MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0049274 0

CENTRAL RESEARCH LIBRARY

ORNL-TM-3679
Vol. 1

cy. 47

REACTOR ON-LINE COMPUTER CONTROL DEVELOPMENT AT THE HFIR, VOL. 1: OBJECTIVES, SYSTEM DESIGN, OPERATING EXPERIENCE, AND SAFETY CONSIDERATIONS

J. B. Bullock
H. P. Danforth

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.
UCN-7969
(3 3-67)

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION • FOR THE U.S. ATOMIC ENERGY COMMISSION

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ORNL-TM-3679
Vol. 1

Contract No. W-7405-eng-26

REACTOR ON-LINE COMPUTER CONTROL DEVELOPMENT AT THE HFIR, VOL. 1: OBJECTIVES, SYSTEM DESIGN, OPERATING EXPERIENCE, AND SAFETY CONSIDERATIONS

J. B. Bullock and H. P. Danforth*

OCTOBER 1972

*Present address: Wyle Laboratories, 3200 Magruder Boulevard, Hampton, Virginia 23366.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0049274 0

ABSTRACT

The HFIR Computer Control Development program was initiated to evaluate and develop methods of utilizing process computers in the control and operation of nuclear reactors. The primary objective was to demonstrate the ability of an on-line digital system in calculating core reactivity balance for use in automatic control algorithms and for core anomaly detection. These objectives were realized by installing, programming, and operating an on-line digital computer at the HFIR.

The development of operator assistance functions revealed additional assets of programmable systems, particularly in data reduction, extrapolating operating conditions, and alarm analysis.

From safety considerations concerning the use of the computer in manipulating control rods, we conclude that no additional hazards are encountered when the normally required independent safety systems are provided.

ACKNOWLEDGMENTS

The authors are grateful for the many suggestions and assistance received from E. P. Epler, Consultant to the Instrumentation and Controls Division, during the planning and performance of the initial portion of this work. The essential contributions to the programming and system operation by G. R. Owens and C. E. Murphy of ORNL are also gratefully acknowledged.

NOTICE

Mention of companies, their products, or brand names in this report is for information purposes only and does not imply an endorsement by Oak Ridge National Laboratory, the U. S. Atomic Energy Commission, or the Union Carbide Corporation.

CONTENTS

	Page
1. Introduction	1
1.1 Background	1
1.2 HFIR Selected for Study	1
2. System Description	3
2.1 Description of HFIR and the Analog Control System	3
2.2 Computer System Specification and Procurement	7
2.2.1 Hardware Specification	7
2.2.2 Software Specification	8
2.2.3 Acceptance Problems	9
2.2.4 Hardware Additions	9
2.3 Executive System Software	10
2.3.1 Priority-Interrupt Scheme	15
2.3.2 Monitor	19
2.3.3 Common Interrupt Handler	20
2.3.4 Request Modules	20
2.3.5 Dispatcher	21
2.3.6 Input-Output Drivers	21
2.3.7 Job Processor	23
2.3.8 Relocatable Binary Loader	24
2.3.9 Assembler and Compiler	24
2.3.10 Source Program Editor	25
2.4 Process Scan Software Description	25
2.4.1 POWCAL	27
2.4.2 PICMID	27
2.4.3 DIFPLT	30
2.4.4 DMPCOM and DRMCOM	30
2.4.5 CONTRL	30

CONTENTS (continued)

Page

2.4.6 RHOCAL 34
2.4.7 RHOROD 37
2.4.8 CALLSC and SC 39
2.4.9 DRC 47
2.5 Operational Surveillance Software Description 48
2.6 System Support Software Description 52
3. Operating Experience 53
3.1 Software Operating Experience 53
3.1.1 Language Considerations 53
3.1.2 The Burnup Model 57
3.1.3 Automatic System Restart 60
3.1.4 Software Quality Assurance 61
3.2 Reactivity Balance 63
3.3 Direct Control of Shim Rods 68
3.3.1 Safety Considerations 68
3.3.2 Misoperation During Early Tests 73
3.4 Hardware Operating Experience 77
3.5 Operator Control Console Experience 84
4. Appendix 95
4.1 Specification SCS-1: HFIR On-Line Data-Acquisition and Process-Control Computer System 97
4.2 SCAN Programs 117
4.3 Logic Diagrams for Support Software 129
4.4 Operating System Loading Procedure 177

1. INTRODUCTION

1.1 Background

The increasing complexity of nuclear reactor control problems coupled with the growing economic incentive to operate plants as closely as possible to the design limits has necessitated serious consideration of fully automated systems for reactor control. However, control system designers have been justifiably reluctant to close the loop and automatically withdraw shim rods at the request of a flux or power level controller without confirming the validity of the reactivity demand. As a result, the reactor operator is vested with the responsibility of manually performing shimming operations. In either case, one should assess whether requests for reactivity addition are reasonable, or whether, in fact, there exists some reactivity anomaly indicative of an abnormal and perhaps dangerous situation. In most cases, the data processing required to decide whether or not a reactivity anomaly exists is beyond the capability of an operator when he is not supplied with computational assistance.

