	1061	42	100c	52	STOW TEST FOR OVERFLOW
3	-1061/AR	CL OVERFLOW	1061	57	10f 8	12	
4	d A	D/1061	100c	50	1061	70	
5	(101067 XU)		0000	00	0000	71	TRIM d (13a1)(1491)
1075→6	-RAD	(*A)U	106e	54	106d	26	
1065→7	U d	LO CLOCK SWITCH	100c	43	1200	10	STOW RESULT IN d
8	BT (1500 ; 0000) 0c		1500	83	0000	0c	BLOCK TRANSFER AIR TO
9	STOP	LO 1248	0002	82	1248	10	LOCATION 0000 - 00BF
11d d	-RA	A 1200	101d	50	1200	42	
106a	LO 11c 3		11c 3	10			
c	SET UP "N" MULTIPLY		100c	50	0000	60	(1070)(1080)
d	EXTRACT 31 BITS		1fff	ff	ffe0	00	(1066)(10d2)(10d4)
e	1031		0000	00	0020	00	(1066)
f	SET UP OPERAND FETCH		0000	71	0000	50	(1060)
N 1070	-RAD	A 1071	106c	54	1071	42	SET UP MULTIPLY
1	(100c 50 XXXX 60)		0000	00	0000	00	MULTIPLY (1070)(1072)(1074)
2	U 1071	AU	1071	43	0000	41	STOW LEAST SIG. BITS
3	UL 31	U 1061	001f	34	1061	43	SHIFT SD BIT 31→SIGN
4	1071 A	AR 13	1071	50	000d	31	BIT
1082→5	(1061 Ad XU	RO 106a	1061	54	0000	00	(13a3)(1493) SHIFT LEAST SIG. BITS RIGHT
6	Ad	LO SWITCH	100c	42	1200	10	AND COMPLETE THE WORD
115→7	A CLOCK	XU	1269	42	0000	71	
8	LO 12f 6		12f 6	10			
9	TEMP.		0000	00	0000	00	(1157)(1158)(1159)(1173)(1174)
a	8@12		0008	00	0000	00	(1154)
b	STROKE COUNTER		0000	00	0000	00	(1134)(1172)
c	END PHOTO IGNORE LIST	11B 3	00	0000	00		(1140)
d							
e	CODE DELETE		0000	00	0000	3f	(1147)
f	TAPE FEED		0000	00	0000	00	(1145)

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRNC-REL-24-50 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPC NO. 4321
 PROGRAMMER B and E. WALD SHEET NO. 8

STOP AND LOCAL ACTION	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
1006 → C	10e0	aU	U 10df	100c	24	10df	43	BRING a TO 10df
	1	YL	11a	0000	71	100c	43	SET a TO 0
	2	RA	A 10e4	10de	54	10e4	42	SET UP TRIM PROCEDURE
	3	(10e4 2)	XXXX 43)	0000	00	10df	50	(1414)
1006 → 4	(10e4 2)	XXXX 43)	0000	00	0000	00	00	(10e2)
	5	LO	CLOCK SWITCH	1200	10			
10e3 → 6	10df U	RO 10e4	10df	24	10e4	11		
111f → 7	"STOP" A	CUR 1120	126a	50	1120	15		
	8	LO 111d	111d	10				
112c → 9	"STOP" A	CUL 112d	126a	50	112d	14		
	a	LO 112b	112b	10				
12fa → b	"STOP" A	CUL 12fe	126a	50	12fe	14		
	c	LO 12f9	12f9	10				
	d	-60d + 10d	ffc4	00	000a	04	04	(110d)
	e	-7d + 2d	fff9	00	0002	04	04	(1113)
1006 → f	-3½d + 29s	fffc	ac	0000	74	(1114)(12f3)(130f)		
1006 → A	10f0	RA	A 10f1	10ff	54	10f1	42	
	1	(10f1 11	XXXX 50)	0000	00	0000	00	(10e2)(10e3)(10e4)(10e5)
	2	aU	CSL 10f6	100c	24	10f6	16	JUMP: UNLIKE SIGNS
	3	A 10f1	110f1A	10f1	42	10f1	52	
	4	1a1 A	CR 10f5	100c	56	10f5	13	JUMP: +
10f4 → 5	LO OVERFLOW	10f1A	10f8	10	10f1	50		JUMP: TO OVERFLOW
10f2 → 6	a A	A a	100c	54	100c	42		
	7	LO SWITCH	1200	10				
1107, 1109, 1063, 1065 → 8	10f8 A	LO 1320	10f8	50	1320	10		PRINT:
	9		13fc	00	0000	00		"OVERFLOW"
	a		ffff	00	0000	00		
	b	TRACE a, p, d	10f8	50	14a0	10		
	c	STOP	0003	82				
	d	CONSTANT	0000	00	000a	04	04	(113B)
	e	SET UP "4" FETCH	1101	11	0000	24	24	(110D)
	f	SET UP "A" OPERAND	10f1	11	0000	50	50	(10f0)

CODING SHEET
 PRNC-REL-24-50 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPC NO. 4321
 PROGRAMMER B and E. WALD SHEET NO. 9

STOP AND LOCAL ACTION	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
1006 → 1100	RA	A 1101	10fe	54	1101	42	(1100)	
	1	(1101 11	XXXX 24)	0000	00	0000	00	BRING OPERAND TO U
	2	aA	CSL 1105	100c	50	1105	16	
1108 → 3	U 10f1	-10f1A	10f1	43	10f1	55		
	4	A a	LO SWITCH	100c	42	1200	10	
1107 → 5	U 10f1	1a1 A	10f1	43	100c	52		
	6	110f1A	CR 1107	10f1	56	1107	13	
1106 → 7	LO OVERFLOW	aA	10f8	10	100c	50		
	8	RO 1103	1103	11				
	9	LO 10f8	10f8	10				
1118 → a	XU	RO 1119	0000	71	1119	11		START FLEX OUTPUT FOR EFFECTIVE "P" (P ₀)
109f → b	TRACE WA	XU	108f	50	0000	71		IS CHARACTER "A" "CR"
1127 → c	AU	"CR" A	0000	41	1275	50		YES: SET SIG TO -60d
1112 → d	CUL 1111	RA	1111	14	10e2	50		ADD EXECUTE ANS., STOP
1113 → e	EXECUTE ANS	A 816	CLOCK	1273	54	126e	42	
1114 → f	108f A	A PRINTER	108f	50	126d	42		PUT CHAR. IN PRINTER
	110	RO 1000	1000	11				
110A → 1	"Aa" A	CUL 1113	1276	50	1113	14		JUMP - NOT TAB
	2	RO 110d	110d	11				
1111 → 3	RA	LO 110e	10ee	50	110e	10		BRING -7d
109f → 4	RA	LO 110e	10ef	50	110e	10		" -3½d
1090 → 5	LITTLE CLOCK	A CL 1132	1269	50	1132	12		
	6	READ ANS	A LITTLE CLOCK	1274	54	1269	42	IF LITTLE ON: ADD READ
	7	CL 111B	EXECUTE ANS	111B	12	1273	54	STILL ON: ADD EXECUTE
	8	A LITTLE CLOCK	CL 110a	1269	42	110a	12	NOW OFF: PULL CHAR.
110A → 9	RO 1000	SCU	1000	11	003f	36		
128A → a	STM	RO 10c6	0000	84	10c6	11		
1117 → b	XU	RO 1122	0000	80	1122	11		
1120 → c	(RA	CUR 111e	0000	50	111e	15		(111e)(10c7) BEGINNING
10e8 → d	RA	LAL SWITCH	101d	50	1200	20		OF BAD CHAR. LIST
111c → e	RO 1000	11 111c	1000	11	111c	22		
	f	-RA	CL 10e7	11fB	55	10e7	12	

NAVAL RESEARCH LABORATORY

49

UNCLASSIFIED

CODING SHEET PERC-NR-38-636 (Rev. 2-58) DATE 4/18/62 PROBLEM NO. 4321 TAPE NO. PROGRAMMER B. and E. WALD SHEET NO. 10

STEP AGE LOCAL- TION	ORDER PAIR		CODED UNDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
10e7 1127	1	LO 111c	LITTLE CLOCK A	111c	10	1269	50	ADD -6A TO LITTLE CLOCK
		* Ad	A LITTLE CLOCK	1267	54	1269	42	
11e6 1118	2	RO 1000	SCU	1000	11	003f	36	READ CHARACTER
12e6 120c	3	STM	LO 10c5	0000	84	10c5	10	
10c6 112c	4	LAL 1125	10dF U	1125	20	10dF	24	SET UP BAR CHARACTER LOOP (1124)(1128)
10e6 131d, 115d 1126	5	() A	CUL 112B	0000	50	112B	14	
	6	108d A	CR 1127	108d	50	1127	13	
	7	RO 1120	RO 110c	1120	11	110c	11	
	8							
130f 1125	9	EXECUTE ANS A	A BIG CLOCK	1273	54	126e	42	SET BIG CLOCK TO -60d + EXECUTION TIME
		108f A	LO 1150	108f	50	1150	10	
		11L 1125	* Ad	1125	22	11fB	55	
		C	LO 10e9	10e9	12	1125	10	
118f, 1275 10e9		LITTLE CLOCK A	* Ad	1269	50	1267	54	SET LITTLE CLOCK TO -6d + EXECUTION TIME
11d		EXECUTE ANS A	A LITTLE CLOCK	1273	54	1269	42	
		RO 1000		1000	11			
	1							
128d, 123f 1115	2	BIG CLOCK A	READ ANS A	126e	50	1274	54	ADD READ ANS TO BIG CLOCK JUMP: OFF
		A CLOCK	CL 11f7	126e	42	11f7	12	
		108d A	CR 113a	108d	50	113a	13	
1313, 1305 113e	3	BIG CLOCK A	EXECUTE ANS A	126e	50	1273	54	ADD EXECUTE ANS. TO BIG JUMP: NOW OFF
		A CLOCK	CL 1138	126e	42	1138	12	
		RO 1000		1000	11			
1136	4	1138 A	PRINT IF CLOCK SWITCH	1138	50	1280	10	PRINT CHARACTER IN PRINTER IF VALID
		* A	LAL 101d	101d	50	1200	20	
1312 1134	5	RO 1000	BIG CLOCK A	1000	11	126e	50	TURN OFF CLOCK SWITCH
		(*) A U	* A	10fd	26	10fd	50	IS TIME IN CRITICAL RANGE
		CUR 11f4	BIG CLOCK A	11f4	15	126e	50	
		* Ad	CL 11f3	12c0	54	11f3	12	
1307 114c	6	113e A	LOGICAL SUM	113e	50	1315	10	YES: PERFORM LOGICAL SUM
		LO 1135		1135	10			

CODING SHEET PERC-NR-38-636 (Rev. 2-58) DATE 4/18/62 PROBLEM NO. 4321 TAPE NO. PROGRAMMER B. and E. WALD SHEET NO. 11

STEP AGE LOCAL- TION	ORDER PAIR		CODED UNDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
11e3 1148, 1143 1146	1	α A	AR 7	100c	50	0007	31	PUSH α TO RIGHT OF NAREC ACCUM
		A TEMP	* A	104B	42	104a	50	
		LAL 114c	XU	114c	20	0000	71	
		"STOP" A	SCU	1049	50	003f	36	
114B	2	CUR 1145	STM	1145	15	0000	84	JUMP: NOT STOP CODE
1144	3	IO WORD	"FEED" A	1160	10	107f	50	
		CUL 1147	RO 1142	1147	14	1142	11	JUMP: NOT TARE FEED
1144	4	"DELETE" A	CUR 1148	107e	50	1148	15	JUMP: NOT CODE DELETE
1147	5	RO 1142	U CHARACTER STOP	1142	11	1048	43	
		UA	(*) A U	0000	40	115e	26	
		"STOP" A	CUR 114B	1049	50	1148	15	JUMP: NOT PSEUDO-STOP
114e	6	RO 1144	CHARACTER STOP U	1144	11	1048	24	I.E. BIT IN STOP CODE
114f	7	() A	CUL 114e	0000	50	114e	14	POSITION BUT NOT LEAST, MOST SIG.
		RO 1141		1141	11			
114c	8	11L 114c	* Ad	114c	22	107c	55	
		(CL(1143))	LO 114c	0000	12	114c	10	(1351)(1367)
112a	9	1150 A	PRINTER	RO 1000	126d	42	1000	11
	1							
	2							
11e3	3	XA	A w ₂	0000	80	1048	42	SET w ₂ TO 0
		* A	A STROKE PRINTER	107a	51	107B	42	SET CR TO -8@12
		α A	AR 7	100c	50	0007	31	MOVE α TO RIGHT OF NAREC ACCUMULATOR
		A TEMP	6000 A	1048	42	6000	50	
		A w ₁	LAR w ₁	1079	42	1048	21	BRING WD FROM SWBOX
		w ₁ A	AL 2	1079	50	0002	30	USE SPECIAL PROCEDURE TO GET FIRST CHAR
		A w ₁	w ₂ A	1079	42	1048	50	
		XU	AL 2b	0000	71	001a	30	(1066)(1352)
1175	4	"STOP" A	CUL 1145	126a	50	0000	14	JUMP - NOT STOP CODE (1379)(13a7)
		(LO 1160)		0000	10			
1171	5	AW	RO 116a	0000	41	116a	11	
		CONSTANT		0000	00	0000	31	(1149)
	6							

NAVAL RESEARCH LABORATORY

NOTE 9: 119d 1190 1172 115c 1145 1163

CODING SHEET
 PERC-BEL-38-896 (REV. 2-58) DATE 4/18/62 PROGRAM NO. 4321
 PROGRAM B. and E. WALD SHEET NO. 12

SEE NOTE	ADDR. LOCAL 1170	ORDER PAIR		CODE ORDER PAIR			REMARKS	
		LEFT ORDER	RIGHT ORDER	ORDER NO.	ADDRESS	ORDER NO.		
	1160	TEMP U	UL 7	104B	24	0007	34	MOVE NEW A TO NORMAL
119d	1	LA	RO 1000	100c	43	1000	11	LCP-30 POSITION
115c	2	"STOP" U	PCU	1049	24	0000	90	PRINT STOP CODE
	3	LO 1160		1160	10			
119B, 119d, 115B, 1527	4	(PCU) LO 1165		0000	90	0000	10	(155-X)1566 PRINT CHARACTER
1146, 115B, 1164	5	LO 15e0 (A)U	15e0	10	117f	26		REARRANGE BITS FOR 4
	6	UA	AL 1	0000	40	0001	30	BIT INALT
15e1	7	104BA	AL 4	104B	54	0004	30	
116c	8	(A) 104B	116c 116c	104B	42	0000	10	(134c)(1375)(137B)
1144, 115B, 114F	9	"TAB" A	116c 116c	127f	50	116c	15	JUMP: NOT TAB
1171, 115d	a	104BU	UL 6	104B	24	0006	34	
	b	UA	A 104B	0000	40	104B	42	REARRANGE BITS FOR 6
1169	c	RO 1168	UA	1168	11	0000	40	BIT INPUT
	d	(A)U	Uw ₃	10Bc	26	10Bf	43	
	e	-w ₃ A	AL 1	10Bf	55	0001	30	
	f	A w ₄	w ₃ A	10Bd	42	10Bf	50	
1170	1	AR 5	w ₄ A	0005	31	10Bd	54	
	2	104BA	LO 115d	104B	54	115d	10	
1168	3	UL 107B	Cl 1160	107B	22	1160	12	INCREMENT CTR.: IF DONE
	4	w ₄ A	XU	1079	50	0000	71	JUMP
	5	AL 6	Aw ₄	0006	30	1079	42	SHIFT ORIGINAL WORD
	6	LO 115B		115B	10			LEFT 6 BITS
	7							
	8							
1019	9	XA	A BIG	0000	80	126e	42	SET BIG AND LITTLE
a		LITTLE CLOCK A	RO 1019	1269	50	1019	11	CLOCKS TO 0
b		END IGNORE LIST		11a2	00	0000	00	(1198)
c		END CONTROL LIST		118f	10	11B3	50	(1194)
d		END START LIST		11a8	00	0000	00	(1191)(10c5)(10c7)(10c9)
e		END JAM LIST		0000	84	11a5	50	(118d)
f		4 BIT CHARACTER		0000	00	0000	1e	(1145)

NAVAL RESEARCH LABORATORY

CODING SHEET
 FORM-NR-34-400 (Rev. 2-58)
 DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 13

STOP- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
117c 1162 1163, 1162 1197, 1193	180	A AR7	100c	50	0007	31	MOVE A TO RIGHT OF W	
	2	A 104B	104B	42	11fe	50		
	3	LAR 118B	118B	21	11fd	50	INITIALIZE LOOPS	
	4	LAL 118f	118f	20	11fc	50		
	4	LAR 1192	1192	21	11fB	50		
1199	5	LAL 1196	YU	1196	20	0000	71	
	6	"STOP" A	SCU	126a	50	003f	36	IS CHARACTER A STOP CODE
	7	CUL 1188	LO 119d	1188	14	119d	10	
1187	8	"FEED" A	CUK 1189	1189	50	1189	15	JUMP: NOT FEED
1188	9	RO 1185	"DELETE" A	1185	11	11fa	50	
	a	CUL 118B	RO 1185	118B	14	1185	11	
1186	b	STM	()A	0000	84	0000	50	JAM LOOP (1182)(1188)
116c	c	CUL 118d	STOP	118d	14	0004	82	
118c	d	IR 118B	-BA	118B	23	117e	55	
	e	CL 118f	RO 118B	118f	12	118B	11	
1192	f	()A	CUK 1190	0000	50	1190	15	START LOOP (1183)(1190)
118c								
	1190	LO 1160	IL 118f	1160	10	118f	22	
118f	1	-BA	CR 1192	117d	55	1192	13	
1191	2	LO 118f	()A	118f	10	0000	50	CONTROL LOOP (1184)(1194)
1195	3	CUL 1194	1194	14	0000	10	(131B)(1399)	
1193	4	IR 1192	-BA	1192	23	117c	55	
	5	CL 1196	RO 1192	1196	12	1192	11	
1195	6	()A	CUL 1198	0000	50	1198	14	IGNORE LOOP (1185)(1199)
1194	7	RO 1181		1181	11			
1196	8	IL 1196	-BA	1196	22	117B	55	
	9	LO 1196		0000	13	1196	10	(130c)(1390)
1193	a	PCU	RO 1181	0000	90	1181	11	
1199	b	PCU	RO 1164	0000	90	1164	11	
	c							
1197	d	STM	LO 1160	0000	84	0000	10	(130d)(138f)(1397)
	e							
	f							

CODING SHEET
 FORM-NR-34-400 (Rev. 2-58)
 DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 14

STOP- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
11a	0		0000	00	0000	0e	28	TAPE JAMPT CODES
	1		0000	00	0000	1a	52	FLEX
	2		0000	00	0000	18	48	JAM
	3		0000	00	0000	1c	56	
	4		0000	00	0000	1e	60	
	5		0000	00	0000	14	60	FLEX (1155)(1166)
	6		0000	00	0000	16	44	START (1157)(1160)
	7		0000	00	0000	12	36	(1139)(1184)(1160)
	8		0000	00	0000	3d	59	
	9		0000	00	0000	3B	55	FLEX
	a		0000	00	0000	39	51	IGNORE
	b		0000	00	0000	37	47	
	c		0000	00	0000	35	43	
	d		0000	00	0000	0e	28	
	e		0000	00	0000	02	L.C.	PHOTO
	f		0000	00	0000	04	U.C.	FLEX IGNORE
11B	0		0000	00	0000	06	C.S.	IGNORE
	1		0000	00	0000	08	C.R.	
	2		0000	00	0000	0a	B.S.	INTERNAL CODES
	3		0000	00	0000	1c	28	
	4		0000	00	0000	34	52	
	5		0000	00	0000	30	48	
	6		0000	00	0000	38	56	
	7		0000	00	0000	3c	60	
	8		0000	00	0000	3B	50	
	9		0000	00	0000	37	55	
	a		0000	00	0000	33	51	
	b		0000	00	0000	2f	47	
	c		0000	00	0000	2B	43	
118c	d	108f U	RO 1127	108f	24	1127	11	
1197	e	108d A	CL 118d	108d	50	118d	12	- if P "00"
	f	LO 112d		112d	10			

NAVAL RESEARCH LABORATORY

NOTE 10: 11d0 11d7 11e0 11ca 11cc
11d5 11d9 11ce 11c3 11c0

CODING SHEET
PRC-91-34-598 (REV. 2-58) DATE 4/18/62 PROGRAM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. Ward SHEET NO. 15

STOP AFC LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
1048 L3 11c0	READ ANS A	CR 11d2	1274	54	11d2	13		
11d2	A	A LITTLE CLOCK	1251	50	1269	42	TURN OFF CLOCK	
11c7	2	LO 1180	1180	10			TYPEWRITER - READER OPTION	
1048, 1049 105c	3	(LITTLE CLOCK) A CU 11d2	1269	50	0000	12	(134BX)(1374)(137a)	
11c7	4	READ ANS A	CR 11c7	1274	54	11c7	13	ADD READANS TO LITTLE
11c7	5	A	A LITTLE CLOCK	1251	50	1269	42	STILL ON: TURN OFF
11c4	6	A	A CLOCK SWITCH	101d	50	1200	42	
11c4	7	LO 1180	XU	1180	10	0000	80	
	8	SCU	STM	003f	36	0000	84	READ ONE FRAME OF TAPE
	9	"PSEUDO" A	CUL 11cB	11a5	50	11cB	14	CHECK FOR STOPS
	a	LO 11d2		11d2	10			AND PSEUDO-STOPS
11c9	b	"PSEUDO" A	CUL 11cd	11a6	50	11cd	14	
	c	LO 11d2		11d2	10			
11cB	d	"PSEUDO" A	CUL 11cf	11a7	50	11cf	14	
	e	LO 11d2		11d2	10			
11c2	f	"STOP" A	CUL 11c5	126a	50	11c5	14	
11d0	g	(LO 11d2)		0000	10			(11d0)
11d0	1	PCU	STOP	0000	90	0000	82	
SEE NOTE #10	2	STOP	XU	0005	82	0000	80	
	3	SCU	STM	003f	36	0000	84	
	4	"PSEUDO" A	CUL 11d6	11a5	50	11d6	14	
	5	LO 11d2		11d2	10			
11d4	6	"PSEUDO" A	CUL 11d8	11a6	50	11d8	14	
	7	LO 11d2		11d2	10			
1136	8	"PSEUDO" A	CUL 11da	11a7	50	11da	14	
	9	LO 11d2		11d2	10			
11d8	a	"STOP" A	CUL 11cl	126a	50	11cl	14	
	b	LO 11d0		11d0	10			
105c	c	11d2 A	LO 1280	11d2	50	1280	10	PRINT CHARACTER IF VALID
	d	LO 106a		106a	10			
	e	CONSTANT		12e0	14	108d	50	(103f)
	f	CONSTANT		1115	14	108d	50	(122c)

NAVAL RESEARCH LABORATORY

CODING SHEET
FORM-NR-28-458 (Rev. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TAKE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 16

STORE AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
11e 0	11ffA	CL STOP	11ff	50	1122	12	LIGHT FLAG OFF - STOP
11c3 → 1	X A	A 11ff	0000	80	11ff	42	LIGHT FLAG ON - TURN OFF
2	LO 1153		1153	10			
1099 → 3	" A	A 11ff	102c	50	11ff	42	SET LIGHT FLAG ON
4	RO 1000		1000	11			
5	" A	A 11ff	102c	50	11ff	42	SET LIGHT FLAG ON
6	RO 1122		1122	11			
14aB → 7	AL	AL 3	1271	24	0003	30	NEGATIVE - PUT 1244
8	A w,	A A	149c	42	132f	50	IN 14 REGISTER
9	AL	PCU	0001	30	0000	40	SHIFT 3 HIGH ORDER BITS
a	w, A	AL 4	149c	50	0004	30	INTO 14 REG, PRINT
b	A w,	A A	149c	42	132f	50	THEN CONTINUE FOR NORMAL
c	AL	PCU	0001	30	0000	90	4 (OR 6) BIT
d	w, A	AL 4	149c	50	0004	30	SHIFT, ADD PROPER CONSTANT
e	A w,	A A	149c	42	132f	42	AND PRINT
f	AL	PCU	0001	30	0000	90	
11f 0	w, A	AL 6	149c	50	0006	30	
1	A Ad	LO 14B1	11f2	54	14B1	10	
2	CONSTANT		1000	00	0000	00	(11f1)
113a 1145 → 3	11f3A	LO ^{MACRO} _{TRIM}	11f3	50	12a6	10	
113c → 4	LO 11f7	BI ^{CLOCK} _A	11f7	10	12a6	50	
5	A Ad	Ci 11f3	12c1	54	11f3	12	
6	LO 113e		113e	10			
1133 1144 → 7	11f7 A	LO ^{PRINT IF} _{VALID}	11f7	50	1280	10	
8	LO 11Be		11Be	10			
9	FEED		0000	00	0000	00	(1184)
a	CODE DELETE		0000	00	0000	3f	(1184)
b	INITIAL IGNORE		11a8	00	0000	00	(1184)(11f3)(112D)(12fc)
c	INITIAL CONTROL		11a2	00	0000	00	(1183)
d	INITIAL START		11a5	00	0000	00	(1183)
e	INITIAL JAM		11a0	00	0000	00	(1181)
f	LIGHT FLAG " " IF ON		0000	00	0000	00	(11e0)(11e1)(11e2)(11e5)

NAVAL RESEARCH LABORATORY

NOTE 11: 1014 12a3 1037 1076 10B5 10c2 10e5 1104
 101a 1022 1067 10a2 10c1 10d3 10f7 123d

NOTE 12: 130a 111d 11c6 100a 109a
 10B3 1139 106a 1097 109e

COUING SHEET
 PRC-91-34-400 (Rev. 2-58)

DATE: 4/18/62 PRIORITY NO. _____ TIME NO. 4321
 PROGRAMMER: B and E WALD SHEET NO. 17

STOR. AGE LOCAL. TION	ORDER PAIR		CODED ORDER PAIR				MARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1200	RO(1200)	XA	0000	11	0000	80	SEE NOTE #12 SET ANS. TO 0
1	A ANS	eA	1250	42	100d	50	COMPUTE $\Delta = (Q_i) - P$
2	AL 17	(k-A)U	0011	30	1252	26	
3	U w ₂	1001 A	1253	43	1001	50	
4	k Ad	(k-A)U	1251	54	1252	26	
5	UA	-w ₂ AR	0000	40	1253	55	
6	CL 1207	k AR	1207	12	1254	54	NEGATIVE: ADD 64
1206 → 7	A w ₂	eA	1253	42	100d	50	EXTRACT ORDER FROM P
8	(k-A)U	kA	1255	26	1256	50	
9	CUL 1215	kA	1215	14	1257	50	
1218 → a	AX	IL ANS	1258	42	1250	22	IF D CTR → -4
1216 → b	k AR	A ANS	125a	54	1250	42	ADD 64 TO ANS
1224 → c	w ₂ A	k AR	1253	50	1268	54	ADD 9 TO ANS
1213 → d	CL 120f	IL ANS	120f	12	1250	22	COMPUTE (X-15) ADD 64 IF NEGATIVE
1211 → e	LO 1230		1230	10			
1214 → f	k AR	CL 1212	125e	54	1212	12	NEGATIVE: SUBTRACT 7
120d → 1	-k AR	XU	125e	55	0000	71	ADD 7 IF = 0
1	CUR 120d	LO 1230	120d	15	1230	10	IF OPTIMUM: EXIT
120f → 2	A IL	IL X	0000	41	1258	22	IF NOT "
3	CR 120d	UA	120d	13	0000	40	ADD 64 TO ANS
4	LO 120f		120f	10			NOT NEG: INCREMENT X
1209 → 5	kA	CUL 1217	125d	50	1217	14	IF X + : NON OP: ADD 64
6	kA	LO 120a	125e	50	120a	10	IF X - : SUBTRACT 7
1215 → 7	kA	CUL 1219	1260	50	1219	14	
8	kA	LO 120a	1261	50	120a	10	
1217 → 9	wA	CL 1221	10Bc	50	1221	12	
a	U w ₂	(k-A)U	10Bc	43	1252	26	
b	U w ₂	kA	1253	43	126f	50	
c	LAR 11d2	RO 12c2	12d2	21	12cd	11	
d	AU	-k AR	0000	41	1265	55	
e	CL 121f	IL ANS	121f	12	1250	22	
121e → f	UA	ANS AR	0000	40	1250	54	

NAVAL RESEARCH LABORATORY

NOTE 13: 1008 12B6 1290 1121 11c1 1230 1115 1077
 12ff 12B4 123f 112d 11c3 129c 10cB 1083
 12fe 12e0 123e 112e 11c5 1118 109c
 12e2 129B 1120 117a 1232 1116 109B

NOTE 14: 1303 1008 113c 1133 105c 1302 1310 1471
 12f8 129f 12e8 1015 105d 105f 12e5
 12f4 1293 113a 1132 110e 1305 12e7
 12a5 1291 1136 1017 12eB 12e3 1129
 12a0 1179 1135 101a 11f4 1470 1311

NOTE 15: 105e 12f4 12ca 12e7 110e 1129 1135
 12ff 12B6 12e2 109c 1117 112e

NOTE 16: 12B5 12e1 1302 11c0 105d 1132 1470
 12d2 12e4 1310 11c4 1116 12ae

CODING SHEET
 PRG-PR-34-68 (Rev. 7-58)

DATE 4/18/62 PROBLEM NO. _____ TAPC NO. 4321
 PROGRAMMER B. and E. Ward SHEET NO. 20

STORE AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1260	"M"		0000	20	0000	00	(1217)
1	-5@12		fffB	00	0000	00	(1218)
2	CONSTANT		10a3	00	0000	00	(1254)
3	"L"		0001	40	0000	00	(14d3)
4	RETURN ADDRESS		121d	00	0000	00	(1218)
5	4@18		0000	10	0000	00	(121d)
6	"T"		0001	60	0000	00	(122)(14e5)
7	-6@12		ffa	00	0000	00	(122)(1223)(121e)(1221)
8	-15@12		fff1	00	0000	00	(120c)
9	LITTLE CLOCK		0000	00	0000	00	SEE NOTE #13
a	STOP		0000	00	0000	10	(1176)(11c)(11a) (1232)(12e7)(12e9)(12eB)(15B)
b	INVALID BEGINNING		11B3	00	0000	00	(1282)
c	LIST OF INVALID		126d	24	11Bd	50	(1285)
d	PRINTER		0000	00	0000	00	(1214)(1257)(1218)(1219)(1220) (120c)(1104)(1232)(1287)
e	BIG CLOCK		0000	00	0000	00	SEE NOTE #14
f	W ₄		0000	00	0000	00	(1298)(129d)
1270	STOP CODE		0000	00	0000	10	(1299)
1	LEAST SIGNIFICANT BIT		0000	00	0000	01	(11e7)(1297)(129d)(14B6)
2	-6@12		ffa	00	0000	00	(1098)(123e)(1282)
3	EXECUTE ANS		0000	00	0000	00	SEE NOTE #15
4	READ ANS		0000	00	0000	00	SEE NOTE #16
5	"CR"		0000	00	0000	10	(110c)
6	"TAB"		0000	00	0000	18	(1111)
7	CONSTANT		10ad	00	0000	00	(1242)
8	1@12		0001	00	0000	00	(126c)(142B)
9	7@3		0e00	00	0000	00	(126d)(128a)
a	12 th → 43		0ff8	00	0000	00	(1287)
b	W ₂		0000	00	0000	00	(1288)(128B)
c	9@6		0240	00	0000	00	(128B)
d	64@18		0001	00	0000	00	(128c)(128e)
e	2@18		0000	08	0000	00	(128d)(128f)
f	CONSTANT		0000	00	0000	0c	(11e7)(15e0)

NAVAL RESEARCH LABORATORY

57

UNCLASSIFIED

NOTE 17: 1138 10B9 101B 131c 1295 11dc 12f1
10dc 1018 11f7 1308 12a2 12ea 141B

CODING SHEET
PASC-901-36-906 (REV. 2-56)

DATE 4/18/62 PROBLEM NO. _____ TAPI NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 21

STOR. AGE LOC. SITH	ORDER PAIR		CODED ORDER PAIR				REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.		
SEE ABOVE 1280	LAR 1284	IR 1284	1284	21	1284	23	(1018)(101B)(11d)(11f7)	
1	LO 1283		0000	10			(1285)(1400) CHECK IF	
1281	A	LAR 1283	126B	50	1283	21	CHARACTER VALID	
1286	PRINTER U	(A)A	126d	24	0000	50	(1285)(1282) IF VALID - PRINT	
1288	CUL 1285	LO(Exit)	1285	14	0000	10	(1280)	
1284	IR 1283	-A	1283	23	126c	55		
1286	CL 1287	RO 1283	1287	12	1283	11		
1281	PRINTER A	(A.A)U	126d	50	1271	26		
8	AR L	UL 5	0001	31	0005	34		
9	U W	W A	128c	43	128c	54		
a	ALL	PCU	0000	41	0000	90		
b	RO 1284		1284	11				
c	w		0000	00	0000	00	(1289)	
1306	d	128d A	LO 12a6	128d	50	12a6	10	MARGINAL TIMING
e	LO 1308		1308	10			ROUTINE ENTRY	
f								
1290	ANS Ad	A LITTLE CLOCK	1250	54	1269	42	ADD ANS TO LITTLE CLOCK	
129c	1	CL 1297	BIG CLOCK A	1297	12	126e	50	JUMP: LITTLE NOT ON
2	CR 1000	ANS Ad	1000	13	1250	54	JUMP: BIG NOT ON	
3	A BIG CLOCK	CL 1295	126e	42	1295	12	ADD ANS TO BIG, JUMP ON	
4	RO 1000		1000	11				
1293	5	1295 A	LO 1280	1295	50	1280	10	PRINT IF VALID
6	RO 1000		1000	11				
1291	7	XU	SCU	0000	71	003f	36	READ CHARACTER
8	STM	U W	0000	84	126f	43	CHECK FOR STOPS	
9	UA	(A.A)U	0000	40	1270	26		
a	XA	CUL 129d	0000	80	129d	14		
129e	b	LITTLE CLOCK A	A	1269	50	1272	54	ADD -6d TO LITTLE
c	A BIG CLOCK	RO 1291	1269	42	1291	11		
129a	d	w A	(A.A)U	126f	50	1271	26	
e	XA	CUL 129B	0000	80	129B	14	JUMP: NOT STOP	
1231	f	BIG CLOCK A	CL 12a3	126e	50	12a3	12	

NAVAL RESEARCH LABORATORY

CODING SHEET
 PERC-881-36-698 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPC NO. 4321
 PROGRAMMER B and E WALD SHEET NO. 22

STORE AGE LOCAT TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
12a5 →	12a0	ANS Ad A BIC	1250	54	126e	42	
	1	CL 12a2 RO 1000	12a2	12	1000	11	
12a1 →	2	12a2 A LO 1280	12a2	50	1280	10	PRINT CHARACTER IF VALID
1094, 117a 129f →	3	12a4 A LAL CLOCK SWITCH	12a4	50	1200	20	TURN OFF CLOCK
	4	RO 1000	1000	11			(12a3)
1230 →	5	BIC A LO 12a0	126e	50	12a0	10	
113, 128a 12ef →	6	LAR EXIT IR EXIT	12a9	21	12a9	23	ROUTINE TO SET UP
	7	14e0 A RU	14e0	50	12Be	24	STORAGE FOR MARGINAL
12ab →	8	CUR 12aa 1001 A	12aa	15	1001	50	TIMING ROUTINE ON
	9	LAL 14ef LO (EXIT)	14ef	20	0000	10	EFFECTIVE 2 (12a6)
12af →	a	*A			12Be	50	
	b	A 14e0 RO 12a8	14e0	42	12a8	11	
12b0 →	c	XU RO 12B1	0000	80	12B1	11	
	d						
	e						
	f						
12B0	CL 12ac	RO 12e3	12ac	12	12e3	11	
12ac →	1	A BIC SCU	1269	42	003f	36	
	2	STM PCU	0000	84	0000	90	
	3	LO 10c9	10c9	10			
109f →	4	HITTLE CLOCK A CL 1132	1269	50	1132	12	
	5	READ Ad CL 12Ba	1274	54	12Ba	12	
	6	EXECUTE Ad A HITTLE	1273	54	1269	42	
	7	CL 12Ba RO 1000	12Ba	12	1000	11	
12Be →	8	SCU STM	003f	36	0000	84	
	9	PCU RO 111a	0000	90	111a	11	
10cc 12b5 →	a	RO 10cB SCU	10cB	11	003f	36	
	b	STM PCU	0000	84	0000	90	
	c	RO 1123	1123	11			
12B7 →	d	XU LO 12B8	0000	80	12B8	10	
	e	CONSTANT	14e1	10	0000	00	(12aa)(12a7)
	f	w	0000	00	0000	00	(12c5)(12c8)

CODING SHEET
 PERC-881-36-698 (REV. 2-58) DATE 4/18/62 PROBLEM NO. TAPC NO. 4321
 PROGRAMMER B and E WALD SHEET NO. 23

STORE AGE LOCAT TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
12c0	50@12	+ 10@36	0032	00	000a	04	(113d)
	1	50@12 + 2@36	0005	00	0002	04	(11f5)
	2	3@12 + 29@42	0002	e0	0000	74	(12c)(1306)
1094 105B →	3	LAR 12ca IR 12ca	12ca	21	12ca	23	COMPUTE EXECUTE TIME
	4	pA AL 17	100d	50	0011	30	00 → 44 @ 18
	5	(R-A)U Uw	1252	26	12Bf	43	
	6	1001 A R Ad	1001	50	1278	54	
	7	(R-A)U UA	1252	26	0000	40	
	8	-w Ad AU	12Bf	55	0000	41	
	9	12c9 A LO TIMER	12c9	50	12d3	10	
	a	EXECUTE Ad LOC	1273	42	0000	10	
1093, 105c 1016 →	b	LAR 12d2 IR 12d2	12d2	21	12d2	23	
	c	1001 A (R-A)U	1001	50	1252	26	
121c →	d	Uw pA	1253	43	100d	50	
	e	AL 17 (R-A)U	0011	30	1252	26	
	f	UA -w Ad	0000	40	1253	55	
12d0	AU	LO 12d1	0000	41	12d1	10	
12d0 →	1	12d1 A LO TIMER	12d1	50	12d3	10	ENTER TIMER @ 12
	2	A READ Ad LOC	1274	42	0000	10	(12cB)(121c)
12d1 12c9 →	3	Uw LAL 12dd	125c	43	12dd	20	TAKE DIFFERENCE BETWEEN
	4	IL 12dd wA	12dd	22	125c	50	SECTORS OF P AND Q
	5	CR 12db R Ad	12db	13	1254	54	AND COMPUTE ACTUAL
12d5 →	6	A w D/A	125c	42	1279	70	NUMBER OF SECTORS
	7	(R-A)U UA	127a	26	0000	40	PASSED : 9B-A
	8	AR 9 Aw	0009	31	127B	42	A: WHOLE 7's
	9	Uw -w Ad	1253	43	1253	51	B: REMAINDER
	a	M R w Ad	1279	60	125c	54	
	b	M 9@0 -w Ad	127c	60	127B	55	
	c	CR 12dd R Ad	12dd	13	127d	54	(12d3)(12d4)
12df 12dc →	d	LOC R Ad	0000	10	127e	55	IF WITHIN 2 SECTORS
	e	CL 12df R Ad	12df	12	127d	54	OR NEGATIVE, ADD 64
12de →	f	R Ad LO 12dd	127e	54	12dd	10	

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRC-REL-36-438 (REV. 2-58)
 DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 24

STOP- AGE LOCAT- TION	ORDER PAIR		CODED ORDER PAIR		REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	
1098 12e0	LITTLE CLOCK A	CL 1470	1269	50 1470 12	
	1 READ ANS Ad	CR 12f5	1274	54 12f5 13	
	2 EXECUTE ANS Ad	A LITTLE CLOCK	1273	54 1269 42	
1300 12B0	3 CL 12f6	BIG CLOCK A	12f6	12 126e 50	
	4 CR 1300	READ ANS Ad	1300	13 1274 54	
	5 A BIG CLOCK	CL 12f1	126e	42 12f1 12	
	6 108d A	CR 12eB	108d	50 12eB 13	
12ec	7 BIG CLOCK A	EXECUTE ANS Ad	126e	50 1273 54	
	8 A BIG CLOCK	CL 12ea	126e	42 12ea 12	
	9 RO 1000		1000	11	
12e8 10cd	a 12ea A	LO PRINT IF VALID	12ea	50 1280 10	
12eb	b RO 1000	BIG CLOCK A	1000	11 126e 50	
	c A Ad	CL 12ef	12e2	54 12ef 12	
	d 12ed A	LO LOGICAL SUM	12ed	50 1315 10	
	e LO 12e7		12e7	10	
12ec	f 12ef A	LO MARGINAL TIMING	12ef	50 12a6 10	
12f0	LO 10dc		10dc	10	
12e5	1 12f1 A	LO PRINT IF VALID	12f1	50 1280 10	
	2 108d A	CL 10ce	108d	50 10ce 12	
10cf	3 RO 1000	A	1000	11 10ef 50	
	4 EXECUTE ANS Ad	A BIG CLOCK	1273	54 126e 42	
12e1	5 LO 10cd	LO 1077	10cd	10 1077 10	
12e3 1078	6 XU	SCU	0000	71 003f 36	
	7 STM	LO 10e9	0000	84 10c9 10	
12d8 10cB	8 () A	CUL 12fc	0000	50 12fc 14	(12fc)(10ca)
10ec	9 BIG CLOCK A	CR 12fa	126e	50 12fa 13	
	a LO 1302	108d A	1302	10 108d 50	
	b CL 130f	LO 112d	130f	12 112d 10	
12f1	c 11 12f8	A Ad	12f8	22 11fB 55	
	d CL 10eB	LO 12f8	10eB	12 12f8 10	
10eB	e LITTLE CLOCK A	A Ad	1269	50 1267 54	
	f EXECUTE ANS Ad	A LITTLE CLOCK	1273	54 1269 42	

CODING SHEET
 PRC-REL-36-438 (REV. 2-58)
 DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 25

STOP- AGE LOCAT- TION	ORDER PAIR		CODED ORDER PAIR		REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	
12e4 1300	RO 12e3	108d A	12e3	11 108d 50	
1098 12fa	1 CL 130f	RO 1000	130f	12 1000 11	JUMP: P "nnx"
	2 READ ANS A	BIG CLOCK Ad	1274	50 126e 54	
1472	3 A BIG CLOCK	CL 1308	126e	42 1308 12	
1304	4 108d A	CR 1305	108d	50 1305 13	JUMP: P "nnx"
	5 LO 1135	BIG CLOCK A	1135	10 126e 50	
	6 12c2 Ad	CL 128d	12c2	54 128d 12	
	7 LO 113e		113e	10	
1303 128e	8 1308 A	LO PRINT IF VALID	1308	50 1280 10	PRINT CHAR. IF VALID
	9 108d A	CL 130f	108d	50 130f 12	
131e	a A	LAL CLOCK SWITCH	101d	50 1200 20	TURN OFF CLOCK SWITCH
	b RO 1000		1000	11	
1318	c 1481 A	A 1199	1481	50 1199 42	
	d 1480 A	A 119d	1480	50 119d 42	
	e LO 1406		1406	10	
14c, 1309 13d, 12f3	f 10ef A	LO 1129	10ef	50 1129 10	
1310	1310	READ ANS Ad	126e	50 1274 54	
1098	1 A BIG CLOCK	CL 131c	126e	42 131c 12	
	2 108d A	CR 113a	108d	50 113a 13	
	3 LO 1135		1135	10	
	4				
113e 12e2	5 LAR EXIT	IR EXIT	1319	21 1319 23	LOGICAL SUM LOOP
	6 W A	(W.A)U	108f	50 126d 26	TAKES LOGICAL SUM
	7 PRINTER A	U PRINTER	126d	50 126d 43	OF PRESENT P ORDER
	8 PRINTER A	W Ad	126d	55 108f 54	AND THE 6 BITS IN
	9 PRINTER	LOC	126d	42 0000 10	THE PRINTER AND
1394	a STOP	RO 1000	0006	82 1000 11	STOWS RESULT IN
141f	b A 1193	LO 130c	1193	42 130c 10	PRINTER
1311	c 131c A	LO PRINT IF VALID	131c	50 1280 10	PRINT CHAR. IF VALID
	d 108d A	CR 1127	108d	50 1127 13	
	e LO 130a		130a	10	
	f				