Nuclear reactor operating experience has been marked by the repeated occurrence of a class of control malfunctions which might have been prevented if the reactivity state of the nuclear system had been utilized as a control parameter.[1] Conventional analog servo-control systems are not capable of utilizing the core reactivity balance calculation and, as a result, are vulnerable to erroneous reactivity demands. Typical system malfunctions that fall in this category are: loss of a control parameter input to the servo; component failure in the servo channel; anomalous core changes, such as coolant-flow blockage or fuel movement; and erroneous readouts to an operator, such as control rod position. Many of these malfunctions would lead to a change in the reactivity state of a nuclear system before actual damage occurs, and therefore an on-line, real-time calculation of the system reactivity balance could be used to establish safe operating contours.

1.2 HFIR Selected for Study

An investigation to determine the feasibility of utilizing a general-purpose, process control digital computer to supplement a conventional control system for a

nuclear reactor was initiated at the Oak Ridge National Laboratory (ORNL) in 1965. The primary justification was the need to develop and demonstrate a method for assessing, in real time, the reactivity state of an operating reactor. The objectives of the study were expanded to include the determination of reactor control problems that are not being solved adequately by analog methods but could possibly be solved by digital techniques. The major tool acquired for and used in this continuing study is a process control digital computer system (Control Data 1700) connected to the High Flux Isotope Reactor (HFIR).

The HFIR exhibits, in addition to the usual burnup phenomena, rapid and complex ^{135}Xe poisoning effects[2]. During a transient which follows a decrease in reactor power as a result of a legitmate reactor scram, an unplanned control rod drop, an ac power failure, or the partial loss of coolant, it is virtually impossible for the operator to evaluate the validity of shimming requirements because of the additional change in reactivity brought about by the rapidly changing xenon concentration. If such a decrease in reactor power occurs, it is desirable to return to full power as soon as possible, at the same time being able to recognize an anomaly which may have been the unrecognized cause of the transient. If the reactor cannot be returned quickly to full power, an unnecessary shutdown due to xenon poisoning can occur. Further, the core could require premature replacement owing to samarium poisoning, inasmuch as partial refueling of the HFIR is not possible.

To reduce the probability of losing a core to xenon, the computer system was evaluated as an automatic controller to reestablish control rod symmetry following a rod drop or power outage. Since the HFIR was designed to operate near burnout, the control rods must be positioned symmetrically before the power level can be increased above 10% of full power. The rapid recovery of rod symmetry, therefore, reduces the probability of xenon poisoning by minimizing the reactor operating time at 10% of full power.

Several additional programs were developed as a bonus from the two main studies of reactivity balance and digital rod control. These include: projected end of cycles, projected time to poison, calculation of plant parameters for limit checking and display to the operator, automatic computer restart and update system,[3] and safety

system annunciator scanning.[4] Additional programs were developed to aid in reactor operations and reactor diagnosis research. All of these tasks were integrated into the overall process control system to gain more information about the capabilities and limitations of an on-line computer in a reactor facility. The following description of the HFIR will provide the basis for a more detailed discussion of the interactions between the digital computer and the analog control system.

2. SYSTEM DESCRIPTION

2.1 Description of HFIR and the Analog Control System

The HFIR is a 100-MW, water-cooled, enriched-uranium research reactor. The primary function of the reactor is to produce ^{252}Cf in the centrally located flux-trap region of the two-piece annular core.[5] Because of a reactor high power density and an inability to partially refuel the core after a ^{135}Xe transient, several control functions were designed to minimize spurious reactor shutdowns. For example, the protection system will not scram the reactor if a power outage occurs, because a 10% primary coolant flow can be maintained by battery power. The control system, also operating on batteries, will automatically reduce the reactor power to 10 MW to compensate for this lower flow.

The HFIR was designed and constructed with a sophisticated analog control system[6] which requires only the actuation of a start button to automatically increase the reactor power from source level to 10 MW. The power is controlled at levels between 10 and 100 MW with a conventional analog servo by setting the power demand to the desired value. (The basic features of the control system are shown in a logic diagram in Fig. 1.) The regulating rod responds to errors between the demand power (set by the operator) and the power measured by three servo channels. The velocity of the regulating rod is proportional to the error and reaches a maximum of 0.25 in./sec when the error exceeds 1 MW. Although the regulating rod velocity is high by some standards, the amount of reactivity that it can add is limited to less than 1.00 dollar by its 1-in. stroke. The position and the major limits are displayed to the operator