NAVAL RESEARCH LABORATORY

NOTE 18: 10f8 1353 1340 14e9 139a 136d 1420
 135c 1345 14e1 14f7 1360 140a

CODING SHEET
 PASC-REL-24-486 (REV. 7-58) DATE 4/18/62 PROGRAMER B. and E. WALD SHEET NO. 26
 TITLE _____

STOP- AGE LOCATI- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
SEE NOTE #18 1320	LAL 1321	IL 1321	1321	20	1321	22	LOOP TO PRINT MESSAGES
1320a	() A	-#LL	0000	50	133f	24	(1329)(1320)(1322)
	CUL 1325	IL 1321	1325	14	1321	22	EXISTS WITH ffff IN
	LAL 1324	LO 1324	1324	20	1324	10	LEFT ADDRESS
1323	4	LO(Exit)	0000	10			(1323)
1326 1327	5	LAR 1326	A A	1326	21	133e	50
	6	A ctr () A	133d	42	0000	50	(1325)(132e)
1328	7	YL AL 6	0000	71	0006	30	
	8	AW -#A	133c	42	133B	50	
	9	CUL 132B	IL 1321	132B	14	1321	22
	a	LO 1321		1321	10		
1329	b	PCU IL ctr	0000	90	133d	22	
	c	CL 132e	w, A	132e	12	133c	50
	d	LO 1327		1327	10		
132c	e	IR 1326	RO 1325	1326	23	1325	11
	f	32	0800	00	0000	00	(11e8)(11eB)(11ec)(1439)(1484)
1330		16	0400	00	0000	00	(1436)
	1	8	0200	00	0000	00	(1433)
	2	4	0100	00	0000	00	(1430)(14c8)
	3	BREAKPOINTS	0f00	00	0000	00	(1415)
	4	TRANSFER CONTROL	0080	00	0000	00	(1408)(14cc)
	5	FLEX COPY INPUT	0002	00	0000	00	(138c)
	6	PRINT STOP CODE	0004	00	0000	00	(1369)
	7	4-6 INPUT SWITCH	0020	00	0000	00	(1356)
	8	OUTPUT	0008	00	0000	00	(137f)
	9	MANUAL INPUT SWITCH	0040	00	0000	00	(1348)
	a	INPUT	0010	00	0000	00	(1343)
	b	CODE DELETE @44	0000	00	0000	3f	(1328)(1485)
	c	W,	0000	00	0000	00	(1328)(132c)
	d	COUNTER	0000	00	0000	00	(1326)(132B)
	e	-7@12	fff9	00	0000	00	(1325)
	f	END OF MESSAGE MARK	ffff	00	0000	00	(1321)

NAVAL RESEARCH LABORATORY

61

UNCLASSIFIED

CODING SHEET
PNC-REL-38-488 (Rev. 2-54)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 27

STORE AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR			REMARKS		
	LEFT ORDER	RIGHT ORDER	ORDER NO.	ADDRESS	ORDER NO.			
1340	1340A	LO 1320	1340	50	1320	10		
1			13c0	00	0000	00	PRINT HEADING	
2			ffff	00	0000	00		
3	CONTROL A	(R.A)U	100e	50	133a	26	JUMP IF PHOTO INPUT	
4	XA	CUL 1360	0000	80	1360	14		
5	1345 A	LO 1320	1345	50	1320	10		
6			13c7	00	0000	00	PRINT: "FLEKOWRITER,	
7			ffff	00	0000	00	HANDL. INPUT BUTTON	
8	CONTROL A	(R.A)U	100e	50	1339	26		
9	XA	CUL 1369	0000	80	1369	14	JUMP: MAN. INPF. SW. DOWN	
a	LO 134B		134B	10				
134a →	b	XA	A 11c3	1451	50	11c3	42	
	c	XA	A 11b8	1452	50	11b8	42	
	d	XA	A 1099	1453	50	1099	42	
	e	LO 134f		134f	10			
134e →	f	XA	LAL 1098	0000	80	1098	20	
1350	LO 1353		1353	10				
135B →	1	A 114f	XA	114f	42	147c	50	
134B →	2	A 115B	LO 135c	115B	42	135c	10	
1350 →	3	1353 A	LO 1320	1353	50	1320	10	
4			13cc	00	0000	00	PRINT: "UP"	
5			ffff	00	0000	00		
137E	6	CONTROL A	(R.A)U	100e	50	1337	26	
137E	7	XA	CUL 1364	0000	80	1364	14	JUMP: 6 BIT INPUT
	8	XA	A 135d	138a	50	135d	42	
	9	XA	A 1051	1457	50	1051	42	
	a	XA	A 1164	1458	50	1164	42	
	b	XA	LO 1351	1459	50	1351	10	
135L →	c	135c A	LO 1320	135c	50	1320	10	
	d	135e		0000	00	0000	00	PRINT: "4 (OR 6) BIT"
	e			ffff	00	0000	00	
	f	LO 137f		137f	10			

CODING SHEET
PNC-REL-38-488 (Rev. 2-54)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 28

STORE AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR			REMARKS		
	LEFT ORDER	RIGHT ORDER	ORDER NO.	ADDRESS	ORDER NO.			
1344	1360A	LO 1320	1360	50	1320	10		
1			13ce	00	0000	00	PRINT: "PHOTOREADER"	
2			fff	00	0000	00		
3	LO 137a		137a	10				
1357 →	4	XA	A 135d	1389	50	135d	42	
	5	XA	A 1051	145a	50	1051	42	
	6	XA	A 1164	145B	50	1164	42	
	7	XA	A 114f	145c	50	114f	42	
	8	XA	LO 1352	147B	50	1352	10	
1349 →	9	CONTROL A	(R.A)U	100e	50	1336	26	
	a	XA	CUR 1378	0000	80	1378	15	JUMP: PRINT STOP
	b	XA	LO 13a6	1386	50	13a6	10	
1379 →	c	XA	A 1370	13a5	50	1370	42	
13a7 →	d	136d A	LO 1320	136d	50	1320	10	
	e			13cd	00	0000	00	PRINT: "DOWN, STOP"
	f			13d3	00	0000	00	CODE IS (NOT)
	g			1387	00	0000	00	PRINTED (136c)(13a6)
1370	1			13d5	00	0000	00	
2				fff	00	0000	00	
3	LO 1374			1374	10	0000	00	
1373 →	4	XA	A 11c3	1461	50	11c3	42	
	5	XA	A 1168	1462	50	1168	42	
	6	XA	A 1099	1463	50	1099	42	
	7	XA	LAL 1098	0000	80	1098	20	
136a →	8	LO 1356	XA	1356	10	1450	50	
	9	A 115c	LO 136c	115c	42	136c	10	
13a3 →	a	XA	A 11c3	1465	50	11c3	42	
	b	XA	A 1168	1466	50	1168	42	
	c	XA	A 1099	1467	50	1099	42	
	d	LO 1374	LAL 1098	101d	50	1098	20	
	e	LO 1356		1356	10			
135f →	8	CONTROL A	(R.A)U	100e	50	1338	26	

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRC-REL-34-498 (Rev. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAP NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 29

STORAGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1380	X A	CUL 1400	0000	80	1400	14	JUMP: EAST OUTPUT
1	LO 1382		1382	10			
1381	A A	A 1180	146a	50	1180	42	
3	A A	A 109F	146B	50	109F	42	
4	LO 1385		1385	10			
1384	A A	A 1281	146d	50	1281	42	
6	LO 1387		1387	10			
1386	A A	A 1231	146f	50	1231	42	
8	A A	1098 Ad	138a	50	1098	54	
9	A 1098	LO 138c	1098	42	138c	10	
a			0001	00	0000	00	(1388)
b							
1387	CONTROL A	(A A) L	100e	50	1335	26	
d	X A	CUL 1396	0000	80	1396	14	JUMP: COPY INPUT
1228	LO 1225	A 139d	1225	10	139d	42	
f	A A	A 119d	1480	50	119d	42	
1390	A A	A 1199	1481	50	1199	42	
1	A A	RO 1399	1482	50	1399	11	
14ec	-1254 A	A 124e	1254	51	124e	42	PRINTS FEEDS
1395	XU	PCU	0000	71	0000	90	
4	IL 124e	CL 131a	124e	22	131a	12	
5	RO 1393		1393	11			
139d	A A	A 139d	1381	50	139d	42	
1049	LO 13d5	A 119d	10d5	10	119d	42	
8	A A	A 1199	1484	50	1199	42	
1391	A A	A 1193	1485	50	1193	42	
a	139a A	LO 1320	139a	50	1320	10	
b			13d8	00	0000	00	PRINT: " OUTPUT --
c			13d8	00	0000	00	FLUORESCENTER
d			13d8	00	0000	00	(1396)(138e)
e			fff	00	0000	00	
f	LO 140e		140e	10			

CODING SHEET
 PRC-REL-34-498 (Rev. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAP NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 30

STORAGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
13a0	A A	A 1428	13a9	50	1428	42	
1	A A	A 1065	1486	50	1065	42	
2	A A	A 10d1	1487	50	10d1	42	
3	A A	A 1075	1488	50	1075	42	
4	A A	RO 1414	1489	50	1414	11	
5	15		13e7	00	0000	00	(136c)(14e8)
136d	A 1370	A A	1370	42	136f	50	
7	A 115c	LO 136d	115c	42	136d	10	
8							
9	EXTENDED		13f7	00	0000	00	(13a0)
a	NORMAL		13f4	00	0000	00	(1490)
b	-22		13f3	00	0000	00	(1446)
c	-16		13f2	00	0000	00	(1444)
d	-4		13f0	00	0000	00	(1440)
e	-8		13f1	00	0000	00	(1442)
f	NONE		13ef	00	0000	00	(1417)
13B0							
1	DUMMY ADDRESS		13e8	00	0000	00	(139f)
2			0B61	eB	1c24	84	copy
3			0a2c	c3	4da1	a8	input
4			0c93	cB	6caa	fc	delete
5	INPUT COPY DELETED		1382	00	0000	00	(1229)
6	NOT		13d7	00	0000	00	(136b)(143f)
7							
8							
9	6 BIT		13d2	00	0000	00	(1364)
a	4 BIT		13d1	00	0000	00	(1358)
b	PHOTOREADER						
c	DOWN		13cd	00	0000	00	(1411)
d	UP		13cc	00	0000	00	(1410)
e							
f			1160	10	0000	00	(13a6)

NAVAL RESEARCH LABORATORY

63

UNCLASSIFIED

CODING SHEET
PRC-NR-34-538 (REV. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TIME NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 31

STORAGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
13c0			0208	30	4f86	f0	CR CR TAB UC S T A
	STATUS OF		02B4	fa	1c55	84	T U S SP O F SP
2	SIMULATED L&P-30		0fa8	BB	40fc	d8	S I M U L A T
3	(R) INPUT --		0caa	84	35f0	08	E D SP L G P Lc
4			08c7	04	8128	08	- 3 O CR UC Lc
5			0B30	d3	6863	8c	N P U T SP --
6			087f	00	0000	00	SP 3f
7			0115	08	3cB3	c4	UC F K L E X D
8	FLEXWRITER,		07e6	a3	6ca6	B4	W R I T E R
9	MANUAL INPUT		086e	f2	cd3c	0c	SP M A N U A L
a	BUTTON		0868	B3	0d36	84	SP I N P U T SP
b			08B4	dB	6c6c	fc	B U T T O N SP
c	UP		0874	c2	1fc0	00	SP U P SP 3f
d	DOWN		086a	c5	fB21	fc	SP D O W N SP 3f
e	PHOTOREADER		0130	0B	8c76	c4	UC P Lc H O T O
f			09B2	f2	aca6	B4	R E A D E R
13d0			087f	00	0000	00	SP 3f
1	4 BIT		0261	8a	8d8f	00	4 SP BIT SP 3f
2	6 BIT		0361	8a	8d8f	00	6 SP BIT SP 3f
3	STOP CODE		0B61	fB	6c70	84	SP S T O P SP
4			0EB1	aB	287f	00	C O D E SP 3f
5	PRINTED		0c26	a2	cdB2	a8	P R I N T E D
6			087f	00	0000	00	SP 3f
7	NOT		0B31	dB	f000	00	N O T 3f
8	OUTPUT --		0204	c4	2d36	c0	CR UC O K U T P
a	FAST PUNCH		0d36	86	38e1	fc	U T SP -- SP 3f
b			0115	0B	cfB6	84	UC F K A S T SP
c	FLEXWRITER		0c34	B3	ae3f	00	P U N C H 3f
d			0115	08	3cB3	c4	UC F K L E X D
e			07e6	a3	6ca6	fc	W R I T E R 3f
f							

CODING SHEET
PRC-NR-34-538 (REV. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TIME NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 32

STORAGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
13e0	OTHER	CONTROLS--	0204	c4	2dB8	c8	CR UC O K T H E
1	TRANSFER	CONTROL	09a1	13	a0B1	BD	R SP UC C Lc O N
2	BUTTON	IS	0da6	c4	3fa1	8c	T R O L S SP -
3			08e1	13	60a6	f0	- SP UC T I C R A
4			0B3e	57	29a1	e8	N S F E R SP C
5			0c6c	da	6c43	84	O N T R O L SP
6			08B4	dB	6c6c	84	B U T T O N SP
7			0a3e	fc	0000	00	I S 3f
8			0fc0	00	0000	00	3f
9	DEPRESSED		0221	86	1861	84	CR SP SP SP SP SP SP
a	BREAKPOINT	SWITCHES	012a	0B	2c26	c8	UC D Lc E P R E
b			0fBe	ca	a862	98	S S E D SP B R
c			0cBc	6f	0c68	BD	E A K P O I N
d			0da1	f9	fa36	e8	T SP S W I T C
e			0e32	fa	18ff	00	H E S SP - 3f
f	NONE		0B31	B3	2fc0	00	N O N E 3f
13f0	-4		08c9	fc	0000	00	- 4 3f
1	-8		08d1	fc	0000	00	- 8 3f
2	-16		08c3	37	f000	00	- 1 6 3f
3	-32		08c7	17	f000	00	- 3 2 3f
4	NORMAL		0221	86	1861	84	CR SP SP SP SP SP SP
5			012c	0B	19ae	f0	UC N Lc O R M A
6			00e1	fc	0000	00	L SP 3f
7			0221	86	1861	84	CR SP SP SP SP SP SP
8	EXTENDED		0132	0B	3d82	BD	UC E Lc X T E N
9			0aB2	a2	1fc0	00	D E D SP 3f
a	PRECISION		0c26	cB	aa3e	a0	P R E C I S I
b			0c6c	20	823f	00	O N CR CE CR 3f
c			0204	c4	2Bf2	98	CR UC O Lc V E R
d	OVERFLOW		0543	c5	f23f	00	F L O W 3f
e							
f							

NAVAL RESEARCH LABORATORY

CODING SHEET
PENC-REL-36-438 (REV. 2-54)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B and E. WALD SHEET NO. 33

STOP-AGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1320 →	LO 103a		103a	10			
103c →	1 -kA	A 11d0	1474	50	11d0	42	
	2 -kA	A 109f	1475	50	109f	42	
	3	LO 1404	1404	10			
1403 →	4 -kA	A 1281	1477	50	1281	42	
	5 -kA	LO 131B	1482	50	131B	10	
130e →	6 -kA	A 1231	1478	50	1231	42	
	7	LO 140a	140a	10			
	8						
	9						
1407 →	a	140a A LO 1320	140a	50	1320	10	
	b		13d8	00	0000	00	OUTPUT -- FAST PUNCH
	c		13da	00	0000	00	
	d		ffff	00	0000	00	
131f →	e	CONTROL A (k.A)U	100e	50	1334	26	
	f	XA CUL 1411	0000	80	1411	14	JUMP - T.C. DOWN
1410	1 -kA	LO 144B	13Bd	50	144B	10	
140f →	2 -kA	LO 1448	13Bc	50	1448	10	
1494, 1444, 144c →	3	CONTROL A CL 13a0	100e	50	1490	12	
	4	LO 1490	13a0	10			
13a4 →	5	A 10e3			10e3	42	
1494 →	6	CONTROL A (k.A)U	100e	50	1333	26	
	7	XA CUL 142f	0000	80	142f	14	JUMP - SOME DEPT. DOWN
	8	kA A 1424	13af	50	1424	42	
	9	kA A 1425	13B1	50	1425	42	
	a	A 1426	A 1427	42	1427	42	
	b	LO 143d	143d	10			
1471 →	c	141B A LO 1280	141B	50	1280	10	
	d	108d A CL 130f	108d	50	130f	12	
	e	LO 112d	112d	10			
	f						

CODING SHEET
PENC-REL-36-438 (REV. 2-54)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B and E. WALD SHEET NO. 34

STOP-AGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1447 →	1420	A LO 1320	1420	50	1320	10	
143c →	1		13e0	00	0000	00	OTHER CONTROLS -- TRANSFER
143e	2		13e2	00	0000	00	(1448) (1449) CONTROL IS UP (DOWN)
	3		13e9	00	0000	00	DEPRESSED BREAKPOINT SWITCH
	4	LEFT ADDRESS MAY BE 13ef, 13fo, or 13e8	0000	00	0000	00	14 (1417)(1433)(1440)
	5		13e8	00	0000	00	15 (1418)(1435)(1442)
	6		13e5	00	0000	00	16 (1419)(1438)(1444)
	7		13e8	00	0000	00	32 (1419)(1438)(1446)
	8		13e4	00	0000	00	NORMAL (EXTENDED) (13ad)(149)
	9		13fa	00	0000	00	PRECISION
	a		ffff	00	0000	00	
	b	100e A (k.A)U	100e	50	1278	26	
	c	XA CUL 14e7	0000	80	14e7	14	JUMP - TRACE
	d	kA RD 143e	1496	50	143e	11	
	e	CONSTANT	101f	42	1011	10	(142f)
1416 →	f	kA A 1010	142e	50	1010	42	
1430	1A	(k.A)U	0000	40	1332	26	CHECK ON EACH
	2	XA CUL 1440	0000	80	1440	14	INDIVIDUAL BREAKPOINT SWITCH
	3	kA A 1424	13B1	50	1424	42	
1441 →	4	100e A (k.A)U	100e	50	1331	26	
	5	XA CUL 1442	0000	80	1442	14	
	6	kA A 1425	13B1	50	1425	42	
1443 →	7	100e A (k.A)U	100e	50	1330	26	
	8	XA CUL 1444	0000	80	1444	14	
	9	kA A 1426	13B1	50	1426	42	
1445 →	a	100e A (k.A)U	100e	50	132f	26	
	b	XA CUL 1446	0000	80	1446	14	
	c	kA A 1427	13B1	50	1427	42	
	d	LO 1420	1420	10			
141a →	e	kA A 1010	144f	50	1010	42	
142a →	f	LO 1420	A 1001	1420	10	1000	42
	g	kA RO 14e8	13B6	50	14e8	11	

NAVAL RESEARCH LABORATORY

65

UNCLASSIFIED

CODING SHEET
PRC-REL-38-486 (Rev. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 35

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1431 → 1440	RA	A 1424	13ad	50	1424	42	
1	LO 1433		1433	10			
1434 → 2	RA	A 1425	13ae	50	1425	42	
3	LO 1436		1436	10			
1437 → 4	RA	A 1426	13ac	50	1426	42	
5	LO 1439		1439	10			
1438 → 6	RA	A 1427	13aB	50	1427	42	
7	LO 1420		1420	10			
1441 → 8	RA	A 1422	1422	42	145d	50	
9	A 10c2	LO 1412	10c2	42	1412	10	
a							
1440 → b	RA	A 1422	1422	42	1456	50	
c	A 10c2	LO 1412	10c2	42	1412	10	
d							
e							
f			101f	42	1015	10	(143d)
1450			1162	10	0000	00	(137f)
1			1269	50	11d2	12	(134B)
2			104B	42	1181	11	(134c)
3			109e	12	109a	10	(134d)
4							
5							
6			1200	10	0000	00	(144B)
7			104f	43	1052	10	(135f)
8			0000	90	1165	10	(135a)
9			1165	12	114c	10	(135B)
a			104f	43	1055	10	(1365)
b			0000	90	1169	10	(1366)
c			1169	12	114c	10	(1367)
d			10B0	10	0000	00	(1448)
e							
f							

CODING SHEET
PRC-REL-38-486 (Rev. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 36

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1460							
1			1269	50	11e0	12	(1374)
2			104B	42	1172	10	(1375)
3			109e	12	11e3	10	(1376)
4							
5			1269	50	1140	12	(137a)
6			104B	42	1141	11	(137B)
7			109e	12	1000	11	(137c)
8							
9							
a			11d1	10	0000	00	(1382)
b			110B	10	0000	00	(1383)
c							
d			1282	10	0000	00	(1385)
e							
f			1232	10	0000	00	(1387)
12e0 ↳ 1470	Big Clock A	READ ANS A d	126e	50	1274	54	
1	A Big Clock	CL 141B	126e	42	141B	12	
2	LO 1304		1304	10			
3							
4			11d2	10	0000	00	(1401)
5			1114	10	0000	00	(1402)
6							
7			1287	10	0000	00	(1404)
8			1290	10	0000	00	(1406)
9							
a							
b			126a	50	1169	14	(1368)
c			126a	50	1165	14	(1351)
d			14ff	10	0000	00	(100a)
e							
f							

NAVAL RESEARCH LABORATORY

NOTE 19: 14e4 14c3 11e8 14B7 14Ba 14Bc 14c0
14d0 14B1 11ee 11ea 11f0 11eB 11ed

If use new coding add:

NOTE 20: 1614 1624 1627 162f 1630 1653

NOTE 21: 17a6 17dd 17e7 16B9 1710 171c 175c 1768
17a8 17df 17e5 16BB 1712 171e 175e 17a2
17aa 17e1 17e3 16Bd 1714 171f 1760 17a4
17ac 17ec 16B2 16Bf 1716 1750 1762
17B5 17ee 16B4 16f1 1718 1754 1764
17dB 180B 16B7 170d 171a 175a 1766

NOTE 22: 1804 1794 17cd 1708 1758 16c8 175f
1805 1795 17d0 1721 1763 16f0 1761
17f0 1798 17d1 1722 16d5 170a 1765
17f1 1799 17d3 1727 16aB 175B
17d9 17cc 17d8 174e 16B0 175d

NOTE 23: 17e0 17e6 182B 17B1 16B3 16ef 16f5 170c
17e2 17e9 179a 17dc 16Be 16f0 16f9 1767
17e4 182a 179f 17de 16c0 16f2 170B 1768

NOTE 24: 16B5 16B8 16f3 16f4 170e 171d 179c 17a2

NOTE 25: 16B6 16Bc 1711 1715 1719 179f
16Ba 170f 1713 1717 171B 17B3

NAVAL RESEARCH LABORATORY

67

UNCLASSIFIED

CODING SHEET
PRC-REL-36-406 (REV. 2-56)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 37

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1480			0000	84	1160	10	(1308)(1384)
1			1164	13	1196	10	(1302)(1390)
2			1194	14	1181	11	(1391)(1405)
3			0000	84	1162	10	(1087)
4			119B	12	1196	10	(1398)
5			1194	14	119a	10	(1399)
6			1067	10	0000	40	(13a1)
7			1084	10	100c	50	(13a2)
8			1061	54	0000	71	(13a3)
9			10e6	10	10df	50	(13a4)
a			0000	71	0000	71	(1491)
b			0000	71	100c	50	(1492)
c			1061	54	1066	11	(1493)
d			0000	71	10df	50	(1494)
14a1 → e	"ce" U	PCU	11B1	24	0000	90	
f	100e A	RO 14a1	100e	50	14a1	11	
1490 → 1	A A	A 1428	13aa	50	1428	42	
2	A A	A 1065	148a	50	1065	42	
3	A A	A 10d1	148B	50	10d1	42	
4	A A	A 1075	148c	50	1075	42	
5	A A	RO 1414	148d	50	1414	11	
6			14e0	10	1498	10	(14e7)
7			14e0	10	1001	22	(1428)
1000 → 8	1498 A	LO 14a0	1498	50	14a0	10	
9	IL 1001	LO 1001	1001	22	1001	10	
a			0011	00	0000	00	(14cB)
b			0000	00	0000	20	(14d7)
c	L W	J	0000	00	0000	00	SEE NOTE #19
d	"0"		0000	00	0000	01	
e	"8"		0000	00	0000	11	
f	"SPACE"		0000	00	0000	21	(14a7)(14a2)(14b2)(14d2)