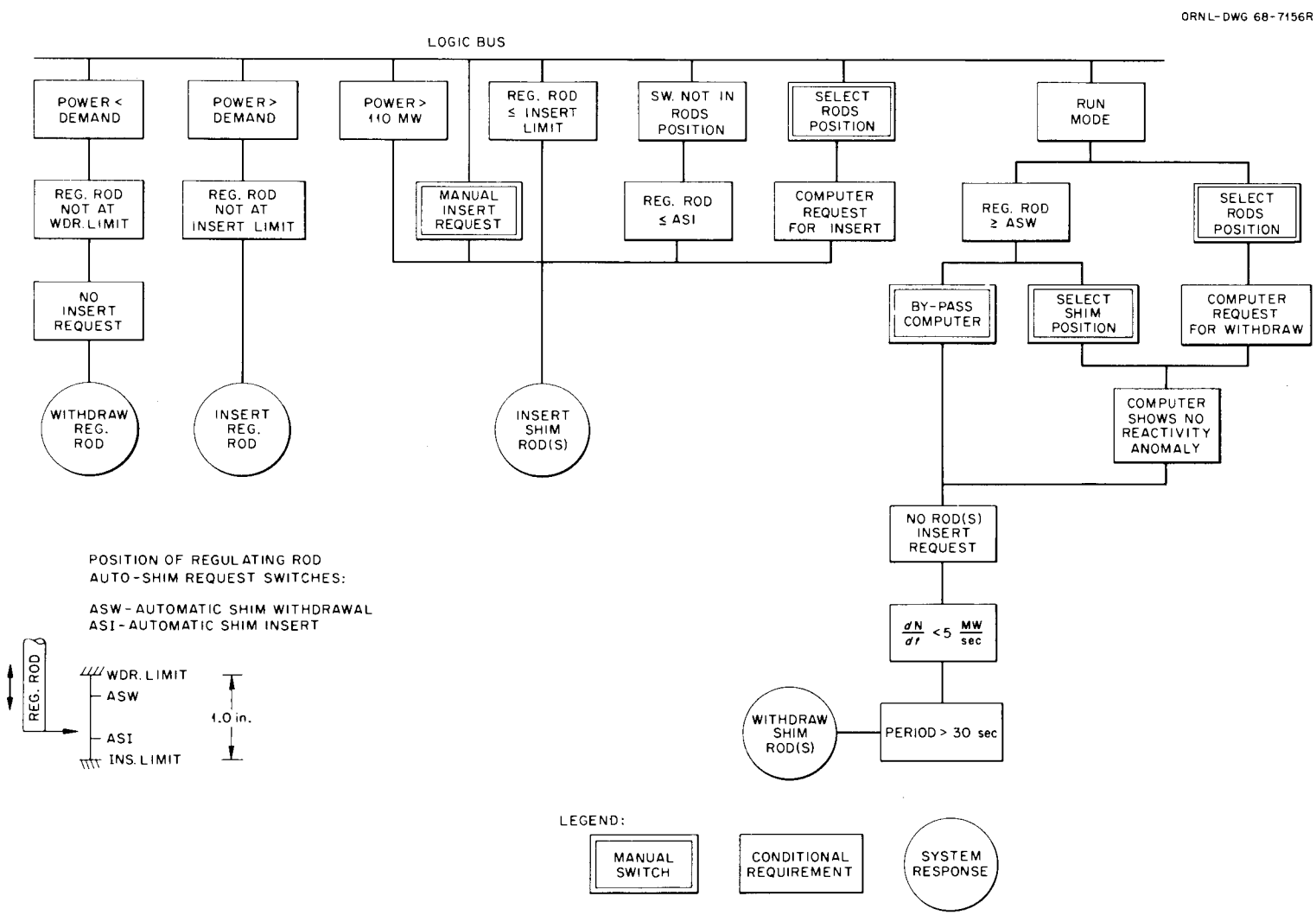


Fig. 1. Basic HFIR control system logic diagram.

by the method shown in Fig. 2. When the regulating rod approaches a limit during its movement, a switch automatically closes to indicate the limit position of the rod. When the rod is withdrawing, for example, the automatic shim-rod withdrawal switch (ASW) will close and indicate to the control system that an addition of reactivity is needed by the shim rods if the demand power is to continue to be supplied. In a similar manner, an automatic shim-rod insert (ASI) request is generated when the regulating rod approaches its insert limit. The locations of the switches are pictorially represented in Fig. 1. Very few conditions must be satisfied to realize an automatic shim-rod insert, whereas the addition of positive reactivity is more highly conditional.

Although automatic analog controllers have been designed to relieve the operator of direct-rod-manipulation duties, human judgment has traditionally been required to perform the important function of limiting the total reactivity that may be added by the automatic (servo) system. This form of administrative control is acceptable for many reactor systems, because the required changes in reactivity are not very complex and can readily be interpreted by the operator. In the case of the HFIR, however, the high power density can result in large and rapidly changing ^{135}Xe poisoning effects, which are a complicating factor. During a normal reactor cycle, the excess reactivity change is 13.80 dollars over the 23-day core life. The rate of change during the first 12 hr is only 0.40 dollar/hr; however, the rate of change following a power reduction varies from 9 to 12 dollars/hr, depending on the integrated power. Therefore, during a reactivity transient following a decrease in reactor power as a result of a legitimate reactor scram, an unplanned control-rod drop, an ac power failure, or the partial loss of coolant, it is virtually impossible for the operator to evaluate the "normal" reactivity requirements because of the rapidly changing Xe concentration. When such a decrease in reactor power occurs, it is desirable to return to full power as soon as possible, while at the same time being able to detect a reactivity anomaly which may have been the unrecognized cause of the shutdown. If the reactor cannot be returned quickly to full power, an extensive shutdown can occur as a result of ^{135}Xe poisoning. In this situation, the administrative control method does not appear to be adequate, and little success has been realized in applying conventional analog methods.