CODING SHEET
PRC-REL-36-406 (REV. 2-56)

DATE 4/18/62 PROBLEM NO. _____ TAPE NO. _____
PROGRAMMER _____ SHEET NO. 38

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
10FB → 1491 → 14a	LAL EXIT	IL EXIT	14Bd	20	14Bd	22	
1491 → 1	LO 148e	CL 14a3	148e	10	14a3	12	
14a2 → 2	A A	RO 14a3	124f	50	14a3	11	
14a1 → 3	A A	A cTR	125f	50	14df	42	
4	a A	(A)U	100d	50	102f	26	
5	A A	CU 14d3	1266	50	14d3	14	WALD: NOT 'T
6	a A	CL 14ce	100c	50	14ce	12	IF 'T' WITH NEGATIVE
14cf → 7	"sp" U	PCU	149f	24	0000	90	d DO NOT PRINT
14ad → 8	LO 14ac	PCU	14ac	10	0000	90	
14d2 → 9	PCU	PCU	0000	90	0000	90	
a	PCU	a A	0000	90	100d	50	
b	CR 14ae	LO 11e7	14ae	13	11e7	10	
14a3 → c	PCU	PCU	0000	90	0000	90	
d	RO 14a8		14a8	11			
14ab → e	"sp" U				149f	24	
f	PCU	PCU	0000	90	0000	90	
14B0	PCU	AL 17	0000	90	0011	30	SHIFT @ LEFT 17
14f1 → 1	Aw	LO 14d5	149c	42	14d5	10	PLACES, PRINT ORDER
14df → 2	14B2 A	LO 148e	1482	50	148e	10	
3	"sp" U	PCU	149f	24	0000	90	
4	PCU	PCU	0000	90	0000	90	
5	XU	a A	0000	71	100c	50	PRINT @
6	CL 14B7	AU	1487	12	1271	24	
14B6 → 7	AL 3	Aw	0003	30	149c	42	
14B2 → 8	A A	AL 6	132f	50	0001	30	
9	PCU	IL cTR	0000	90	14df	22	
a	CL 14Bd	w A	14Bd	12	149c	50	
b	XU	AL 4	0000	71	0004	30	
c	Aw	LO 1488	149c	42	14B8	10	
14B4 → d	LO (EXIT)		0000	10			
14c5, 14d1 → e	LAL 14c5	IL 14c5	14c5	20	14c5	22	
14B2 → f	A A	LAR 14cd	14cd	50	14cd	21	

NAVAL RESEARCH LABORATORY

COODING SHEET
 PERC-REL-34-650 (REV. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 39

STOR- AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
14c0	wA	(K.A)U	149c	50	14d	26	
1	UA	LO 14c6	0000	40	14c6	10	
2	KA	LAR 14cd	14dc	50	14cd	21	
3	wA	AL 6	149c	50	0006	30	
4	LO 14c6		14c6	10			
5	(EXIT)		0000	10			(14Be)
14c4	ARL	D/10@4	0001	31	14dB	70	
14c1	(K.A)U	Uw	14da	26	14d9	43	
8	wAd	KA	14d9	55	1332	54	
14d2	XU	AL 4	0000	71	0004	30	
a	PCU	wA	0000	90	14d9	50	
b	M/10@4	KA	14dB	60	149a	54	
c	XU	AL 8	0000	71	0008	30	
14Bf	PCU	LO ()	0000	90	0000	10	(14e2)
14ab	100e A	(K.A)U	100e	50	1334	26	
f	XA	CUL 14a7	0000	80	14a7	14	JUMP: T.C. DOWN
14d3	KA	Aw	1001	50	149c	42	
1	14d1 A	LO 14Be	14d1	50	14Be	10	
2	"sp" U	LO 14a9	14af	24	14a9	10	
14e5	"U" A	CUL 14d0	1263	50	14d0	14	
4	LO 14a7		14a7	10			
14B1	UA	(K.A)U	0000	40	133B	26	
6	UA	ARL	0000	40	0001	31	
7	KA	AL	149B	54	0000	41	
8	PCU	LO 14B2	0000	90	14B2	10	
9	wAd		0000	00	0000	00	(14e7)(14c8)(14ca)
a			01ff	ff	ffff	ff	(14c7)
b	10@4		0a00	00	0000	00	(14c6)(14cB)
c			14c5	00	0000	00	(14c2)
d	TRACK @6		0fc0	00	0000	00	(14c0)
e			14c2	00	0000	00	(14BF)
f	CTR.		0000	00	0000	00	(14a3)(14B9)

COODING SHEET
 PERC-REL-34-650 (REV. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 40

STOR- AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
100B 1000	LO (14Bf)		0000	10			(12a7)(12aB)
14e0	14e1 A	LO 1320	14e1	50	1320	10	
2			14f0	00	0000	00	
3			ffff	00	0000	00	
4	14ef A	A w	14ef	50	149c	42	
5	14e5 A	LO 14Be	14e5	50	14Be	10	
6	LO 14F7		14f7	10			
142c	KA	A 1000	1495	50	1000	42	
143f	KA	A 14eB	13a5	50	14eB	42	
9	14e9 A	LO 1320	14e9	50	1320	10	
a			14fB	00	0000	00	
b			13a7	00	0000	00	(14e8)
c			14fc	00	0000	00	
d			ffff	00	0000	00	
e	LO 1392		1392	10			
f	(STOW MARGINAL "P")		0000	00	0000	00	(14e4)(12a9)
14f0			0230	9a	8B36	84	CR PRINT SP
1			0c66	aB	29a1	a0	ORDER SP 1
2			0B21	0f	1eBc	d8	N SP LOC AT
3			0a31	B2	1fc0	00	TON SP 3f
4			086e	f2	65e8	30	SP MARG IN
5			0f03	0e	4876	a0	ALL Y SP T I
6			0BB2	aB	f000	00	M E D 3f
14e6	14f7 A	LO 1320	14f7	50	1320	10	
8			14f4	00	0000	00	
9			fff	00	0000	00	
a	STOP		0001	82	1000	11	
b			0dab	f3	0ca1	fc	TRACE SP 3f
c			0872	cf	2eB4	d8	SP EX EC UT
d			0caa	20	8208	fc	ED CR CE CR CE 3f
e							
14e0	STOP		0001	82	1000	11	

NAVAL RESEARCH LABORATORY

69

UNCLASSIFIED

CODING SHEET
 PRAC-NRL-58-456 (Rev. 2-58)
 DATE 4/18/62 PROBL. NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 41

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1500			0001	a0	3580	00	C 0143
1			0001	00	1600	00	P 0044
2			0000	80	0000	00	I 0000
3			0001	60	3000	00	T 0132
4			0001	e0	1780	00	S 0047
5			0001	60	1100	00	T 0034
6			0001	e0	1880	00	S 0049
7			0001	60	3100	00	T 0134
8			0001	e0	3600	00	S 0144
9			0001	60	3f00	00	T 0162
a			0001	40	0580	00	U 0011
b			0001	e0	1780	00	S 0047
c			0001	60	2680	00	T 0113
d			0001	e0	1880	00	S 0049
e			0001	60	4480	00	T 0209
f			0001	e0	3600	00	S 0144
1510			0001	60	3a80	00	T 0153
1			0001	40	0900	00	U 0018
2			0001	e0	1780	00	S 0047
3			0001	60	5800	00	T 0248
4			0001	80	1380	00	H 0039
5			0001	20	2000	00	E 0100
6			0000	c0	3280	00	N 0137
7			0001	e0	0100	00	S 0002
8			0001	a0	4180	00	C 0203
9			0001	a0	5f80	00	C 0263
a			0000	20	1800	00	B 0048
b			0001	20	1380	00	E 0039
c			0001	40	0f80	00	U 0031
d			0000	00	0080	00	Z 0001
e			004f	Bf	Bf80	00	,027KWKWJ
f			0001	c0	4180	00	A 0203

CODING SHEET
 PRAC-NRL-58-456 (Rev. 2-58)
 DATE 4/18/62 PROBL. NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 42

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1520			0001	40	1B00	00	U 0054
1			0000	00	0000	00	Z 0000
2			0000	60	1f80	00	R 0063
3			0001	40	1900	00	U 0050
4			0001	e0	2080	00	S 0101
5			0001	60	2100	00	T 0102
6			0001	40	2280	00	U 0105
7			0000	00	0000	00	Z 0000
8			0000	00	0000	00	Z 0000
9			0000	60	1f80	00	R 0063
a			0001	40	1900	00	U 0050
b			0001	e0	2080	00	S 0101
c			0001	60	2180	00	T 0103
d			0001	c0	4880	00	A 0217
e			0001	40	2280	00	U 0105
f			0200	00	0000	00	,1000 0000
1530			0000	07	ff80	00	Z 6363
1			0200	00	0000	00	,1000 0000
2			0000	c0	0e80	00	N 0029
3			0001	20	0f00	00	E 0030
4			0001	80	2500	00	H 0110
5			0001	40	1c80	00	U 0057
6			0000	40	4A00	00	Y 0226
7			0001	80	4c80	00	H 0225
8			0001	40	4800	00	U 0222
9			0001	20	2e80	00	E 0129
a			0000	e0	2f00	00	M 0130
b			0001	c0	2500	00	A 0110
c			0001	80	1380	00	H 0039
d			0001	40	3d00	00	U 0158
e			0000	00	0040	00	,0000 0002
f			0001	40	4000	00	U[0200]

NAVAL RESEARCH LABORATORY

CODING SHEET
 PERC-981-36-958 (REV. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 43

STOR- AGE LOC- ATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1540			00ff	e0	0000	00	,07ww 0000
1			0012	00	0000	00	,0090 0000
2			0001	c0	3680	00	A 0145
3			0001	20	5700	00	E 0246
4			0001	c0	1080	00	A 0033
5			0001	a7	e000	00	C 6300
6			0000	20	3880	00	B 0149
7			0001	c0	0e80	00	A 0029
8			0001	40	2d00	00	U 0126
9			01ff	ff	e000	00	,0www ww00
a			0000	00	0000	00	,0000 0000
b			0e00	00	0000	00	,7000 0000
c			0001	40	1f00	00	U 0062
d			0000	60	1f80	00	R 0063
e			0001	40	1900	00	U 0050
f			0001	40	2d00	00	U 0126
1550			0001	60	3100	00	T 0134
1			0001	e0	5e00	00	S 0260
2			0001	40	4280	00	U 0205
3			0000	40	2280	00	Y 0105
4			0000	40	3880	00	Y 0149
5			0001	e0	3580	00	S 0143
6			0001	60	2c00	00	T 0124
7			0001	40	0000	00	U 0000
8			0001	a0	4180	00	C 0203
9			0001	40	3780	00	U 0147
a			0000	40	2280	00	Y 0105
b			0000	40	3880	00	Y 0149
c			0001	40	0000	00	U 0000
d			0000	78	7800	00	,0003 1300
e			1a00	00	0000	00	,K000 0000
f			1400	00	0000	00	,f000 0000

CODING SHEET
 PERC-981-36-958 (REV. 2-58) DATE 4/18/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 44

STOR- AGE LOC- ATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1560			0001	c0	2580	00	A 0111
1			0001	60	1480	00	T 0041
2			0001	02	0300	00	P 1606
3			0000	00	0000	00	Z 0000
4			0001	04	a080	00	P 3701
5			0000	00	0200	00	Z 0004
6			0001	a0	2500	00	C 0110
7			0001	40	4680	00	U 0213
8			0001	a0	4c80	00	C 0225
9			1001	60	0000	00	,8000 0000
a			0001	40	5a00	00	U 0252
b			0000	00	0000	00	Z 0000
c			0200	00	0000	00	,1000 0000
d			0012	00	0000	00	,0090 0000
e			0001	a0	3580	00	C 0143
f			0001	00	0d00	00	P 0026
1570			0000	80	0000	00	T 0000
1			0001	a7	e000	00	C 6300
2			0000	20	0e80	00	B 0029
3			0001	c0	2280	00	A 0105
4			0001	40	2980	00	U 0119
5			0000	60	1f80	00	R 0063
6			0001	40	1900	00	U 0050
7			0000	40	3c80	00	Y 0157
8			0000	04	0000	00	Z 3200
9			0001	40	0000	00	U 0000
a			0001	20	2480	00	E 0109
b			0000	e0	2f80	00	M 0131
c			0001	c0	1380	00	A 0039
d			0001	40	1f80	00	U 0063
e			0000	60	1f80	00	R 0063
f			0001	40	1900	00	U 0050

NAVAL RESEARCH LABORATORY

71

UNCLASSIFIED

CODING SHEET
PRNC-REL-34-658 (REV. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TAPL AC. 4321
PROGRAMMER B. and E. WALD SHEET NO. 45

STOP- AGE LOCA- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1580			0001	a0	4180	00	C0203
1			0001	00	1600	00	P0044
2			0000	80	0000	00	I0000
3			0000	00	0000	00	Z0000
4			0001	40	0000	00	U0000
5			0001	60	0000	00	T0000
6			0001	40	3100	00	U0134
7			0001	ff	ff80	00	,000w wwwj
8			0000	00	0000	00	Z0000
9			0000	60	1f80	00	R0063
a			0001	40	1900	00	U0050
b			0000	40	1080	00	Y0033
c			0001	40	0000	00	U0000
d			0001	01	Bc00	00	,0008 OKQ0
e			0800	00	0000	00	,4000 0000
f			0001	01	Bd00	00	,0008 OKQ8
1590			0000	40	1f80	00	Y0063
1			1000	00	0000	00	,8000 0000
2			0001	04	7400	00	,0008 23f0
3			0800	00	0000	00	,4000 0000
4			0001	01	Bc00	00	,0008 OKQ0
5			0001	40	2580	00	U0111
6			0001	a0	5d00	00	C0258
7			0001	00	0100	00	P0002
8			0000	80	0000	00	I0000
9			0000	00	0000	00	Z0000
a			0000	20	0000	00	B0000
b			0001	60	5c00	00	T0256
c			0001	e0	4700	00	S0214
d			0001	60	5e80	00	T0261
e			0001	e0	5d80	00	S0259
f			0001	60	5e80	00	T0261

CODING SHEET
PRNC-REL-34-658 (REV. 2-58)

DATE 4/18/62 PROBLEM NO. _____ TAPL AC. 4321
PROGRAMMER B. and E. WALD SHEET NO. 46

STOP- AGE LOCA- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
15a0			0001	40	5080	00	U0233
1			0001	c0	1f00	00	A0262
2			0001	c0	5f80	00	A0263
3			0001	e0	4700	00	S0214
4			0001	60	5400	00	T0240
5			0001	e0	5d80	00	S0259
6			0001	60	5400	00	T0240
7			0001	40	5480	00	U0241
8			0001	c0	4980	00	A0219
9			0001	a0	5f80	00	C0263
a			0001	e0	4380	00	S0207
b			0001	c0	5d00	00	A0258
c			0001	60	3400	00	T0140
d			0001	40	1f00	00	U0054
e			1001	e7	ff80	00	,800w 3wwwj
f			0000	00	0000	00	Z0000
15b0			0000	60	1f80	00	R0063
1			0001	40	1900	00	U0050
2			0001	c0	2280	00	A0105
3			0001	40	3700	00	U0146
4			0001	00	00f8	00	P0063
5			0000	80	0000	00	I0000
6			0001	e0	5f80	00	S0263
7			0001	40	2800	00	U0116
8			0001	c0	4700	00	A0214
9			0001	40	4e80	00	U0229
a			0000	00	0000	00	Z0000
b			07ff	ff	ffc0	00	,3www wwwj
c			0000	00	0040	00	,0000 0002
d			0001	c0	4980	00	A0219
e			0001	40	5100	00	U0234
f			0000	00	0000	00	Z0000

NAVAL RESEARCH LABORATORY

CODING SHEET
PERC-REL-36-638 (REV. 2-54)

DATE 4/18/62 PROGRAM EN NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 47

STOP AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
15c0	"n"U	PCU	15d8	24	0000	90	PRINTS N'S, M'S
	PCU	PCU	0000	90	0000	90	LOCATIONS
2	PCU	PCU	0000	90	0000	90	0000-6343 WITH
3	"e"U	PCU	15d9	24	0000	90	NO CR'S OR
4	"TAB"U	PCU	15da	24	0000	90	LINE NUMBERS
5	XA	LAL 15cb	0000	80	15cb	20	
15c8→6	RCC	IL 15cb	0000	33	15cb	22	(15c5)
7	-RAd	CL 15c9	15ce	55	15c9	12	
8	LD 15cb		15cb	10			
15c7→9	"e"U	PCU	15d9	24	0000	90	
a	"m"U	PCU	15dB	24	0000	90	
b	PCU	PCU	0000	90	0000	90	
c	PCU	PCU	0000	90	0000	90	
d	STOP		0008	82			
e			1000	33	15cb	22	(15c7)
f			1000	50	15dc	26	(15d5)
15d0	XA	LAL 15d1	0000	80	15d1	20	RELOADING OF DUMP
1	RD()	XA	0000	32	0000	80	(15d0) EXTRACTS ALL BUT
2	LAL 15d3	LAL 15d4	15d3	20	15d4	20	NEGATIVE SIGN BITS
15d6→3	()A	(R-A)U	0000	50	15dc	26	(15d2)
4	U()	IL 15d3	0000	43	15d3	22	(15d2)(15d6)
5	-RAd	CL 15d7	15cf	55	15d7	12	
6	IL 15d4	LO 15d3	15d4	22	15d3	10	
15d5→7	STOP	LO 1248	0009	82	1248	10	
8	"N"		0000	00	0000	06	(15c0)
9	"e"		0000	00	0000	25	(15c3)(15c9)
a	"TAB"		0000	00	0000	29	(15c4)
b	"M"		0000	00	0000	07	(15ca)
c			ffff	ff	ffff	ff	(15d3)
d							
e							
f							

CODING SHEET
PERC-REL-36-638 (REV. 2-54)

DATE 4/18/62 PROGRAM EN NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 48

STOP AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1165 ↳ 15e0	"TAB" A	CUL 15e2	127f	50	15e2	14	
	TEMP A	RO 1167	104B	50	1167	11	
15e0→2	UA	RO 1165	0000	40	1165	11	
3							
4							
5			0000	83	2000	ff	0000-offff →
6			0ff0	83	2ff0	01	2000-2fff
7			007a	82	1000	11	
8			2000	83	0000	ff	2000-2fff →
9			2ff0	83	0ff0	01	0000-offff
a			007B	82	1000	11	
b			0000	83	3000	ff	0000-offff →
c			0ff0	83	3ff0	01	3000-3fff
d			007c	82	1000	11	
e			3000	83	0000	ff	3000-3fff →
f			3ff0	83	0ff0	01	0000-offff
15f0			007d	82	1000	11	
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							

CODING SHEET
PASC-REL-28-48M (REV. 2-54)

DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 49

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1600	1000A	(R-A)U	1000	50	1645	26	
1	X A	CUL 1604	0000	80	1604	14	
2	R A	A SWITCH	1672	50	1675	42	
1648 → 3	STOP	RO 1000	0006	82	1000	11	
1602 → 4	U w ₁	-w ₁ A	1637	43	1637	51	w ₁ = CTR 1
5	A w ₁	X U	1637	42	0000	71	
6	X A	A 1840	0000	80	1840	42	
7	T(1840, 1841, 3F)	1840	81	1841	3F	(1603)	
8	R A	LAL TABLE	1607	50	1620	20	
9	R A	A SWITCH	1672	50	1675	42	
a	LO 160c		160c	10			
b							
162a → c	R A	LAL INIT. TAB	1646	50	160f	20	
160a → d	6000A	A w ₂	6000	50	1638	42	w ₂ = SUBROUTINE CONTROL WORD
e	(R-A)U	U w ₂	163d	26	1639	43	w ₂ = SUBROUTINE LABEL (1600)(1675) SEE NOTE #20
f	(TAB. 4) A	A w ₄	0000	50	163a	42	w ₄ = W ₄ FROM TABLE 1
162e → 1610	(R-A)U	w ₃ A	163d	26	1639	50	
1636 → 1	CUL 1630	w ₂ A	1630	14	1638	50	
2	(R-A)U	U w ₅	163e	26	163B	43	
3	IL 160f	LAR 1621	160f	22	1621	21	w ₅ = RAW FIRST LOCATION OF SUBROUTINE
4	IL 160f	LAL 161f	160f	22	161f	20	
5	w ₅ A	(R-A)U	163B	50	163f	26	
6	U A	A w ₆	0000	40	163c	42	w ₆ = RAW TRACK
7	(R-A)U	U A	1640	26	0000	40	
8	M/-604	w ₆ Ad	1647	60	163c	54	
9	AR 2	A w ₆	0002	31	163c	42	w ₆ = BINARY TRACK @23
a	w ₅ A	(R-A)U	163B	50	1641	26	
b	U A	A w ₅	0000	40	163B	42	
c	(R-A)U	U A	1642	26	0000	40	
d	M/-604	w ₆ Ad	1647	60	163B	54	
e	w ₆ Ad	LO 1648	163c	54	1648	10	BINARY FIRST PHYSICAL LOCATION OF SUB.
1649 → f	(R-A)U	LO 1620	0000	54	1620	10	(1614)

CODING SHEET
PASC-REL-28-48M (REV. 2-54)

DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 50

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
161f → 1620	A (TABLE STOP)	IL 1620	0000	42	1620	22	(1608)(1622)(164B)
1	LAL 1622	(a+1) A	1622	20	0000	50	(1612)
2	A ()	IL 1620	0000	42	1620	22	(1621)
1653 → 3	LO 1652	LAL 1625	1652	10	1625	20	
162f → 4	IL 160f	RO 164f	160f	22	164f	11	
1651 → 5	() A	w ₆ Ad	0000	50	163c	54	(1623)
6	A ()	X U	0000	42	0000	71	(1650)
7	IL 160f	LAL 1628	160f	22	1628	20	
8	() A	R U	0000	50	1643	24	(1627)(164c)
1633 → 9	CUL 162c	IR w ₁	162c	14	1637	23	
a	CI 164B	LO 160c	164B	12	160c	10	
b							
1629 → c	A w ₄	(R-A)U	163a	42	1644	26	
d	R A	CUL 162f	1644	50	162f	14	
e	w ₄ A	LO 1610	163a	50	1610	10	
167d → f	160f A	RO 1623	160f	50	1623	11	
1611 → 1630	IL 160f	LAL 1631	160f	22	1631	20	
1635 → 1	() A	A w ₄	0000	50	163a	42	(1630)
2	R U	CUL 1634	1643	24	1634	14	
3	RO 1629		1629	11			
1632 → 4	(R-A)U	R A	1644	26	1644	50	
5	CUL 1630	w ₄ A	1630	14	163a	50	
6	LO 1610		1610	10			
7	w ₁		0000	00	0000	00	(160a)(1660)
8	w ₂		0000	00	0000	00	(1603)(1605)(1629)
9	w ₃		0000	00	0000	00	(1608)(1611)
a	w ₄		0000	00	0000	00	(1602)(1610)
b	w ₅		0000	00	0000	00	(1635)(1604)
c	w ₆		0000	00	0000	00	(1612)(1625)(1631)
d			0000	00	0000	00	(1618)(1613)
e			0000	00	0000	00	(1617)(1623)(1616)
f			0000	00	0000	00	(1619)(1625)
			ffff	00	0000	00	(1616)(1619)(1612)
			0000	ff	ff00	00	(1612)
			0000	ff	0000	00	(1615)

NAVAL RESEARCH LABORATORY

CODING SHEET
 PREC-REL-34-48 (REV. 7-54)
 DATE 4/20/62 PROGRAM NO. _____ TAPE NO. 4321
 PROGRAMMER B and E. WALD SHEET NO. 51

STOP-AGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1640			0000	f0	0000	00	(1617)
1			0000	00	ff00	00	(161a)
2			0000	00	f000	00	(161c)
3			ffff	ff	ffff	ff	(1628)(1632)(1640)(1647)
4			0000	00	0000	0f	(162c)(162d)(1634)
5			0000	00	00ff	00	(1600)
6			1880	00	0000	00	(160c)
7	-6@4		fa00	00	0000	00	(1618)(161d)
164e → 8	AR L	AWL	0001	31	163c	42	
9	LO 161f		161f	10			
a							
162a → b	1620 A	LAR 164c	1620	50	164c	21	
c	*A	A()	1643	50	0000	42	
d	LO 1603		1603	10			
e							
1624 → f		LAL 1650			1650	20	
1650 () A	LAL 1626		0000	50	1626	20	(164f)
1	LO 1625		1625	10			
1623 → 2	*A	A TURN ON	1673	50	1675	42	
3	IL 160f	RO 1623	160f	22	1623	11	
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f		eA			100d	50	

CODING SHEET
 PREC-REL-34-48 (REV. 7-54)
 DATE 4/20/62 PROGRAM NO. _____ TAPE NO. 4321
 PROGRAMMER B and E. WALD SHEET NO. 52

STOP-AGE LOCATION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1660 (*A)U	U	U	166f	26	1637	43	ENTER HERE TO USE FROM ALTERNATE D. S.I.B.
1	*A	A TABLE 2	1670	50	1662	42	
166a → 2	() A	CUL 166b	0000	00	0000	00	(1661)(1663)(1666)
3	IL 1662	LAL 1664	1662	22	1664	20	
4	() A	LAL 1665	0000	50	1665	20	(1663)
5	LO()		0000	10			(1664)
1662 → 6	IL 1662	IL 1662	1662	22	1662	22	
7	LAL 1668	*UL	1668	20	1643	24	
8	() A	CUL 166a	0000	50	166a	14	(1667)
9	LO NORMAL		1676	10			
1668 → a	U, U	RO 1662	1637	24	1662	11	
b	(R ORDER	SWITCH)					
c	*A d	RO 1040	1028	54	1040	11	
d							
e							
f			0001	ff	ff80	00	(1742)(1782)(17c2)(1822) (1640)(16a2)(1622)(1702)
1670			1840	50	1666	14	(1661)
1		U	0000	00	0000	00	(1675)(1676)
2			1671	42	1676	10	(1602)(1609) SWITCH OFF
3			1671	42	165f	11	(1652) SWITCH ON
4							
5	A w	()	1671	42	0000	00	(1672)(1609) ON RO 165f (1602)(1652) OFF LO 1676
1669 → 6	w A	*A d	1671	50	1028	54	
7	RO 1040		1040	11			
8							
9							
a							
b							
c							
d							
e							
f							

1666
 1663
 1703
 1748
 1783
 17c6
 1823

NAVAL RESEARCH LABORATORY

75

UNCLASSIFIED

CODING SHEET
 PRIC-REL-38-536 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 53

TITLE _____

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1680							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							
1690							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							

CODING SHEET
 PRIC-REL-38-536 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 54

TITLE _____

STOR- AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
16a0	1001 A	LAL 16a2	1001	50	16a2	20	SIN COS SUBROUTINE 14.0
1	IL 16a2	LO 16a2	16a2	22	16a2	10	
2	() A	(*)A)U	0000	50	16bf	26	(16a0)(16a1)
3	X ₁ A	IL 1001	16a8	50	16a5	15	
4	IL 1001	αA	1001	22	100c	50	
5	LO SIN	X ₂ A	16ac	10	16a9	50	
6	CUL NORMAL	IL 1001	1676	14	1001	22	
7	αA	LO cos	100c	50	16ad	10	
8	()	X ₁)	0000	00	0000	00	(16a3)
9	()	X ₂)	0000	00	0000	00	(16a5)
a							
b	A _{W1}	LO 16af	16d5	42	16af	10	
c	CL 16c2	R ₂ AD	16c2	12	16c8	54	
d	CL 16c3	R ₂ AD	16c3	12	16c9	54	
e	CL 16aB	RO 16aB	16aB	12	16ad	11	
f	R ₂ AD	CL 16c5	16ca	55	16c5	12	
16B0	R ₂ A	W ₂ AD	16cB	50	16d5	55	
16c5	D/R ₂	R ₂ AD	16cc	70	16cd	54	
1	(R ₂)AU	UA	16ce	26	0000	40	
2	A _{W1}	M/W ₂	16d6	42	16d6	60	
3	(R ₂)AU	UA	16ce	26	0000	40	
4	A _{W1}	AL	16d7	42	0001	30	
5	A _{W1}	M/R ₂	16d8	42	16cf	60	
6	(R ₂)AU	UA	16ce	26	0000	40	
7	R ₂ AD	M/W ₂	16d0	54	16d7	60	
8	(R ₂)AU	UA	16ce	26	0000	40	
a	R ₂ AD	M/W ₂	16d1	54	16d8	60	
b	(R ₂)AU	UA	16ce	26	0000	40	
c	R ₂ AD	M/W ₂	16d2	54	16d8	60	
d	(R ₂)AU	UA	16ce	26	0000	40	
e	R ₂ AD	M/W ₂	16d3	54	16d6	60	
f	(R ₂)AU	UA	16ce	26	0000	40	

NAVAL RESEARCH LABORATORY

CODING SHEET PERC-NR-38-486 (Rev. 2-58) DATE 4/20/62 PROBLEM NO. 4321 PROGRAMMER B. and E. WALD SHEET NO. 55

STORE LOCALIZATION	ORDER PAIR		CODED ORDER PAIR			REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		
16c0	47 AL J	48 W2 AR	0001	30	16d6	54	
1	49 Ad	RD 1000	100c	42	1000	11	
16ac → 2	5 Ad	LO 16ad	16cB	55	16ad	10	
16ad → 3	5 Ad	CL 16aB	16c9	55	16aB	12	
4	57 Ad	LO 16aB	16c9	54	16aB	10	
16af → 5	5 Ad	LO 16B1	16d4	55	16B1	10	
6							
7							
8	k0	270@9	0870	00	0000	00	4380 0000 (16ac)
9	k1	360@9	0840	00	0000	00	5F00 0000 (16c3)
a	k2	180@9+1	0520	00	0040	00	2K00 0002 (16af)
b	k3	90@9	02d0	00	0000	00	(16af)(16B)(16B0) 1680 0000 (16c2)
c	k4	90@8	0520	00	0000	00	2K00 0000 (16B)
d	k5	1@31	0000	00	0020	00	(16B)(16B)(16B)(16B)(16B) 0000 0001
e	k6	30 BITS	1fff	ff	ffc0	00	SEE NOTE #21
f	k7	C9@-4	0009	ed	7B40	00	004W 6GKF (16B)
16d0	k8	C7@-3	1f66	d9	9cc0	00	WG36 JJA6 (16B)
1	k9	C5@-1	028c	d1	5ec0	00	1466 8FW6 (16B)
2	k0	C3@0	15aa	21	f4c0	00	fKSI 0Wf6 (16B)
3	k1	C1@1	0490	fd	a900	00	2487 GK48 (16B)
4	k2	90@9-1	02cf	ff	ffc0	00	167W WWWQ (16c5)
5	w1		0000	00	0000	00	SEE NOTE #22
6	w2		0000	00	0000	00	SEE NOTE #23
7	w3		0000	00	0000	00	SEE NOTE #24
8	w4		0000	00	0000	00	SEE NOTE #25
9	k3	1@30	0000	00	0040	00	0000 0002 (17L)
a	k4		0000	3f	ffc0	00	0001 WWWQ (16ad)
b	k5		0000	3f	c000	00	0001 WWWQ (16ad)
c	k6		0001	ff	ffc0	00	000W WWWQ (16af)
d	k7		000f	c0	0000	00	00WQ 0000 (16f6)
e	k8		016a	09	e500	00	0G50 4W28 (16f7)
f	k9		000f	ff	ffc0	00	00WQ WWWQ (16f8)

CODING SHEET PERC-NR-38-486 (Rev. 2-58) DATE 4/20/62 PROBLEM NO. 4321 PROGRAMMER B. and E. WALD SHEET NO. 56

STORE LOCALIZATION	ORDER PAIR		CODED ORDER PAIR			REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		
16e0	1001A	1A1 16e2	1001	50	16e2	20	SQUARE ROOT SUBROUTINE 15.1
1	11 16e2	LO 16e2	16e2	22	16e2	10	
16e1 → 2	() A	(A) 11	0000	50	16e2	26	(16e2)(16e1)
3	X2 A	CUL NORMAL	16e6	50	1676	14	
4	11 1001	dA	1001	22	100c	50	
5	LO SQ. RT.		16e8	10			
6	() X3		0000	00	0000	00	(16e3)
7							
16e5	8	A w1	CL 16ea	16d5	42	16ea	12
59 RT → 9	24	STOP	0000	82			
16e8 → a	25	Ad	CL 16ed	16d9	55	16ed	12
16f9 → b	25	XA	A 100c	0000	80	100c	42
c	20	RO 1000		1000	11		
16ea → d	4	Ad	CL 16f6	16da	55	16f6	12
e	31	Ad	CL 16f8	16dB	54	16f8	12
16f7	f	10A	A w2	16dc	50	16db	42
16f8	19	10A	D/W2	16d5	51	16db	70
16f5	16fa	16f0					
1	Ad	(A) 11	16cd	54	16ce	26	
2	1A	w1 Ad	0000	40	16d6	54	
3	16 AR 1	A w3	0001	31	16d7	42	
4	-w3 A	CL 16f9	16d7	51	16f9	12	
5	w3 Ad	RO 16ef	16db	54	16ef	11	
16ed → 6	6	Ad	CL 16fa	16dd	55	16fa	12
7	16 Ad	RO 16ef	16de	50	16ef	11	
16ee → 8	52	13 A	RO 16ef	16df	50	16ef	11
16f4 → 9	44	w1 A	RO 16eB	16db	50	16eB	11
16f6 → a	9	14 A	RO 16ef	16fc	50	16ef	11
b							
c	k14		0fff	ff	ffc0	00	7WWW WWWQ (16fa)
d	k15		1721	00	0000	00	(1709)
e	k16	1@9	0008	00	0000	00	(170a)(1727)
f	k17	1@8	0010	00	0000	00	(170B)

NAVAL RESEARCH LABORATORY

77

UNCLASSIFIED

CODING SHEET
 PRC-REL-38-638 (Rev. 2-54) DATE 6/20/62 PROBLEM NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 57

STEP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR		REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.		
1700	1001 A	LAL 1702	1001	50	1702 20	ARC TAN SUBROUTINE 16.0
1	1L 1702	LO 1702	1702	22	1702 10	
1701 → 2	() A	(A) U	0000	50	166f 26	(1100)(1701)
3	X y A	CUL NORM	1706	50	1676 14	
4	1L 1001	α A	1001	22	100c 50	
5	LO ARC TAN		1708	10		
6	(X y)		0000	00	0000 00	(1703)
1705 ARC TAN → 7						
8	A w ₁	CL 1724	16d5	42	1724 12	
9	A w ₂	AL 1721	16fd	50	1721 20	
10	A w ₃	w ₁ Ad	16fe	50	16d5 55	
1708 → 11	A w ₄	A w ₁ Ad	16db	42	16ff 55	
12	D/w ₁	A w ₂ Ad	16db	70	16cd 54	
13	(A) U	U A	16ce	26	0000 40	
14	A w ₃	M/w ₁	16d7	42	16d7 60	
15	A w ₄	M/w ₂	16d8	42	172b 60	
1710	(A) U	U A	16ce	26	0000 40	
1	A w ₁ Ad	M/w ₁	172c	54	16d8 6c	
2	(A) U	U A	16ce	26	0000 40	
3	A w ₂ Ad	M/w ₁	172d	54	16d8 60	
4	(A) U	U A	16ce	26	0000 40	
5	A w ₃ Ad	M/w ₁	172e	54	16d8 60	
6	(A) U	U A	16ce	26	0000 40	
7	A w ₄ Ad	M/w ₁	172f	54	16d8 60	
8	(A) U	U A	16ce	26	0000 40	
9	A w ₁ Ad	M/w ₁	1730	54	16d8 60	
10	(A) U	U A	16ce	26	0000 40	
11	A w ₂ Ad	M/w ₁	1731	54	16d8 60	
12	(A) U	U A	16ce	26	0000 40	
13	A w ₃ Ad	M/w ₁	1732	54	16d7 60	
14	(A) U	U A	16ce	26	0000 40	
15	M/w ₁	(A) U	1733	60	16ce 26	

CODING SHEET
 PRC-REL-38-638 (Rev. 2-54) DATE 6/20/62 PROBLEM NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 58

STEP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR		REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.		
1720	U A	A w ₁ Ad	0000	40	1734 54	
1	RO () A	A w ₁	0000	11	16d5 42	(1709)(1726)
1725 → 2	w ₁ A	A 100c	16d5	51	100c 42	
3	RO 1000		1000	11		
1708 → 4	A w ₂ Ad	CL 1726	16d9	55	1726 12	
5	X A	RO 1722	0000	80	1722 11	
1724 → 6	A w ₃ A	LAL 1721	1735	50	1721 20	
7	A w ₄ A	w ₁ Ad	16fe	50	16d5 54	
8	LO 1702		1703	10		
9						
10						
11	A w ₁	C15@1	1ff7	B2	8180 00	WWRK 940J (170f)
12	A w ₂	C13@1	002c	C5	9480 00	0166 2JF4 (1711)
13	A w ₃	C11@1	1f8d	7f	1200 00	WJ6G W8W0 (1713)
14	A w ₄	C9@1	00c5	77	df40 00	062G GQWf (1715)
15	A w ₅	C7@1	1ee3	27	3a40 00	W719 39K2 (1717)
1730	A w ₆	C5@1	0198	81	4B40 00	DJJ4 OFSF (1719)
1	A w ₇	C3@1	1d55	67	9080 00	affG 3J84 (1718)
2	A w ₈	C1@1	07ff	ff	a680 00	3WwW WK34 (171A)
3	A w ₉	572-e8	0394	B8	8300 00	Nf5 KJ18 (171f)
4	A w ₁₀	45-e9	0168	00	0000 00	PG40 0000 (1720)
5	A w ₁₁		1722	00	0000 00	(1726)
6	A w ₁₂	1a9-e0	04d1	04	d440 00	2688 2682 (1750)
7	A w ₁₃	1@1	0800	00	0000 00	4000 0000 (1752)
8	A w ₁₄	1@-2	0666	66	6680 00	3333 3334 (1754)
9	A w ₁₅	a7@-4	003d	1f	25c0 00	0198 W929 (1759)
10	A w ₁₆	a6@-3	0053	B8	3480 00	029K J1F4 (175B)
11	A w ₁₇	a5@-2	011d	6d	7900 00	08AG 6G38 (175d)
12	A w ₁₈	a4@-1	0255	9e	e180 00	12FJ W70J (175f)
13	A w ₁₉	a3@0	0411	fe	ff40 00	208W W7Wf (1761)
14	A w ₂₀	a2@1	054d	45	d980 00	2f6f 2QJJ (1763)
15	A w ₂₁	a1@2	049a	ec	7e40 00	24K7 63W2 (1765)

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRNC-WL-26-498 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPI NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 59

STORE AGE LOCATION	ORDER PAIR		CODED ORDER PAIR			REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ORDER NO.	
1740	1001 A	LAL 1742	1001	50	1742	20 EXPONENTIAL 17.0
1	IL 1742	X ₆ U	1742	22	1742	10
1741 → 2	() A	(*) A) U	0000	50	166f	26 (1740)(1741)
3	X ₆ A	CUR 1745	174B	50	1745	15
4	IL 1001	α A	1001	22	100c	50
1743 → 5	LO 2 ^c	X ₆ A	1750	10	174c	50
6	CUL 1748	IL 1001	1748	14	1001	22
7	α A	LO e ^c	100c	50	176d	10
1746 → 8	X ₇ A	CUL ^{NOGRAM} R	174d	50	1676	14
9	IL 1001	α A	1001	22	100c	50
a	RO 10 ^c		1751	11		
b	(X ₆)		0000	00	0000	00 (1743)
c	(X ₆)		0000	00	0000	00 (1745)
d	(X ₇)		0000	00	0000	00 (1748)
1751 → e	IR 1769	16d5 A	1769	23	16d5	50
f	RO 1759		1759	11		
1745 → 1750	M/k ₂₃	(*) A) U	1736	60	16ce	26
174a	U A	CL 176a	0000	40	176a	12
2 ²⁵	k ₂₄ Ad	AL	1737	54	0000	41
3	1753 A	LO → 33	1753	50	1758	10
4	M/k ₂₅	(*) A) U	1738	60	16ce	26
1757 → 5	U d	RO 1000	100c	43	1000	11
6	1756 A	LO → 33	1756	50	1758	10
176c → 7	AL 2	LO 1755	0002	30	1755	10
1753 → 8	U w	PAR EXIT	16d5	43	1769	21
174b → 9	LO 174c	M/k ₂₆	174e	10	1739	60
1744	(*) A) U	U A	16ce	26	0000	40
a	k ₂₇ Ad	M/w	173a	54	16d5	60
b	(*) A) U	U A	16ce	26	0000	40
c	k ₂₈ Ad	M/w	173B	54	16d5	60
d	(*) A) U	U A	16ce	26	0000	40
e	k ₂₉ Ad	M/w	173c	54	16d5	60
f	(*) A) U	U A	16ce	26	0000	40

CODING SHEET
 PRNC-WL-26-498 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPI NO. 4321
 PROGRAMMER B. and E. Wald SHEET NO. 60

STORE AGE LOCATION	ORDER PAIR		CODED ORDER PAIR			REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ORDER NO.	
1760	(*) A) U	U A	16ce	26	0000	40
1	k ₃₀ Ad	M/w	173d	54	16d5	60
2	(*) A) U	U A	16ce	26	0000	40
3	k ₃₁ Ad	M/w	173e	54	16d5	60
4	(*) A) U	U A	16ce	26	0000	40
5	k ₃₂ Ad	M/w	173f	54	16d5	60
6	(*) A) U	U A	16ce	26	0000	40
7	k ₃₃ Ad	A w ₂	1770	54	16d6	42
8	M/w ₂	(*) A) U	16d6	60	16ce	26
9	U A	LO (EXIT)	0000	40	0000	10 (174e)(175e)
1751 → a	AL	AL	0000	41	0000	41
b	176B A	LO → 33	176B	50	1758	10
1747 → c	LO 1757		1757	10		
d	M/k ₃₆	RO 1750	1771	60	1750	11
e						
f						
1770	k ₃₇	1e3	0200	00	0000	00 1000 0000 (1767)
1	k ₃₈	log 10 ^e	06f2	de	c540	00 3796 w62f (176d)
2	k ₃₉	1e5-1e30	007f	ff	ff c 00	00 03ww wwww (178a)
3	k ₄₀	1e5	0080	00	0000	00 0400 0000 (178f)
4	k ₄₁	5e5	0280	00	0000	00 1400 0000 (178b)
5	k ₄₂		fff	ff	f800	00 wwwww wwww (1777)
6	k ₄₃	1e1 + 1e30	0800	00	0040	00 4000 0002 (179a)
7	k ₄₄	1e2 @ 2	05a8	27	9980	00 2K41 3J4J (177b)
8	k ₄₅	C7e5	0037	95	d2c0	00 01GJ fQ96 (17a5)
9	k ₄₆	C5e5	0049	cd	8400	00 024q 6j20 (17a7)
a	k ₄₇	C3e5	007B	1c	4980	00 03K8 Q24J (17a9)
b	k ₄₈	C1e5	0171	54	7640	00 0G8f f3G2 (17aB)
c	k ₄₉	.5e5	0040	00	0000	00 0200 0000 (17a8)
d	k ₅₀	1e29	0000	00	0080	00 (17af)(17b7)(1789)
e	k ₅₁	1/10q.e@1	1a74	6f	4040	00 K3F3 7F02 (1788)
f	k ₅₂	1/10q.10e1	1d97	7d	9600	00 QJGG QJGO (1789)

NAVAL RESEARCH LABORATORY

79

UNCLASSIFIED

CODING SHEET
 FMRC-REL-36-438 (Rev. 2-58) DATE 4/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 61