Fig. 2. Regulating rod position indicator.

2.2 Computer System Specification and Procurement

2.2.1 Hardware Specification

One of the most important factors affecting the choice of a digital machine was the necessity of real-time reactivity calculations which require rapid solution of a set of 10 differential equations describing the concentrations of fuel, poisons, and fission products in the reactor core. Since the calculation involves small differences in very large numbers, along with many iterations, both round-off and truncation errors can affect the accuracy of the solution.

A study was made using a CDC 1604 computer at ORNL to determine the feasibility of using a small process computer for reactivity calculations. The results indicated that a computer with a 16-bit word length and a memory cycle time of <4 μsec would have adequate accuracy and speed. An alternative to specifying cycle time, multiply time, and the number of bits per word was considered, i.e., that these parameters could best be defined by including a representative program in the specification package and require that the execution of this program yield a specific answer within a given time.

A flexible input-output structure was also required, because the computer system would be a research tool, and it was expected that changes and additions would be made as functions were added or modified. Studies were made to ascertain which reactor variables should be monitored. Engineering development for signal acquisition and conditioning was begun as the computer specifications were being prepared to ensure that interface problems would be minimized. Analog inputs were required from pneumatic transducers, position-sensing potentiometers, neutron detectors, and other devices unspecified at that time. To provide for the necessary inputs, two analog input subsystems were specified: a high-level (±5 V), high-speed (5000 points/sec) subsystem; and a low-level (10-50 mV), low-speed (200 points/sec) subsystem. The high scan rate for the high-level subsystem was included for neutron fluctuation analysis. Digital contact inputs were used to monitor safety annunciators and rod position sensors.

The output signals consisted of digital relay drivers for rod control and one analog output to drive a recorder that is used to continually display the difference between the calculated burn-up reactivity and the reactivity added by the rod withdrawals.

Communication between the reactor operator and the computer was one area of investigation. The present methods include:

1. a heavy duty teletypewriter for program writing and off-line program processing,
2. a Selectric typewriter for alarm messages and process logs,
3. a digital display (lights) to display process variables,
4. console function switches for entering operator requests to the computer.

High-speed paper tape is the main binary input-output device for loading and punching programs.

The decision as to the amount of core memory required for the system was the responsibility of the computer vendor; however, a minimum of 12K words was specified. The functional descriptions of the system tasks provided bases for the amount of core actually included in the computer.

Originally a back-up power supply for the system was specified, but it was later deleted because of economic considerations.

Since the system was to be used for reactor control, the control and scanning programs were to respond to changes in the state of the reactor system according to the importance and time response of an individual control loop. This consideration, along with a desire to utilize the computer in a background-foreground mode, required inclusion of a priority interrupt system. Sixteen levels of hardware priority were specified to allow for both control interactions and peripheral equipment responses.

2.2.2 Software Specification

In anticipation that the process control programming could all be accomplished in Fortran to reduce the programming training time for the control engineers, the software specification required use of a Fortran language that could operate in an interruptable process environment. This required that the Fortran callable subroutines be reentrant and capable of easy communication with assembly language programs.

The usual standard library arithmetic functions were also specified. The efficiency of these routines and the compiler was compared among vendors through the use of the Manufacturer's Test Program included in the specification. A copy of the original specification is included in the Appendix, Sect. 4.1.

2.2.3 Acceptance Problems

Software deficiencies was the primary reason for a 1-year delay in accepting the system. Although the original delivery date specified for the system was December 15, 1966, hardware checkout and software integration problems delayed the delivery until November 2, 1967. Checkout of the system by ORNL personnel revealed numerous deficiencies in the original software. After several months of on-site programming, the system was conditionally accepted on May 27, 1968.

The major problems included: interference of operator console functions with normal system operations, integration of process programs into the overall operating system, interference of job processor (off-line) programs with normal process control, lack of a program for the low-speed paper punch and read on the teletypewriter, limitation of the high-speed scan to 2 kHz, and a general lack of documentation.

On April 4, 1969, a final settlement was authorized, and the system was accepted.

2.2.4 Hardware Additions

In addition to the planned digital control studies, a proposal was accepted for on-line reactor diagnosis experiments using neutron fluctuation analysis techniques. The high-speed analog inputs could be used effectively for auto- and cross-correlation calculations if the computer system had a bulk memory device. This need, along with the added capability of a more flexible system software configuration, led to the addition of a 512K-word, fixed-head disk storage unit, which is usually referred to as a "drum memory unit."