STOP AGE LOCAL TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1780	1001 A	LAL 1782	1001	50	1782	20	LOG X SUBROUTINE 18.0
1	IL 1782	LO 1782	1782	22	1782	10	
1781 → 2	(JA	(A-A)U	0000	50	166f	26	(1780)(1781)
3	X ₂ A	CUL	1785	50	1676	14	
4	IL 1001	LO 1792	1001	22	1792	10	
5	(X ₂		0000	00	0000	00	(1783)
6	w ₂		0000	00	0000	00	(1783)
7	w ₂		0000	00	0000	00	(1783)(1787)
1792 log x → 8	⁰ - R ₂ Ad	CL 178a	1689	55	178a	12	(1783)(1787)(1789)(1790)
9	STOP		0000	82			ERROR STOP
1789 → a	² - R ₂ Ad	CL 178f	1772	55	178f	12	
b	¹¹ R ₂ Ad	AL 5	1773	54	0005	30	
c	¹³ AU	AU	0000	41	0000	41	
d	178d A	LO → 16	178d	50	1794	10	
e			0280	00	0000	00	505 1400 0000
178a → f	⁴ R ₂ Ad	(R ₂ A)U	1773	54	1775	26	
1790	⁰⁴ 1790 A	LO → 16	1790	50	1794	10	
1	CONSTANT		0000	00	0000	00	
1784 → 2	d A	LO 1788	100c	50	1788	10	
3			0000	00	0000	00	
178d → 4	U ₂	LAL EXIT	16d5	43	1784	20	
1790 → 5	IL EXIT	w ₂ A	1784	22	16d5	50	
1799 → 6	¹⁵³ R ₂ Ad	CL 179a	1737	55	179a	12	
7	¹⁵³ R ₂ Ad	UA	1775	26	0000	40	
8	¹⁵⁴ R ₂ Ad	w ₂ Ad	1776	54	16d5	54	
9	¹⁶ R ₂ Ad	LO 179b	16d5	42	179b	10	
179b → a	¹⁹ R ₂ Ad	A ₂	1776	54	16d6	42	
b	²¹ AR	²² R ₂ Ad	0001	31	1777	54	
c	²³ A ₂	IL 1001	16d7	42	1001	22	
d	LAL 179e	LO 179e	179e	20	179e	10	
179d → e	(JA	AL 24	0000	50	0018	30	(179d)
f	²⁴ A ₂	w ₂ A	16d8	42	16d6	50	

CODING SHEET
 FMRC-REL-36-438 (Rev. 2-58) DATE 4/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 62

STOP AGE LOCAL TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
17a.0	²⁸ AR L	- R ₂ Ad	0001	31	1777	55	
1	³⁰ D/w ₂	- R ₂ Ad	16d7	70	16cd	54	
2	(R ₂ A)U	UA	16ce	26	0000	40	
3	A ₂ w ₂	M/w ₂	1786	42	1786	60	
4	(R ₂ A)U	UA	16ce	26	0000	40	
5	³⁵ A ₂ w ₂	M/R ₂ Ad	1787	42	1778	60	
6	(R ₂ A)U	UA	16ce	26	0000	40	
7	R ₂ Ad	M/w ₂	1779	54	1787	60	
8	(R ₂ A)U	UA	16ce	26	0000	40	
9	⁴¹ R ₂ Ad	M/w ₂	177a	54	1787	60	
a	(R ₂ A)U	UA	16ce	26	0000	40	
b	R ₂ Ad	M/w ₂	177B	54	1786	60	
c	(R ₂ A)U	UA	16ce	26	0000	40	
d	⁴⁶ R ₂ Ad	A ₂ w ₂	177c	54	178f	42	
e	IL 1001	LAL 17af	1001	22	17af	20	
f	(JA	- R ₂ Ad	0000	50	177d	55	(179e)
17B.0	¹¹³ CL 17B7	¹¹⁴ R ₂ A	17B7	12	1737	51	
1	¹¹⁶ A ₂ w ₂	¹¹⁷ w ₂ A	178e	42	16d6	50	
2	¹²² AL 25	¹²³ R ₂ A)U	0019	30	178c	26	
3	UA	¹²⁴ w ₂ Ad	0000	40	16d8	55	
4	¹²⁶ (JA	¹²⁷ w ₂ Ad	0000	54	178f	55	(1794)(1795)
5	¹²⁹ M/w ₂	(R ₂ A)U	178e	60	16ce	26	
6	U ₂	¹³² RO 100c	100c	43	1000	11	
7	¹³³ R ₂ Ad	¹³⁴ CL 17B9	177d	55	1789	12	
8	¹⁴³ R ₂ A	LO 17B1	177e	50	17B1	10	
9	¹⁴⁵ R ₂ A	LO 17B1	177f	50	17B1	10	
a							
b							
c	DRDP SIGN		ffff	ff	ffff	00	7www wwww (1782)
d	w ₂		0000	00	0000	00	(1802)(180e)(1810)
e	w ₂		0000	00	0000	00	(1782)(1785)(1802)(1800)
f	w ₂		0000	00	0000	00	(1782)(1785)(1802)(1800)

NAVAL RESEARCH LABORATORY

CODING SHEET
FORM-NRL-36-438 (REV. 2-58)

DATE 6/20/62 PROGRAM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 63

STEP AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR			REMARKS		
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		ORDER NO.	
17c0	1001 A	LAL 17c2	1001	50	17c2	20	MARSHIN-ARCLOS SUBROUTINE 20.0	
	1	IL 17c2	LO 17c2	17c2	22	17c2	10	
17c1	2	() A	(A) U	0000	50	166f	26	(17c0)(17c1)
	3	X ₁ A	CUR 17c5	17c8	50	17c5	15	
	4	IL 1001	α A	1001	22	10pc	50	
17c3	5	LO ^{ARC} ₂₇₅	X ₂ A	17d0	10	17c9	50	
	6	CUL ^{ARC} _{NOEHAL}	IL 1001	1676	14	1001	22	
	7	α A	LO ^{ARC} ₂₀₅	100c	50	17cc	10	
	8	() X ₁	()	0000	00	0000	00	(17c3)
	9	() X ₂	()	0000	00	0000	00	(17c5)
17d6, 17d9	a							
17d4, 17c7	b	A ₁	²¹ RO 1000	100c	42	1000	11	
ARC	c	²¹ CR 17d6	^{21a} A ₁	17d6	13	16d5	42	
17c5	d	-w ₁ A	ALL	16d5	51	0000	41	
	e	17ce A	LO → 44	17ce	50	17da	10	
	f	¹³⁰ K ₂ AD	LO 17cB	16cB	54	17cB	10	
ARC	g	¹³⁰ CR 17d4	⁵ A ₁	17d4	13	16d5	42	
17c5	h	-w ₁ A	ALL	16d5	51	0000	41	
	i	17d2 A	LO → 44	17d2	50	17da	10	
	j	⁹ A ₁	-w ₁ A	16d5	42	16d5	51	
17d0	k	LO 17cB	ALL	17cB	10	0000	41	
	l	¹⁴ 17d5 A	LO → 44	17d5	50	17da	10	
17cc	m	LO 17cB	ALL	17cB	10	0000	41	
	n	¹⁴ 17d7 A	LO → 44	17d7	50	17da	10	
	o	A ₁	¹¹⁷ K ₂ A	16d5	42	16cB	50	
17d5	p	-w ₁ AD	LO 17cB	16d5	55	17cB	10	
17ce	q	¹⁴ 17d182a	⁵⁰ M/ ₁ ₁₄₀	182a	10	1811	60	
17d2	r	(K ₁) A U	U A	16ce	26	0000	40	
182c	s	¹⁴ K ₁ AD	⁵² M/ ₁ ₁₄₀	1812	54	16d6	60	
	t	(K ₁) A U	U A	16ce	26	0000	40	
	u	⁵³ K ₁ AD	⁵⁴ M/ ₁ ₁₄₀	1813	54	16d6	60	
	v	(K ₁) A U	U A	16ce	26	0000	40	

CODING SHEET
FORM-NRL-36-438 (REV. 2-58)

DATE 6/20/62 PROGRAM NO. _____ TAPE NO. 4321
PROGRAMMER B. and E. WALD SHEET NO. 64

STEP AGE LOCAL- TION	ORDER PAIR		CODED ORDER PAIR			REMARKS		
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		ORDER NO.	
17e0	⁵⁵ K ₁ AD	⁵⁶ M/ ₁ ₁₄₀	1814	54	16d6	60		
	1	(K ₁) A U	U A	16ce	26	0000	40	
	2	⁵⁷ K ₁ AD	⁵⁸ M/ ₁ ₁₄₀	1815	54	16d6	60	
	3	(K ₁) A U	U A	16ce	26	0000	40	
	4	⁵⁹ K ₁ AD	⁶⁰ M/ ₁ ₁₄₀	1816	54	16d6	60	
	5	(K ₁) A U	U A	16ce	26	0000	40	
	6	⁶¹ K ₁ AD	²⁰⁶ M/ ₁ ₁₄₀	1817	54	16d6	60	
	7	(K ₁) A U	U A	16ce	26	0000	40	
	8	⁶² K ₁ AD	¹⁰⁸ A ₁ ₃₂	1818	54	16d7	42	
	9	²⁰⁷ K ₁ AD	³² -w ₁ AD	1737	50	16d6	55	
	a	AR L	ALL	0001	31	0000	41	
	b	³⁹ 17eB A	LO → 100	17eB	50	17f0	10	
	c	⁵⁷ M/ ₁ ₁₄₀	(K ₁) A U	16d7	60	16ce	26	
	d	U A	⁴¹ M/ ₁ ₁₄₀	0000	40	1819	60	
17eB	e	(K ₁) A U	U A	16ce	26	0000	40	
1825	f	⁴³ K ₁ AD	⁴³ LO (EXIT)	16cB	54	0000	10	(182a)(182b)
SA	g	¹⁰⁰ U ₁	LAR EXIT	16d5	43	1810	21	
RT	h	IR EXIT	w ₁ A	1810	23	16d5	50	
	i	¹⁰² CL 17f3	¹⁶¹ STOP	17f3	12	0000	82	ERROR STOP
17f2	j	¹⁰¹ K ₁ AD	¹⁰² CL 17f5	177d	55	17f5	12	
	k	¹⁶² X A	¹⁶² RO 1810	0000	80	1810	11	
17f3	l	¹⁰³ K ₁ AD	¹⁰⁴ CL 17fd	181a	55	17fd	12	
	m	¹⁰⁴ K ₁ AD	¹⁰⁵ CL 17fa	181c	54	17fa	12	
	n	¹⁰⁵ K ₁ AD	¹⁰⁶ ALL	181d	54	0000	41	
	o	17f8 A	LO → 155	17f8	50	1804	10	
	p			0001	ff	ffc0	00	INITIAL GUESS #4 DOWN NUMBER
17f6	q	¹¹⁶ K ₁ AD	ALL	181d	54	0000	41	
	r	17fB A	LO → 155	17fB	50	1804	10	
	s			001f	ff	ffc0	00	INITIAL GUESS #3 DOWN NUMBER
17f5	t	¹⁰⁵ K ₁ AD	¹⁰⁴ CL 1801	181e	55	1801	12	
	u	¹⁰⁵ K ₁ AD	ALL	181f	54	0000	41	
	v	17ff A	LO → 155	17ff	50	1804	10	

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRNC-REL-36-436 (REV. 2-58)

DATE 4/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Ward SHEET NO. 65

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR			REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		
1800			00ff	ff	ffc0	00	INITIAL GUESS #2 ZUWUW WUWUW
17fd → 1	¹⁵⁵ K ₄ Ad	ALL	181f	54	0000	41	
2	1802 A	LO → 155	1802	50	1804	10	
17f8	3		0fff	ff	ffc0	00	INITIAL GUESS #1 ZUWUW WUWUW
17ff	4	¹⁵⁵ U ₁ W	LAL EXIT	16d5	43	1808	20
1802	5	IL EXIT	W ₁ A	1808	22	16d5	50
6	AR L	¹⁵⁸ A ₁ W ₅	0001	31	1787	42	
7	LO 1808		1808	10			
1807	8	¹⁵⁹ JA	¹⁵³ A ₁ W ₆	0000	50	17Bf	42 (1804)(1805)
180f →	9	AR L	A ₁ W ₇	0001	31	17Be	42
a	¹⁵⁷ W ₅ A	¹⁶⁰ D/W ₆	1787	50	17Bf	70	
b	¹⁵⁶ K ₂ Ad	(¹⁵⁶ A ₁ W ₁)	16cd	54	16ce	26	
c	¹⁵⁴ U ₁ A	¹⁵² W ₁ Ad	0000	40	17Be	54	
d	¹⁵² A ₁ W ₅	¹⁵¹ W ₁ Ad	17Bd	42	17Bf	55	
e	¹⁵¹ CL 1810	¹⁵² W ₅ A	1810	12	17Bd	50	
17f4	f	RO 1808	1808	11	0000	00	
180e → 1810	0	¹⁵⁹ W ₅ A	¹⁶⁰ LO(Exit)	17Bd	50	0000	10 (17f0)(17f1)
1	¹⁴⁰ K ₄₀	a7@-6	1E85	0B	a9c0	00	w5f8 5K4G (17da)
2	¹⁴¹ K ₄₁	a6@-5	036a	43	15c0	00	1Q57 18fQ (17dc)
3	¹⁴² K ₄₂	a5@-4	1Ba0	1c	d380	00	KK00 q69J (17de)
4	¹⁴³ K ₄₃	a4@-3	03f4	43	e140	00	1wf2 1wof (17df)
5	¹⁴⁴ K ₄₄	a3@-2	1cc9	f1	B6c0	00	q64w 8KG6 (17eg)
6	¹⁴⁵ K ₄₅	a2@-1	02d8	ea	7600	00	16J7 53G0 (17eh)
7	¹⁴⁶ K ₄₆	a1@0	1c91	00	d8c0	00	q488 06J6 (17ei)
8	¹⁴⁷ K ₄₇	a0@1	0c90	fd	a740	00	6487 qK3F (17ej)
9	¹⁴⁸ K ₄₈	-1@7	18d6	88	f980	00	J6G4 47J (17ek)
a	¹⁴⁹ K ₄₉	2@30	0000	1f	ff80	00	0000 wwwJ (17el)
b	¹⁴⁰ K ₄₀	1@15	0000	20	0000	00	0001 0000
c	¹⁴² K ₄₂	1@15-1@22	0000	1f	0000	00	0000 w900 (17fk)
d	¹⁴⁰ K ₄₀	1@22	0000	00	4000	00	0000 0200 (17fl)
e	¹⁴² K ₄₂	1@8-1@15	000f	e0	0000	00	007w 0000 (17fm)
f	¹⁴⁰ K ₄₀	1@8	0010	00	0000	00	0080 0000 (17fn)

CODING SHEET
 PRNC-REL-36-436 (REV. 2-58)

DATE 4/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. Ward SHEET NO. 66

STOP AGE LOC. TION	ORDER PAIR		CODED ORDER PAIR			REMARKS	
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS		
1820	1001 A	LAL 1822	1001	50	1822	20	SQUARE ROOT INTERNAL TO 20.0
1921 → 2	() A	(¹⁵⁶ A ₁ W ₁)	0000	50	166f	26	(1820)(1821)
3	X ₁ A	CUL ¹⁵⁶ A ₁ W ₁	1628	50	1676	14	
4	1L 1001	d 1L	1001	22	100c	24	
5	1825 A	LO 17f0	1825	50	17f0	10	
6	A α	RO 1000	100c	42	1000	11	
7							
8	() X ₁		0000	00	0000	00	(1823)
9							
17da → a	U ₁ W ₂	LAR 17ef	16db	43	17ef	21	
b	IR 17ef	W ₁ A	17ef	23	16db	50	
c	RO 17da		17da	11			
d							
e							
f							
1830							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							

NAVAL RESEARCH LABORATORY

CODING SHEET
 PRNC-871-34-436 (Rev. 2-54)

DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B and E. Ward SHEET NO. 67

TITLE TABLE 2

STORE AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1840							(1606)
1							(1607)
2							↓
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							
1850							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							

CODING SHEET
 PRNC-871-34-436 (Rev. 2-54)

DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B and E. Ward SHEET NO. 68

TITLE TABLE 2 (CON'T)

STORE AGE LOC- TION	ORDER PAIR		CODED ORDER PAIR				REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	ADDRESS	ORDER NO.	
1860							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							
1870							
1							
2							
3							
4							
5							
6							
7							
8							
9							
a							
b							
c							
d							
e							
f							

CODING SHEET
 PRC-NRL-34-496 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 69

TABLE 1

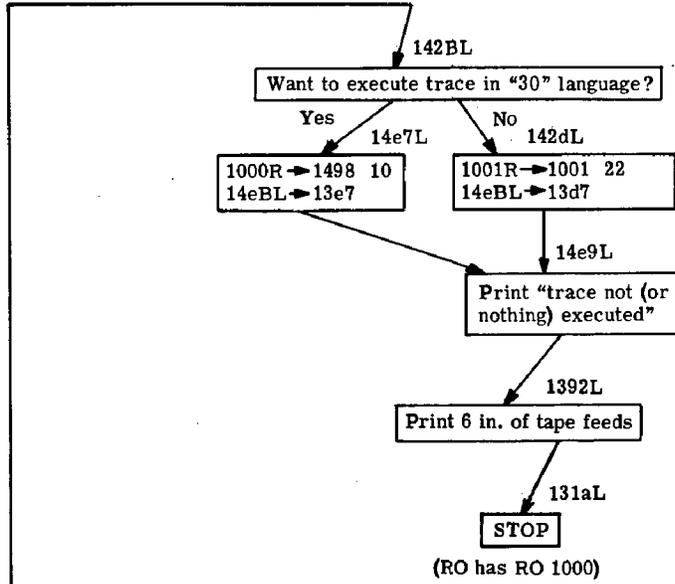
STOR- AGE LOCA- TION	ORDER PAIR		CODED ORDER PAIR		REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	
1880			0140	00 0000	OF SIN-COS 14.0
1			16a0	00 0000	00
2			0000	60 1880	00
3			0001	40 0000	00
4			16a8	00 0000	00 Y ₁
5			0001	40 0200	00
6			16a9	00 0000	00 Y ₂
7			0151	00 0000	OF SQ. ROOT 15.1
8			16e0	00 0000	00
9			0000	60 1900	00
a			0001	40 0000	00
b			16e6	00 0000	00 X ₃
c			0160	00 0000	OF ARC TAN 16.0
d			1700	00 0000	00
e			0000	60 1980	00
f			0001	40 0000	00
1890			1706	00 0000	00 Y ₄
1			0170	00 0000	OF EXPONENTIAL 17.0
2			1740	00 0000	00
3			0000	60 0480	00
4			0001	40 0000	00
5			174B	00 0000	00 Y ₅
6			0001	40 0100	00
7			174c	00 0000	00 Y ₆
8			0001	40 0180	00
9			174d	00 0000	00 Y ₇
a			0180	00 0000	OF LOG _e X 18.0
b			1780	00 0000	00
c			0000	60 0c00	00
d			0001	40 0000	00
e			1785	00 0000	00 Y ₈
f			0200	00 0000	OF ARCSIN, ARCCOS 20.0

CODING SHEET
 PRC-NRL-34-496 (Rev. 2-58) DATE 6/20/62 PROBLEM NO. _____ TAPE NO. 4321
 PROGRAMMER B. and E. WALD SHEET NO. 70

STOR- AGE LOCA- TION	ORDER PAIR		CODED ORDER PAIR		REMARKS
	LEFT ORDER	RIGHT ORDER	ADDRESS	ORDER NO.	
18a0			17c0	00 0000	00
1			0000	60 0a80	00
2			0001	40 0000	00
3			17c8	00 0000	00 Y ₉
4			0001	40 4580	00
5			17c9	00 0000	00 Y _a
6			0200	00 0000	OF SQ. RT. OF 20.0
7			1820	00 0000	00
8			0000	60 3900	00
9			0001	40 2000	00
a			1828	00 0000	00 Y _B
b			ffff	ff ffff	ff
c					
d					
e					
f					
18B0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
a					
b					
c					
d					
e					
f					

Appendix E

Flow Charts for the NAREC Simulation of LGP-30 Operand Orders



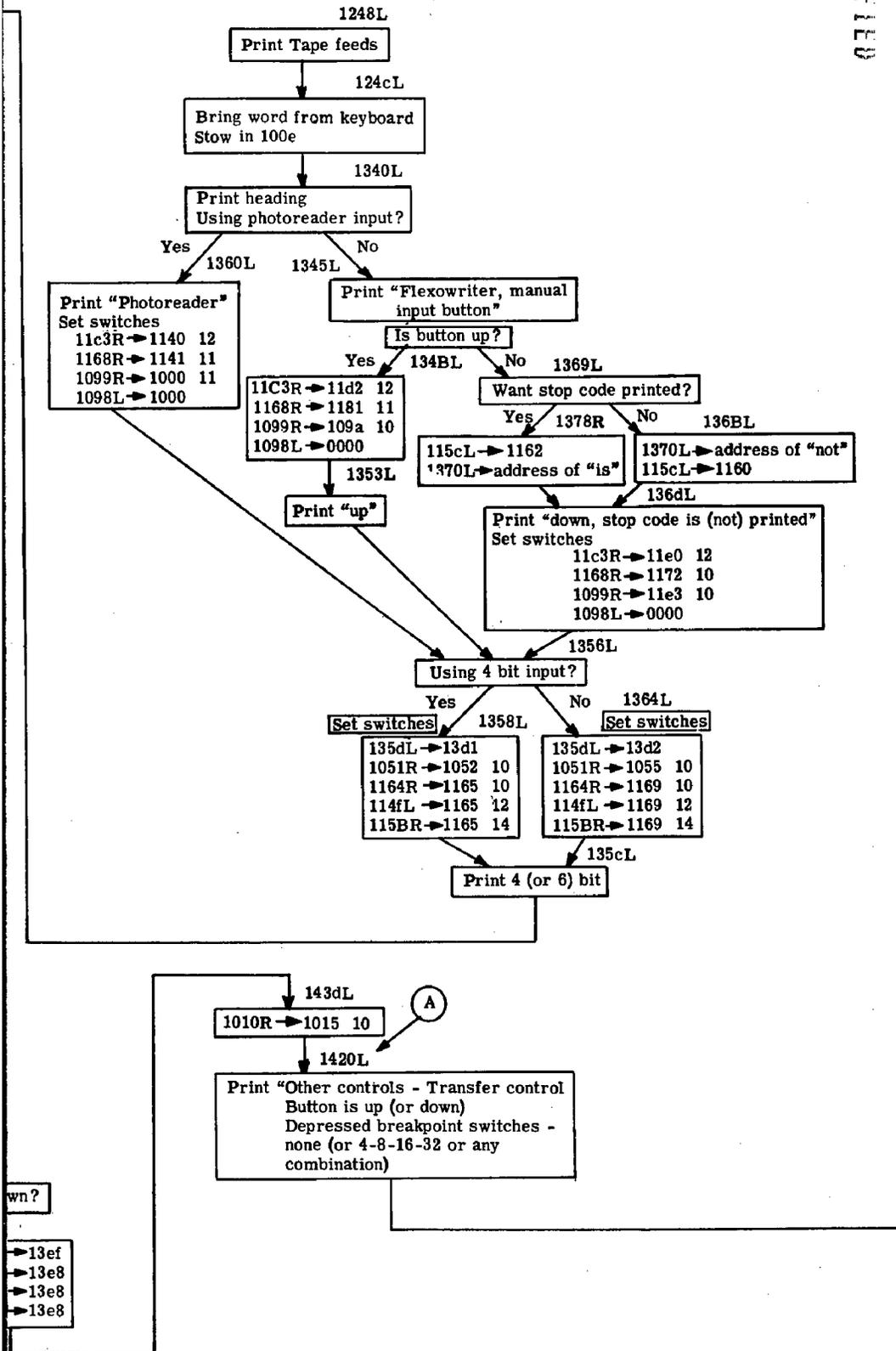
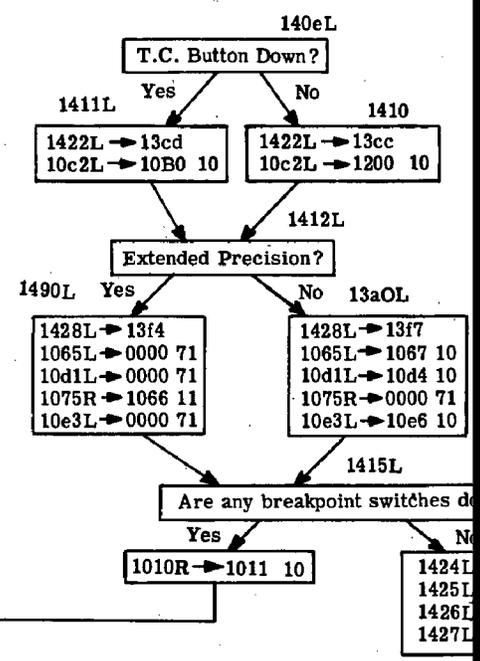
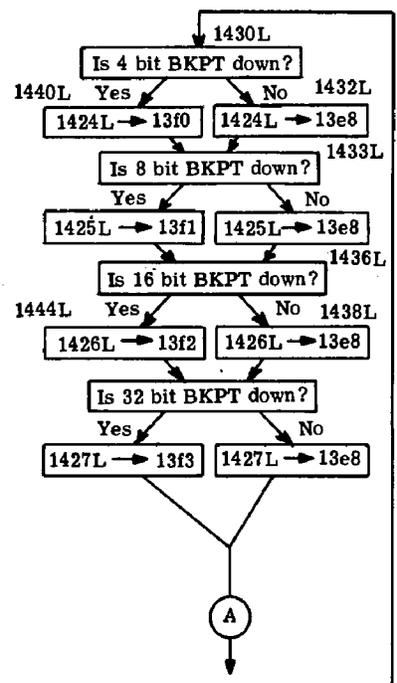
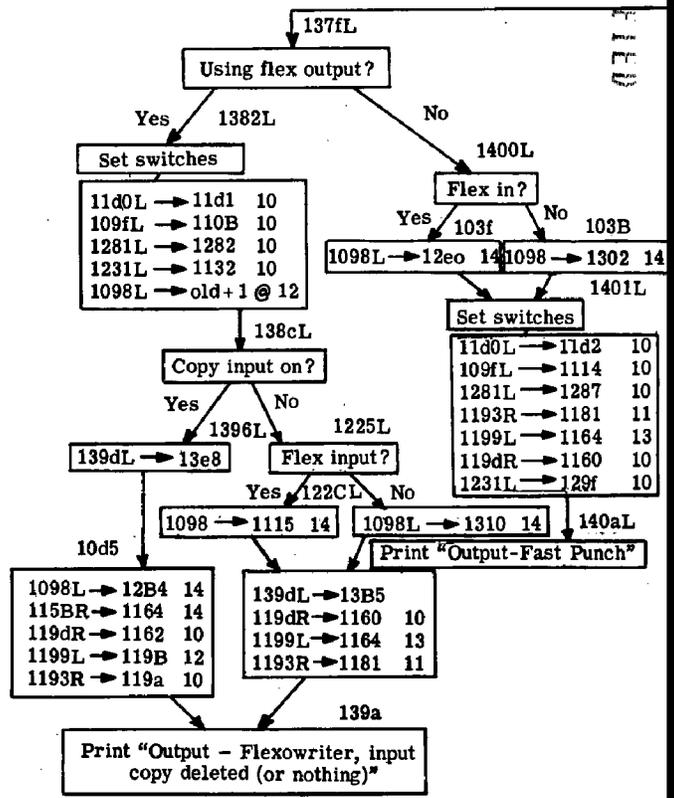


Fig. E1 - Control word simulator subroutine

Know flex in fast out copy in on.



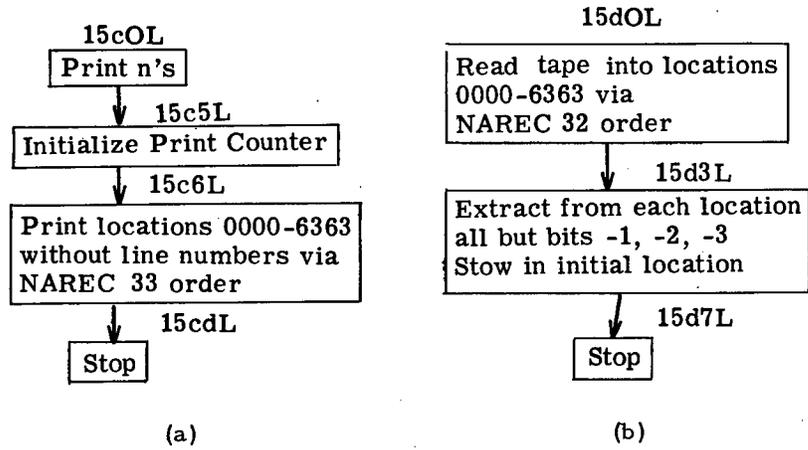
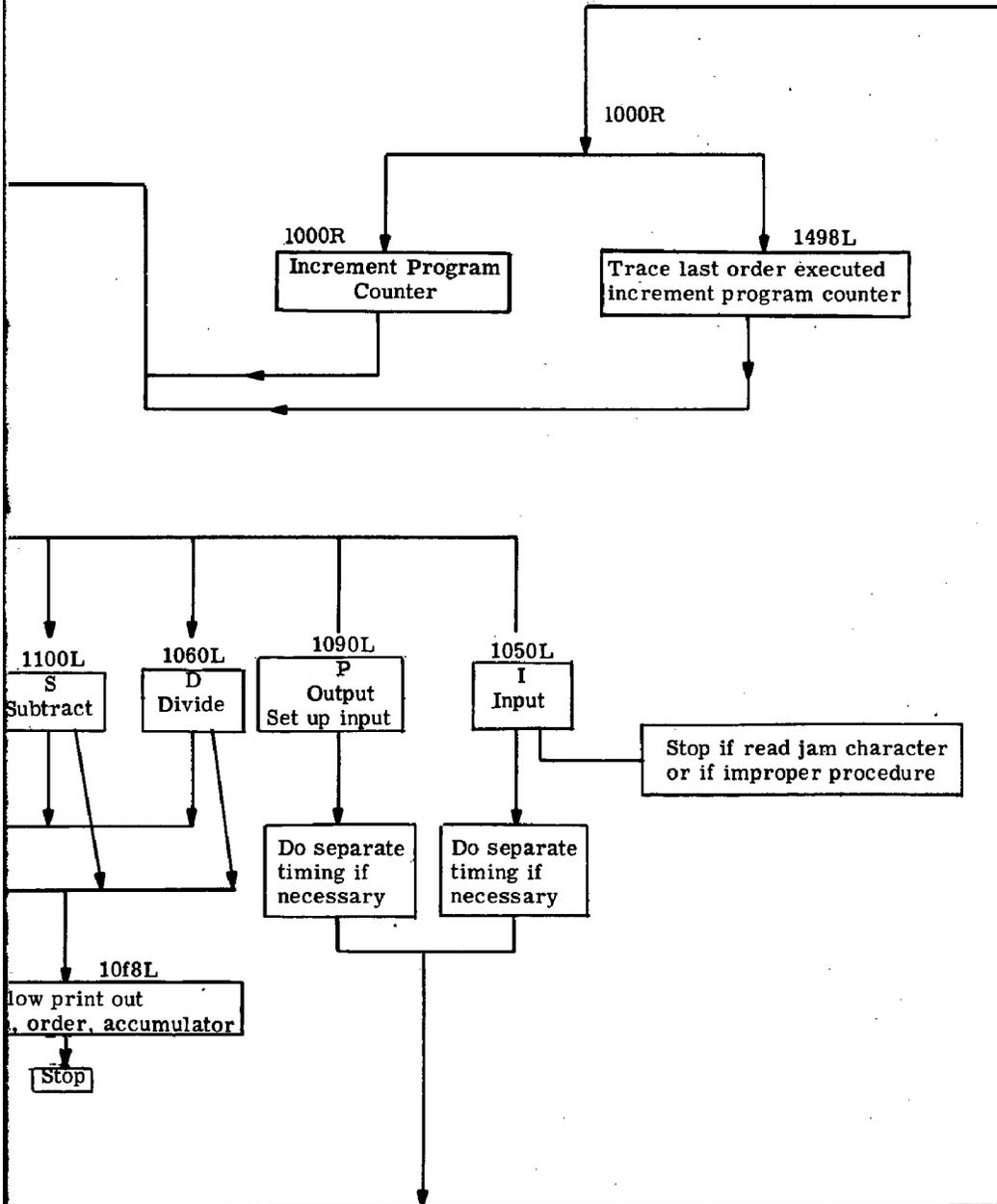


Fig. E2 - Simulator subroutines for (a) memory dump and (b) reloading of memory dump



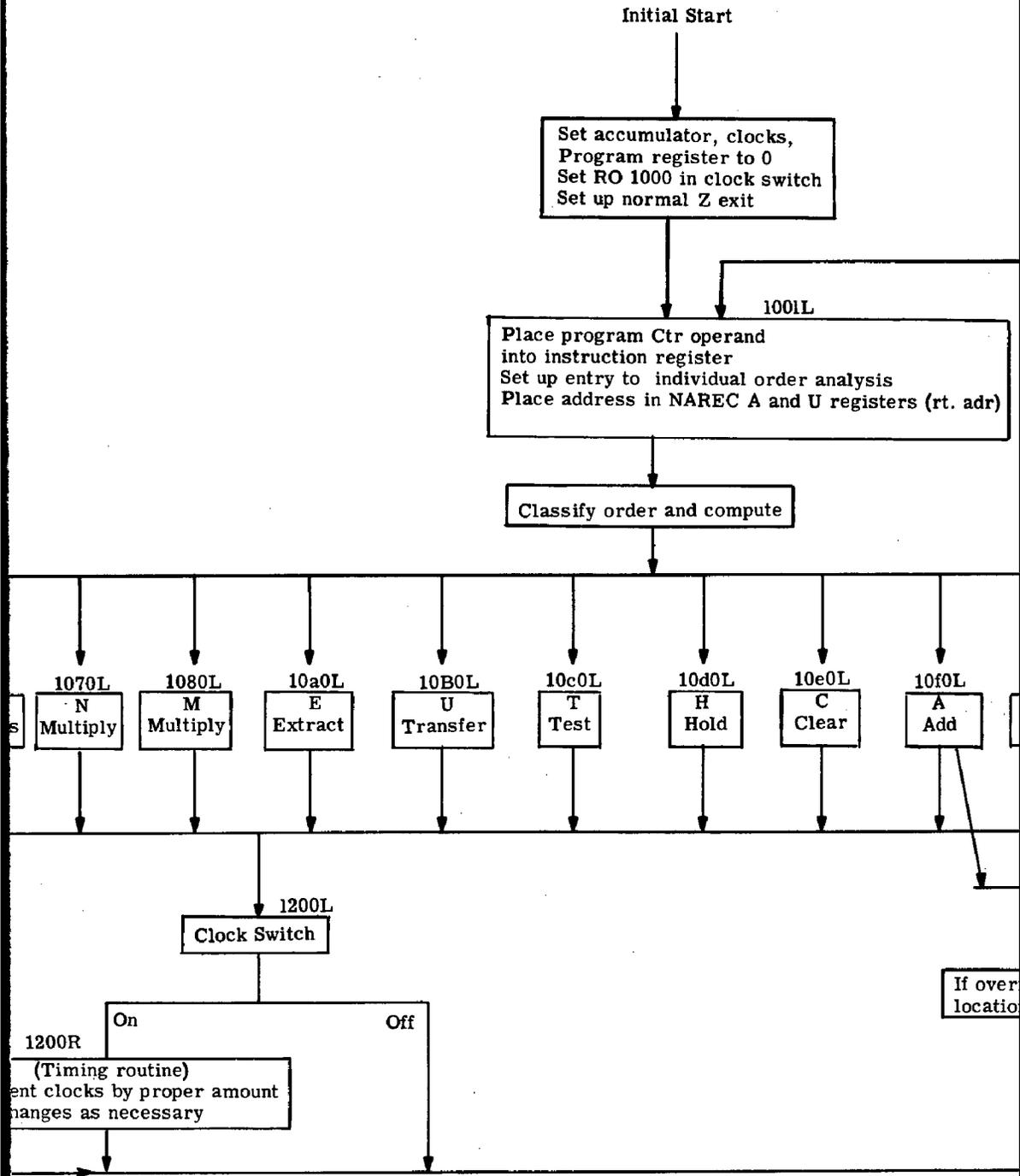
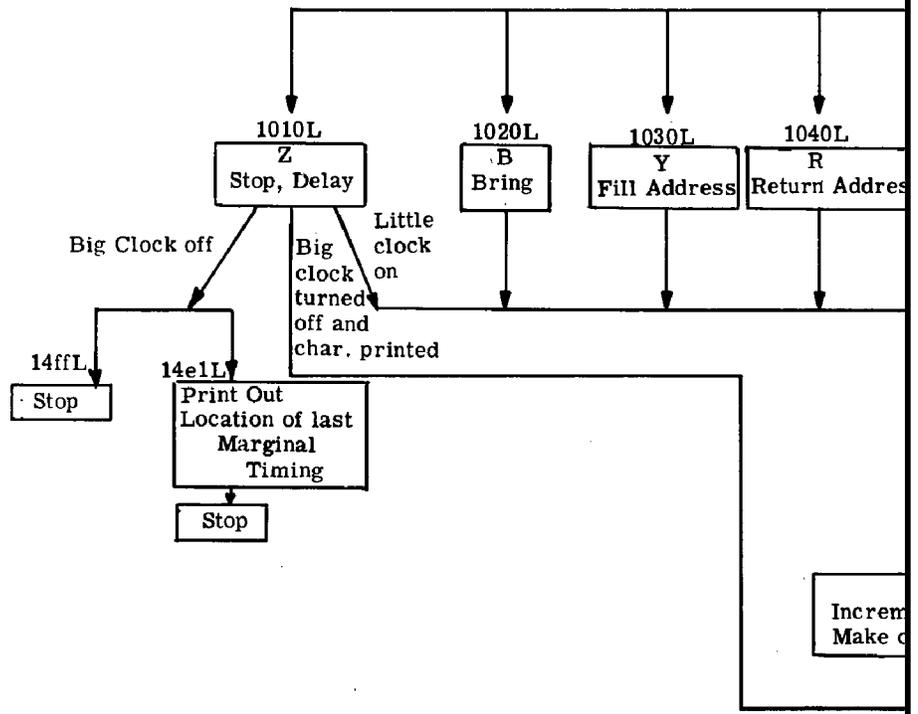


Fig. E3 - Flow of control subroutine



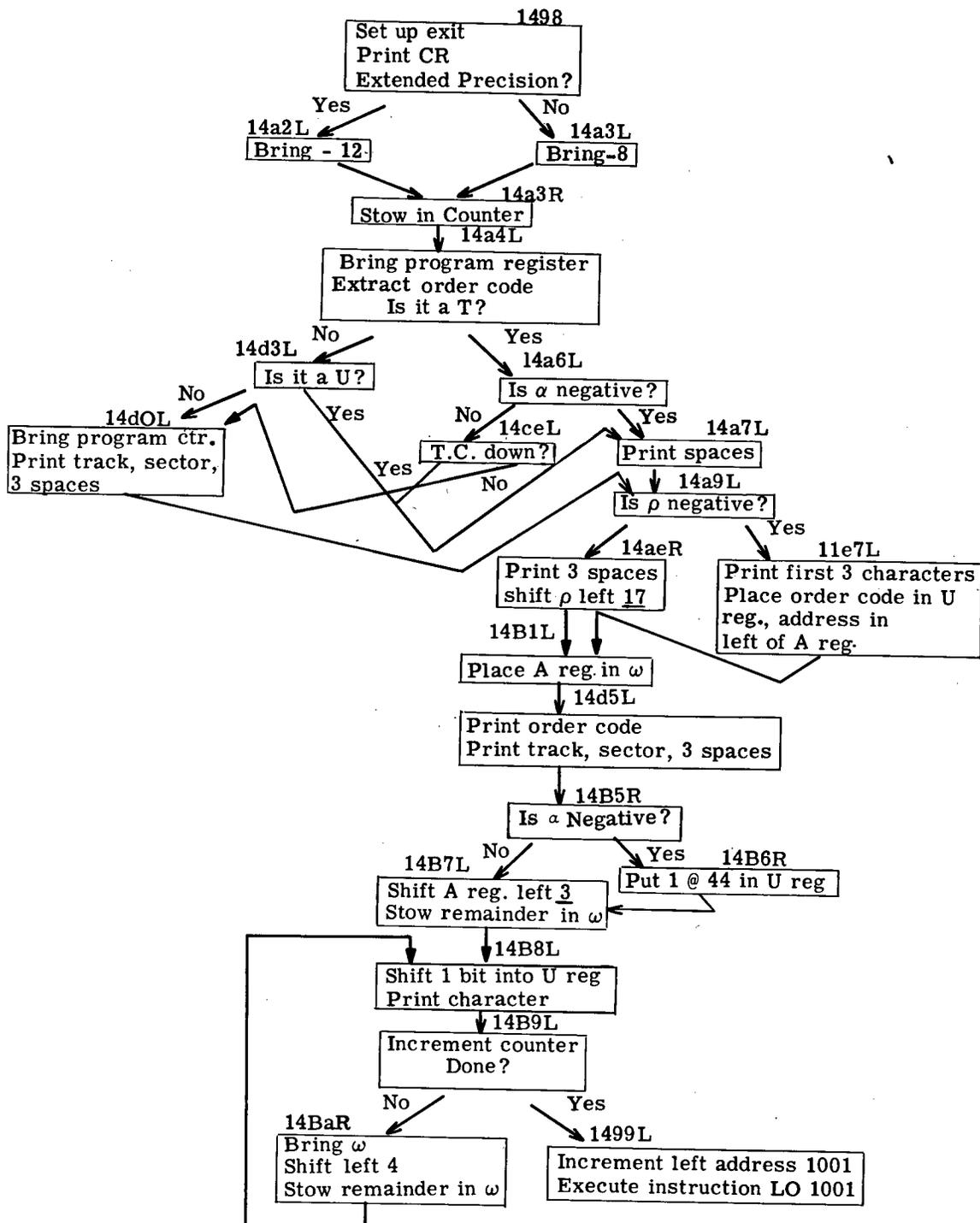
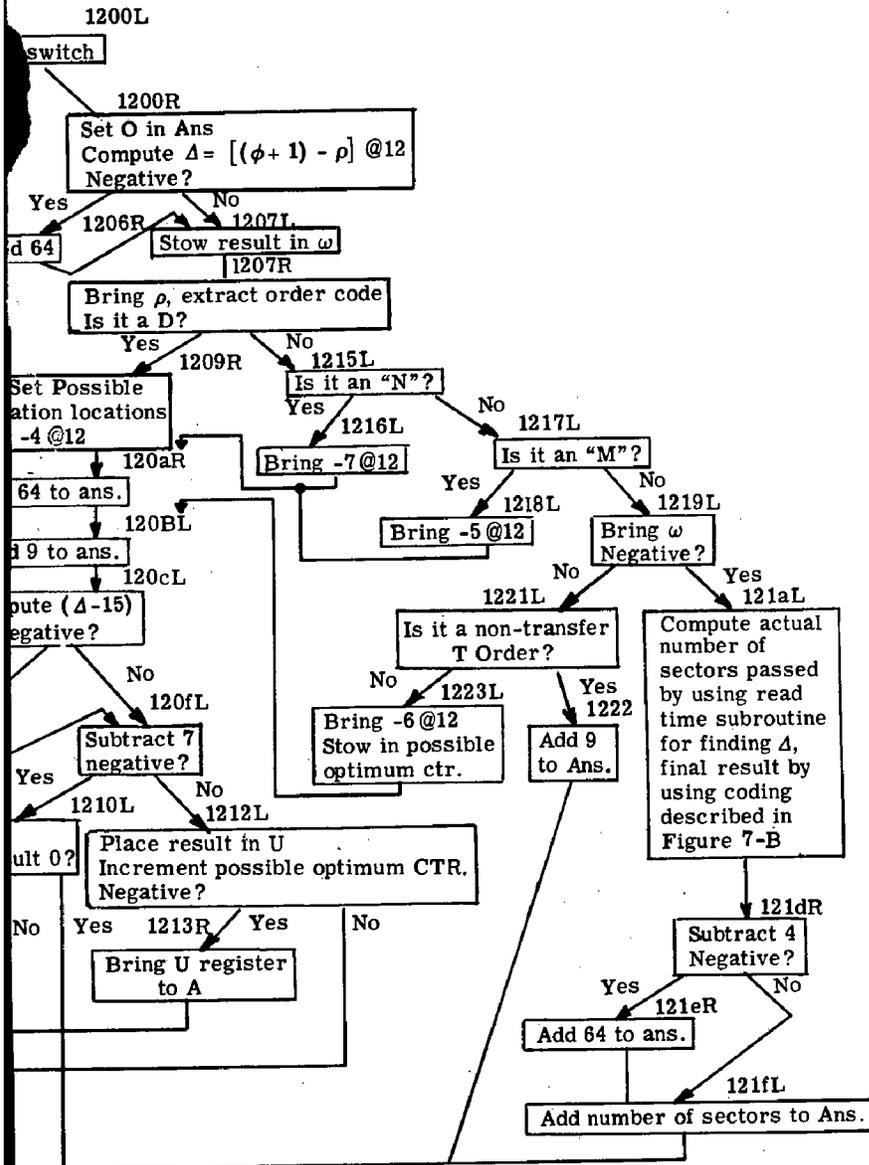