To move all reactor control rods, incorporate shim permissive circuits into the HFIR analog control system, and allow spare outputs for control studies, the total

number of power-relay-driver output circuits was increased to 32. The number of digital-to-analog outputs was increased to four to allow for both X-Y plotting and strip-chart trend recording.

Several digital inputs were added to accommodate a Scanivalve pneumatic multiplexer. The inputs to this scanner are from pressure transducers connected to the primary coolant system of the reactor and are used to calculate the flow patterns through the reactor vessel.

The present configuration and photographs of the HFIR Data Acquisition and Digital Control system are shown in Figs. 3-5.

2.3 Executive System Software

The HFIR executive system software is based on a hardware configuration containing fast core memory (both fixed and allocatable) and bulk memory. Each core memory word is provided with a "protect" bit which prevents instructions not having the bit set from changing any protected words. This permits an executive system design that can safely accommodate on-line program debugging requirements.

The system program locations are arranged in the manner described in Fig. 6. The core memory is composed of two fixed regions that are generally protected. The region from hexadecimal address $0 to $4300 is used only by programs that remain in the location in which they were originally loaded by the System Initializer program. The allocatable area of core is divided into protected and unprotected. The protected area is used only by drum resident programs which are a part of the on-line system. These programs are manipulated by the executive monitor on a priority-demand basis and usually operate only in the protected allocatable area. However, the unprotected area (UPA) can be used when adequate space is unavailable in the protected area. This process, called "swapping," consists of halting the execution of any program in the UPA, transforming the entire region to the Scratch area on the drum, setting the protect bits in the UPA, and reading the protected program into core from the guarded drum as usual. When the monitor is notified by the protected program that execution has terminated, the reverse process is followed.

ORNL DWG 72-7250

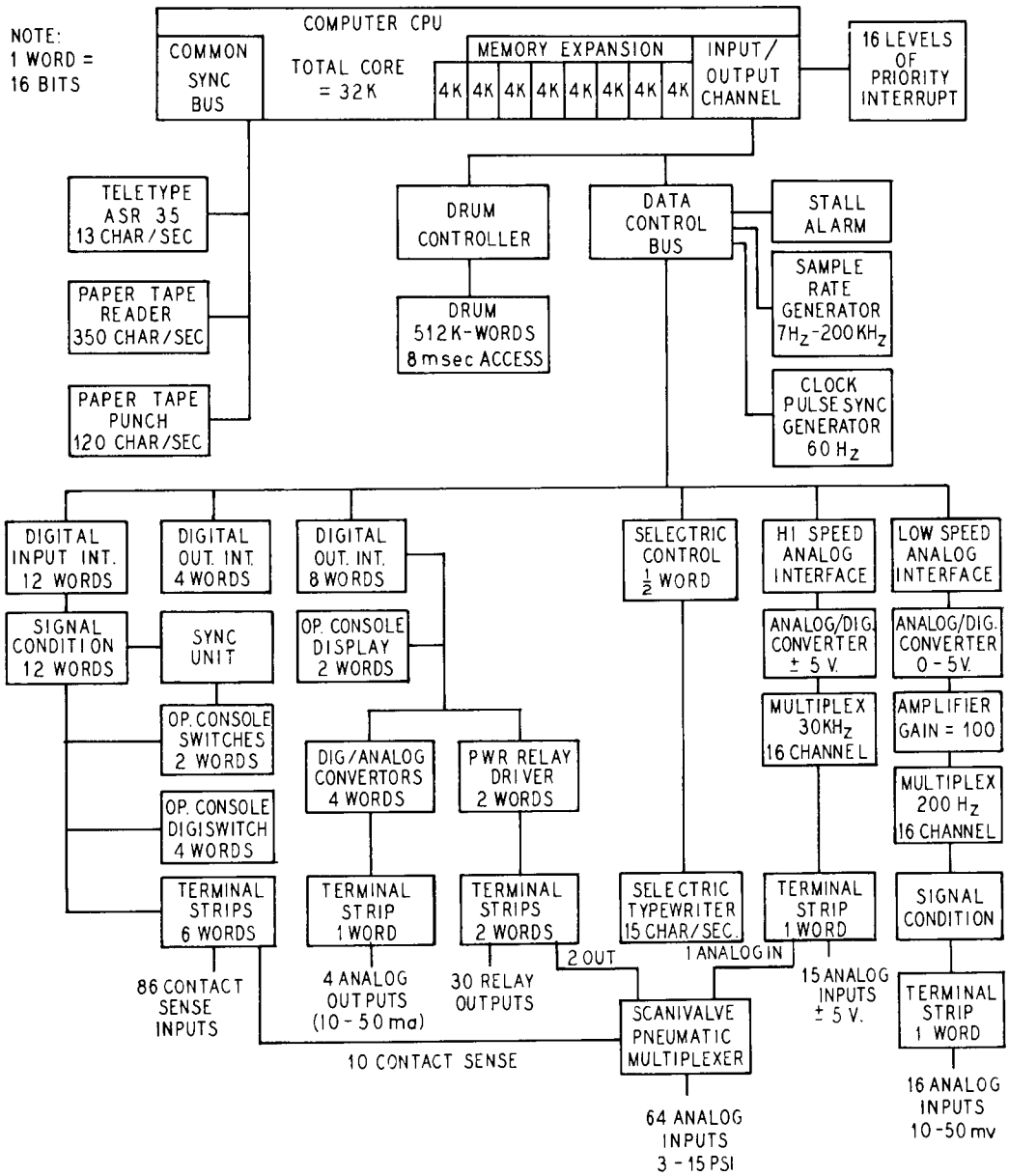


Fig. 3. HFIR computer system configuration.

Fig. 4. Computer with input-out equipment.

Fig. 5. Operator console and message typewriter.

72-5865

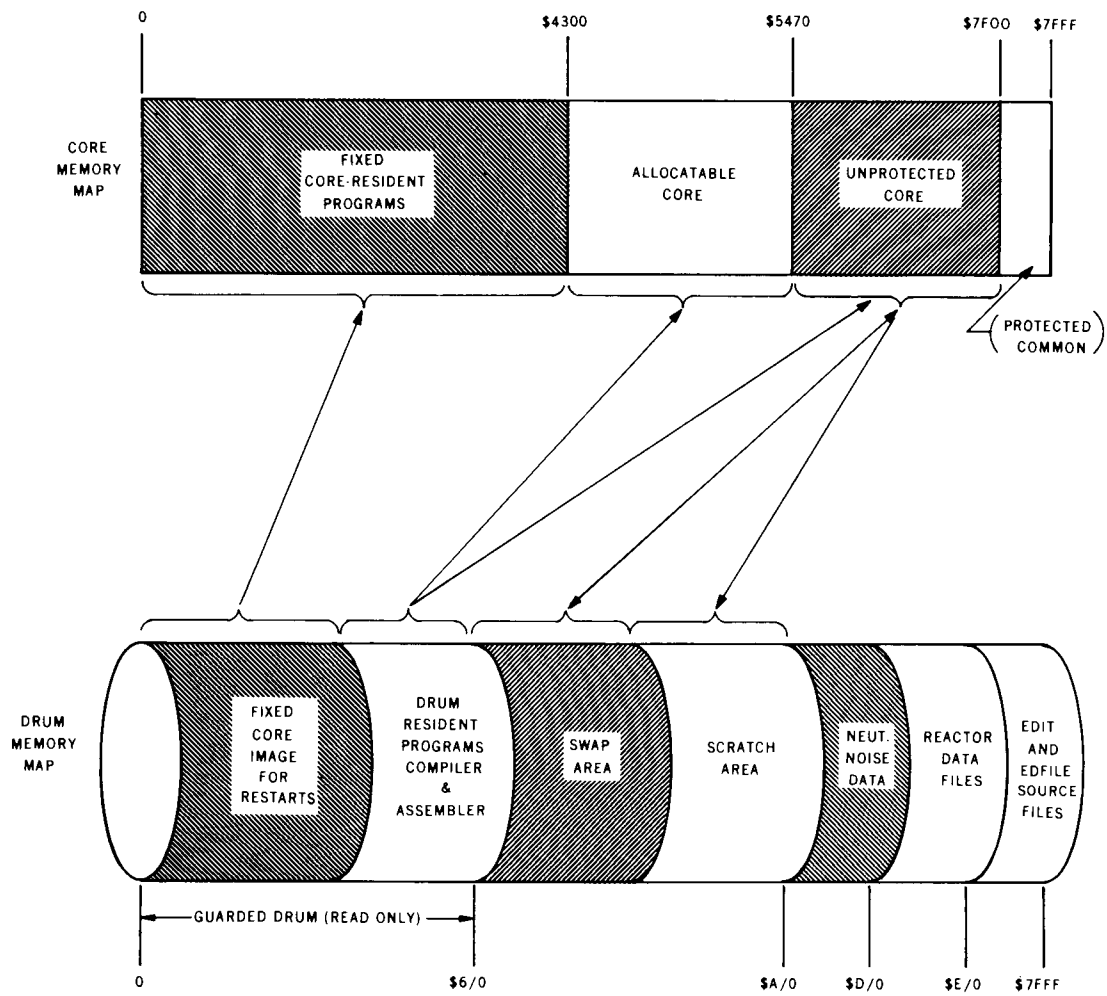


Fig. 6. Drum memory space-allocation.

The UPA is generally available for low-priority off-line program compilation or assembling or execution. The executive monitor Job Processor communicates with the user through the teletypewriter. When the Fortran compiler is called, it is loaded from guarded drum into the UPA where it executes by using multiple passes. During the course of a normal assembly or compilation, the tables of variable names and addresses will be written onto the drum. (The area reserved for temporary use by these programs is termed the "scratch area.")

To prevent accidental modification of the drum resident programs, the user may manually set guard switches, and the monitor will not violate them. After the operating system is built, the switch positions are calculated by program PATCH and set accordingly (see Fig. 7).