Fig. E4 - Trace subroutine

A. Main Timing Routine



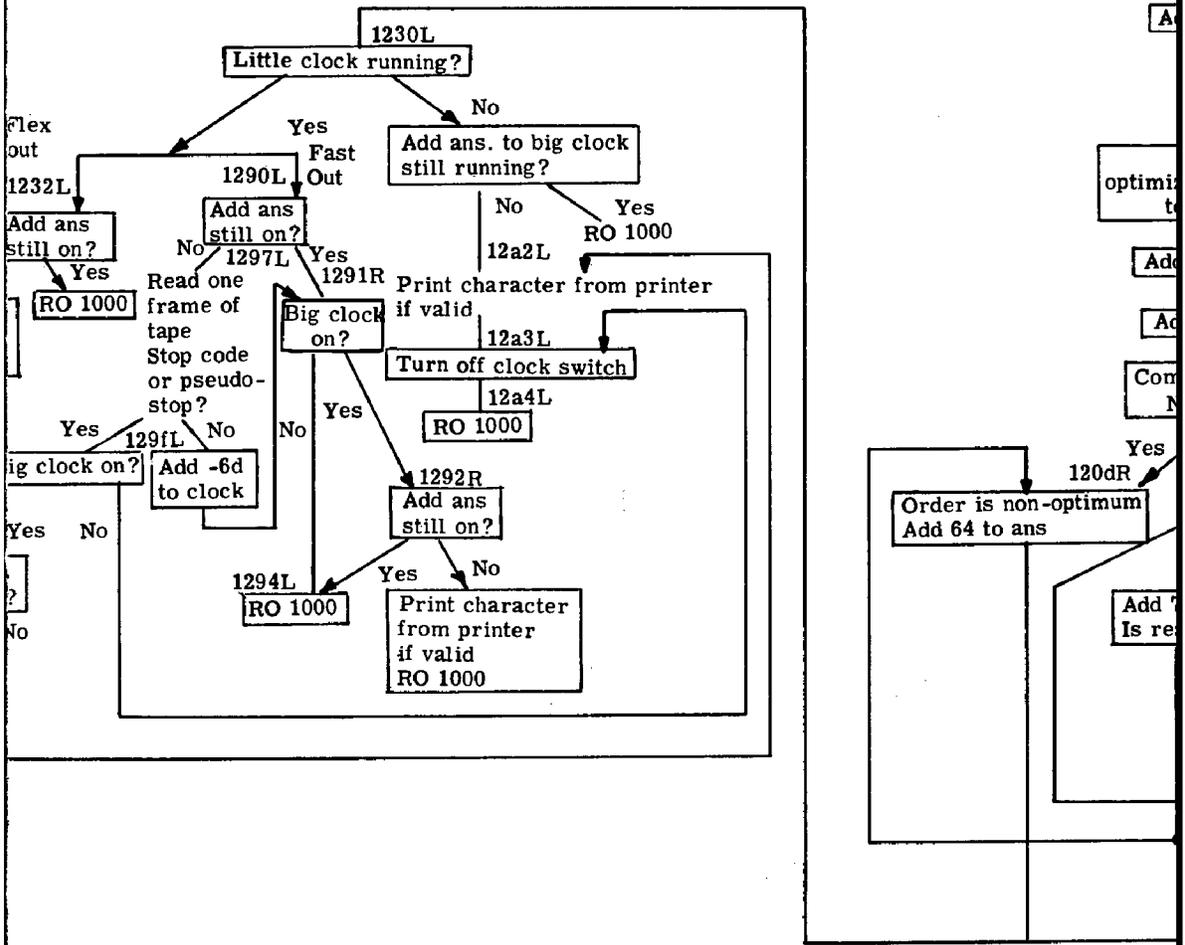


Fig. E5 - Timing subroutines

Fig. E6 - Overflow subroutine

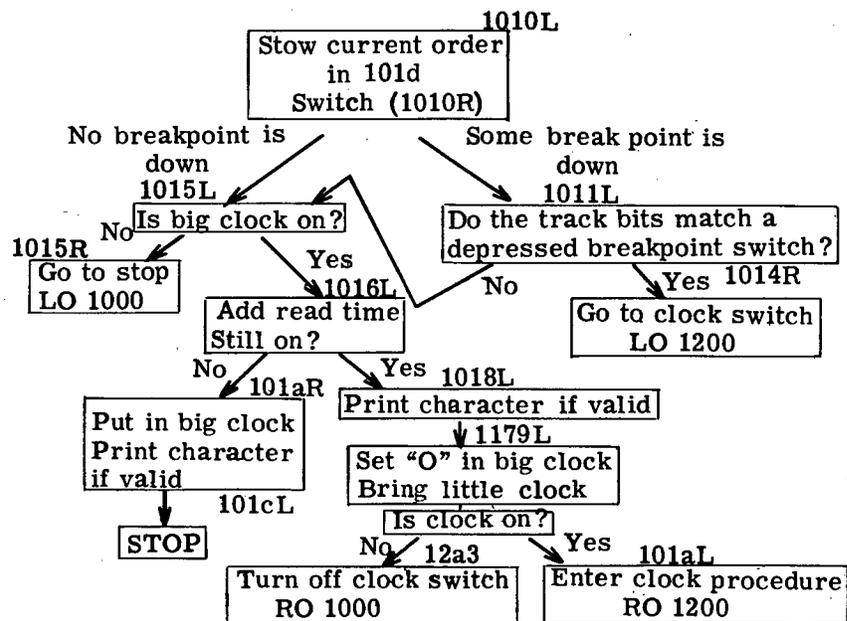
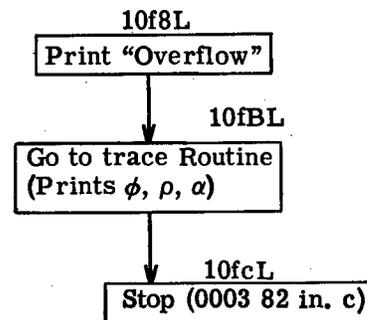


Fig. E7 - Z-order subroutine

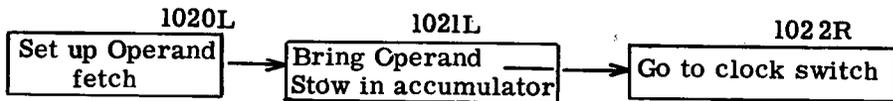


Fig. E8 - B-order subroutine

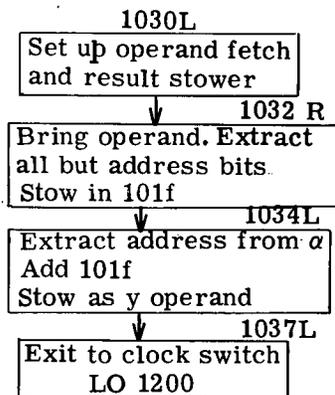


Fig. E9 - Y-order
subroutine

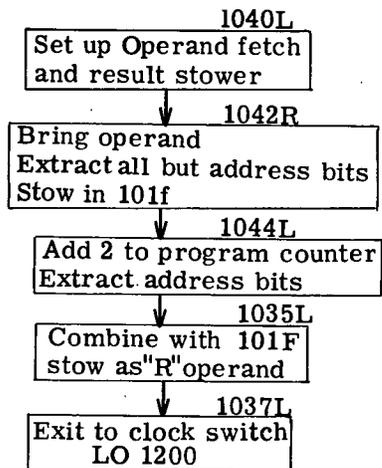


Fig. E10 - R-order
subroutine

UNCLASSIFIED

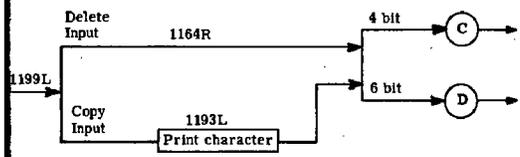
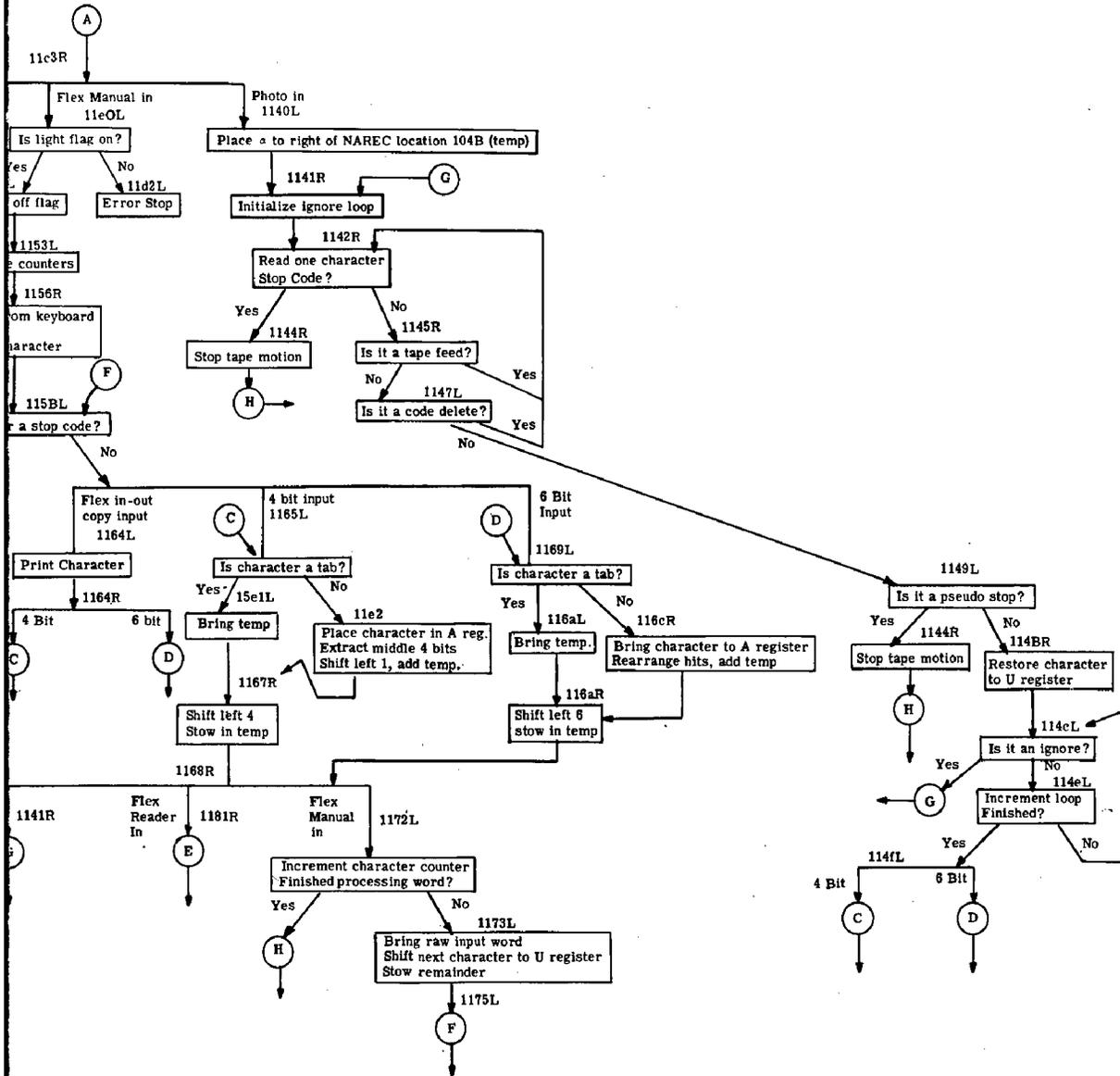


Fig. E11 - I-order

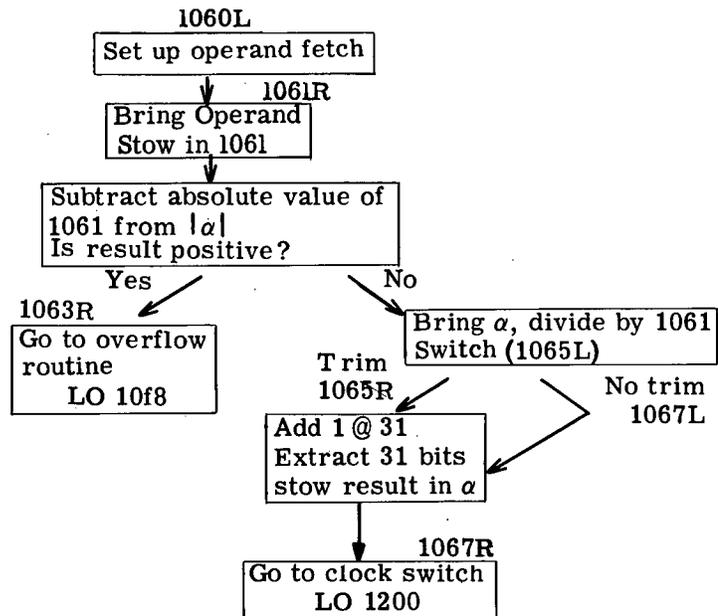


Fig. E12 - D-order

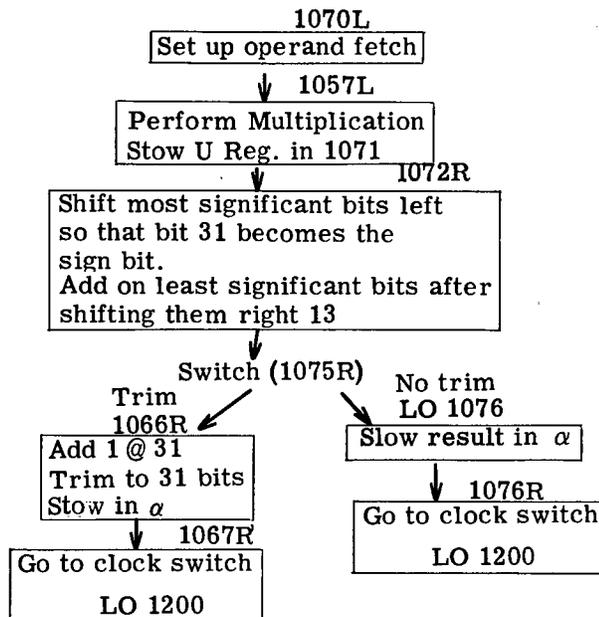


Fig. E13 - N-order

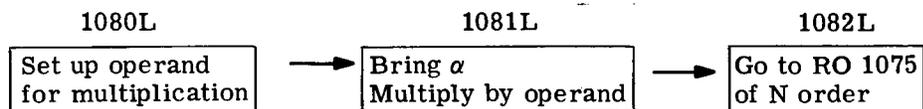
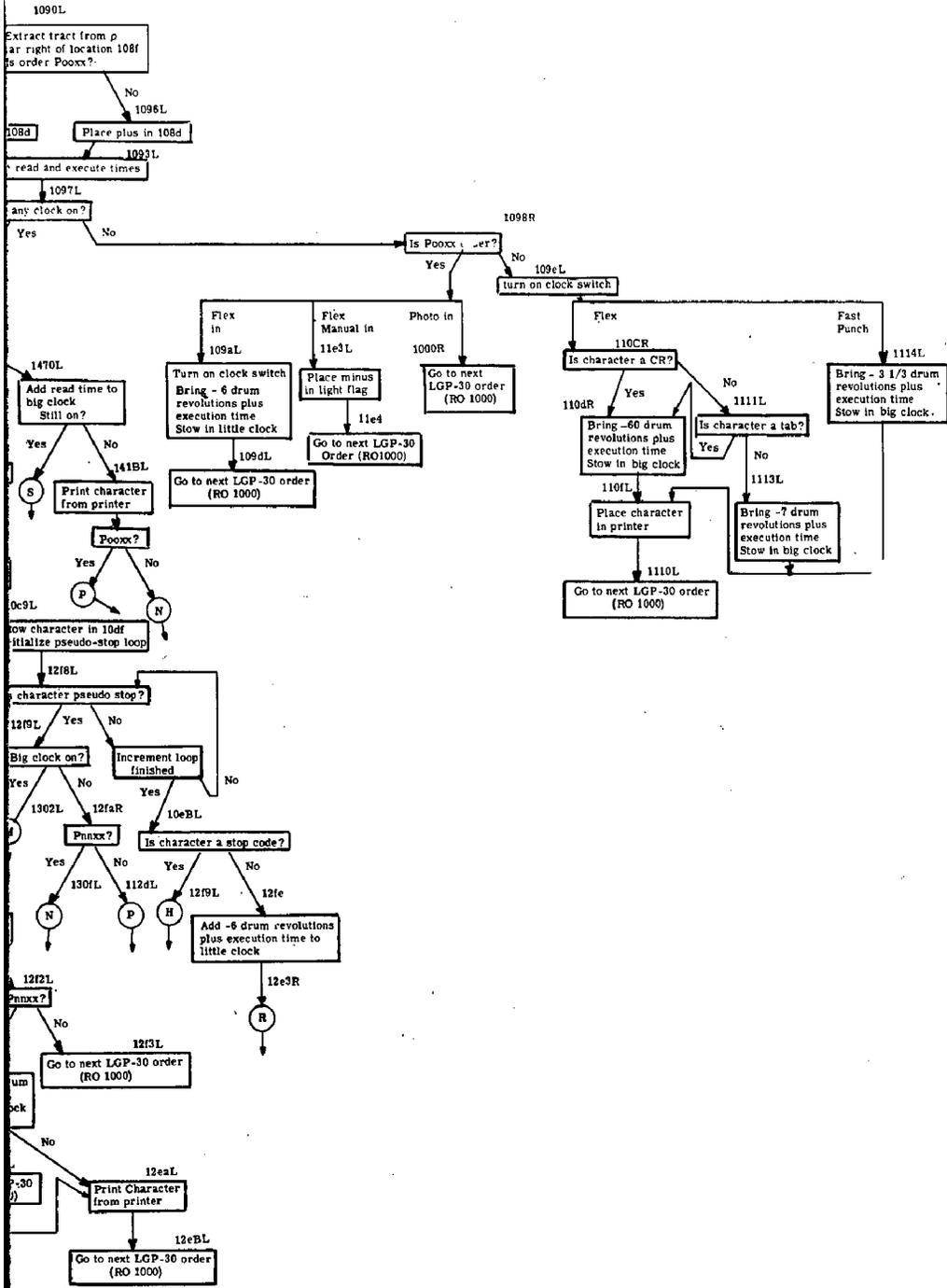


Fig. E14 - M-order



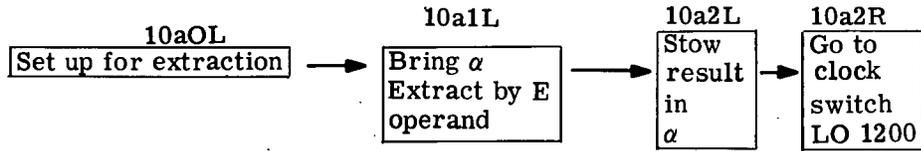


Fig. E16 - E-order

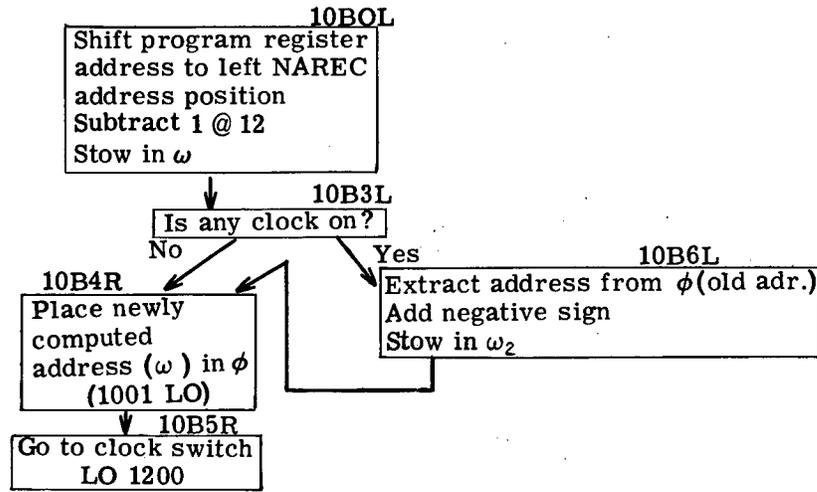


Fig. E17 - U-order

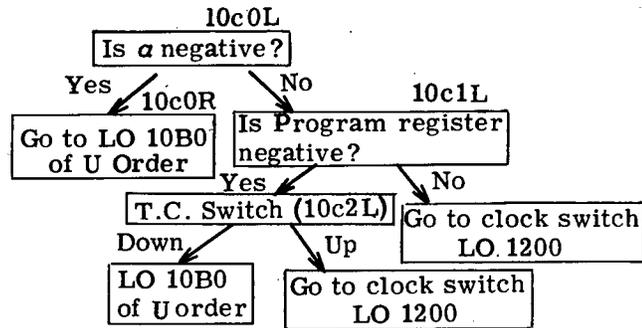


Fig. E18 - T-order

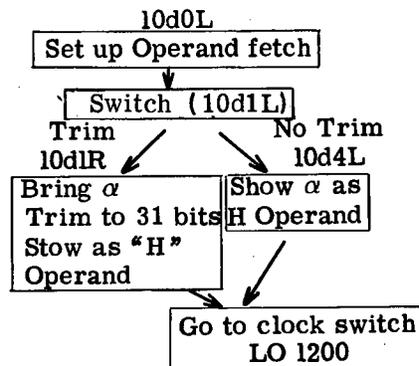


Fig. E19 - H-order

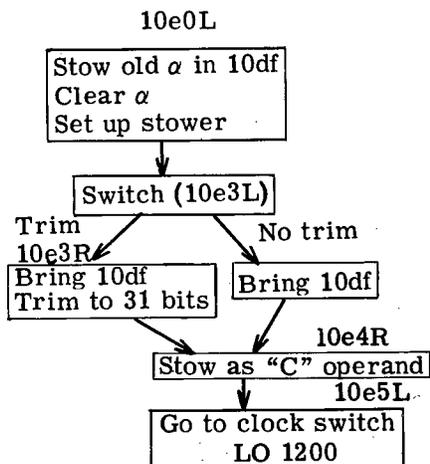


Fig. E20 - C-order

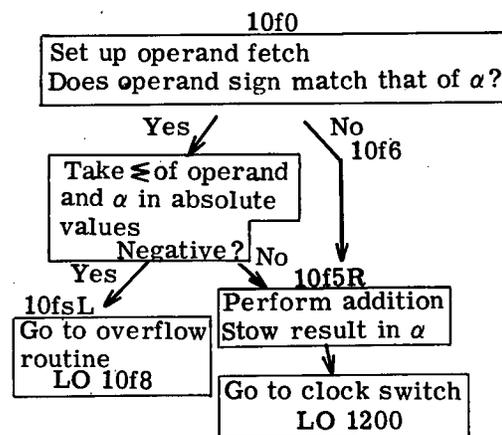


Fig. E21 - A-order

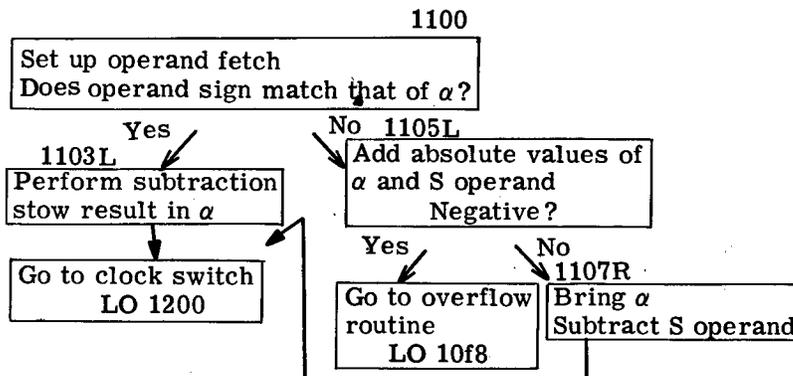


Fig. E22 - S-order