The operating system executive programs were furnished by the computer manufacturer and modified to conform to the needs of the HFIR. The experimental aspects of HFIR operation require not only an on-line, real-time process control system, but also an off-line mode for program development while the process is being controlled.

The operating system is based on a priority-interrupt scheme and includes the following programs: monitor, job processor, source program editor, library editor, and an on-line program modifier. These programs and their major subroutines are shown in Fig. 8 and are described below.

2.3.1 Priority-Interrupt Scheme

The priority system is used to establish a queue of process programs for execution. Each user program is written with an ascribed priority level from 0 to 15, with 15 being the highest priority. In addition to this software control, there is a hardware external interrupt system connected into the computer's data and control bus. These 15 computer interrupt lines use signals from the external process equipment to initiate the execution of specific computer programs. Each of these interrupt lines can accommodate 16 input signals from different devices. Besides the 15 external interrupt lines, one internal interrupt line responds to power failure, parity errors, or protect faults in the central processor. The priority level of each external interrupt line is

established by the user and set by software. The interrupt levels established for the HFIR system are shown in Table 1.

Any interrupt of higher priority will interrupt the computer while it is processing a lower priority interrupt. The function of the monitor is to save the registers and program counter of the interrupted program and to reinstate them after the higher level program has finished. In this manner all interrupts are properly placed on a queue and executed according to priority level. The software priority assigned to a particular process program is coordinated with the hardware interrupt system by the monitor through the use of the mask register. Priority levels 5, 6, and 7 are required for programs utilizing floating-point operations. Interrupts of these levels are treated differently because the two-word core memory locations, which serve as floating-point registers, must be saved and reinstated along with the hardware registers.

Typical examples of operating hierarchies are shown in Table 2.

Table 1. HFIR computer priority levels

Interrupt Line Number	Priority Level	Unit
3	15	Programmable Sample Rate Generator
0	14	Parity/Protect Error
11	14	Stall Alarm Trip
2	13	Line Synchronous Clock (60 Hz)
1	11	Teletypewriter Paper-Tape Reader
6	10	Typewriter
7	10	Operator Console
4	9	Drum Memory
5	9	High-Level Multiplexer
8	9	Low-Level Multiplexer
9	0	Spare
10	0	Spare
12	0	Spare
13	0	Spare
14	0	Spare
15	0	Spare

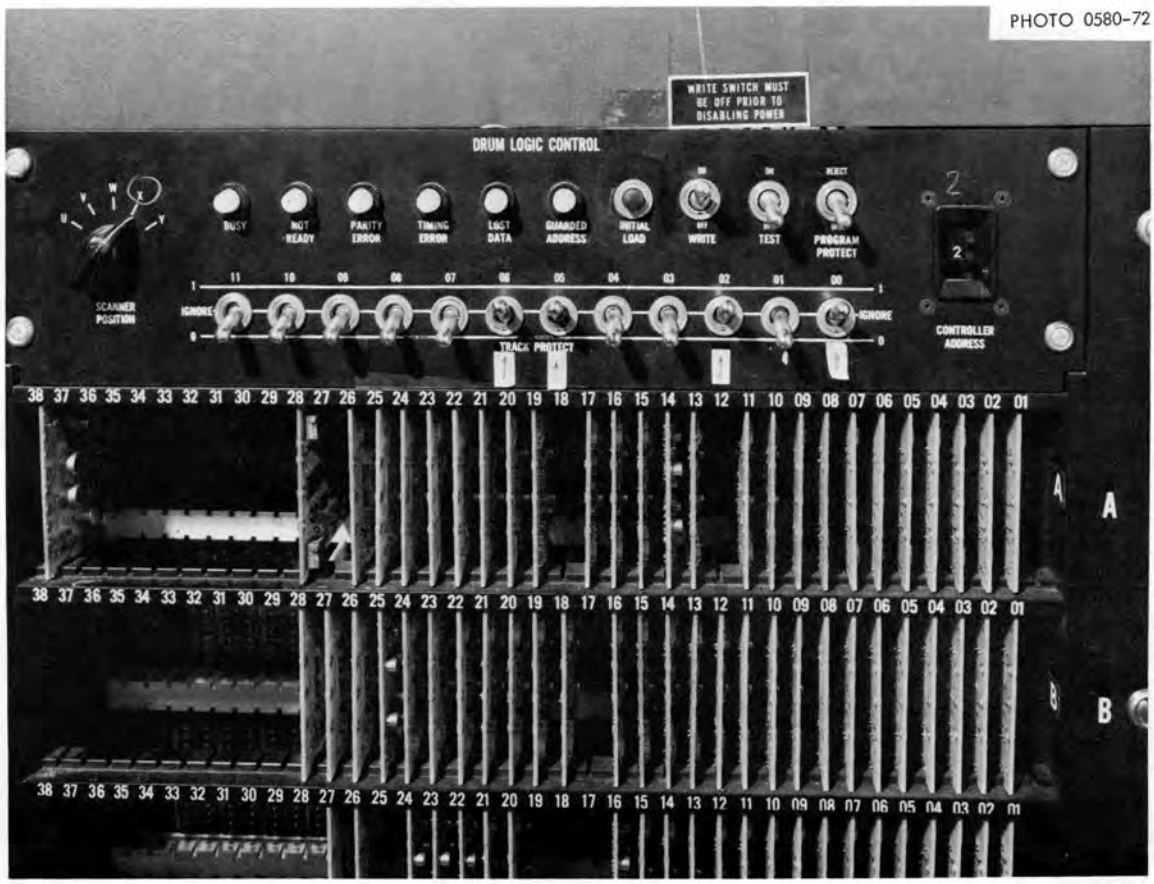


Fig. 7. Drum memory guard switches.

Table 2. Typical examples of operating hierarchies

Program	Priority Level
Off-line, background	0 and 1
Annunciator scanning	8
Digital rod control	7
Reactivity balance	7
High-speed analog input	15
Noise analysis calculations	3

The priority level of a program is an important parameter and is shown at the entry point on all program logic block diagrams.

2.3.2 Monitor

The monitor programs allocate system components to assigned programs and control the sequential operation of the hardware. The monitor programs are reentrant in that they may be interrupted, called by the interrupting program, and then resumed with no loss of continuity. The monitor is composed of the following modules: common interrupt handler, request modules, dispatcher, and input-output drivers (Fig. 9).

A user program request transfers control to the request entry processor which performs several bookkeeping chores and transfers control to an appropriate request processor, e.g., read, write, schedule a program, etc. Control is then returned to the requesting program which has the option of continuing or going to the dispatcher. The dispatcher routine determines which program or request should be executed next by examining the top entries of the scheduler and interrupt stacks. Control is passed to the program with the highest priority.

The interrupt stack is a waiting list formed by partially executed programs that have been interrupted by a program with a higher priority. The scheduler stack is a list of programs constructed by the scheduler request processor in response to user or driver module requests.

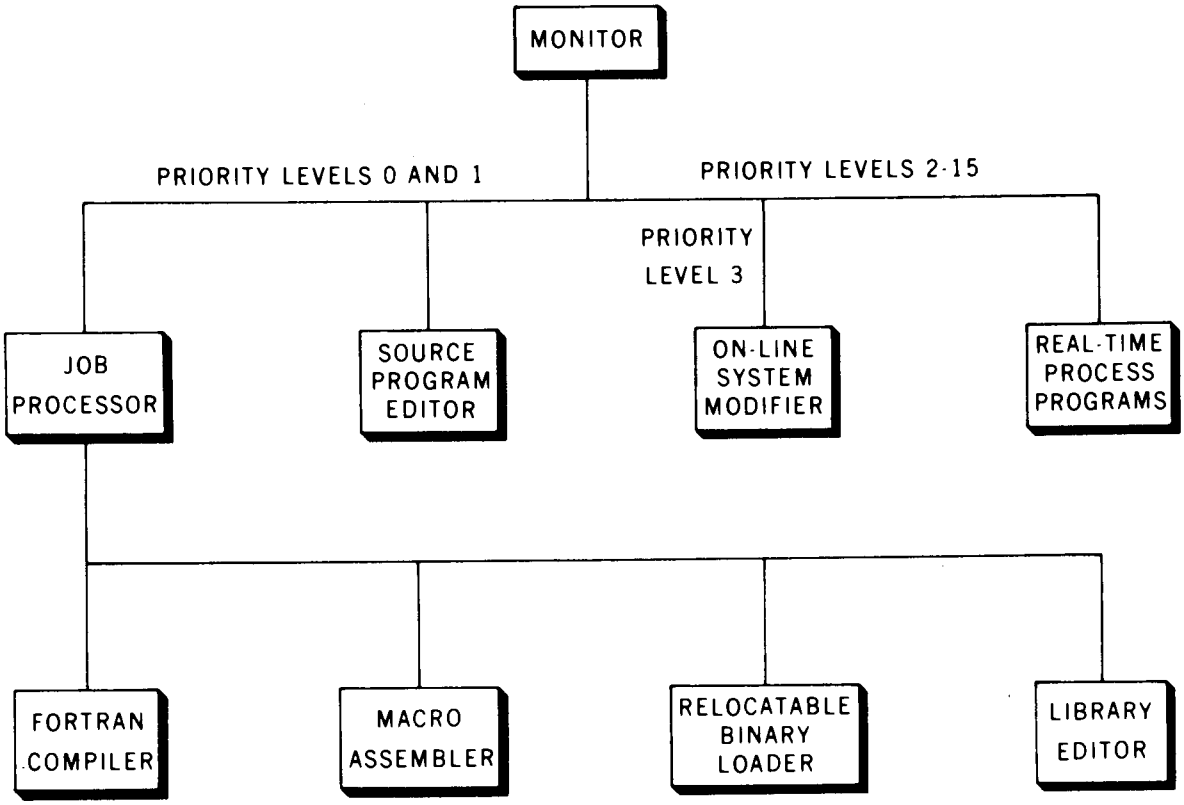


Fig. 8. Executive system modules.

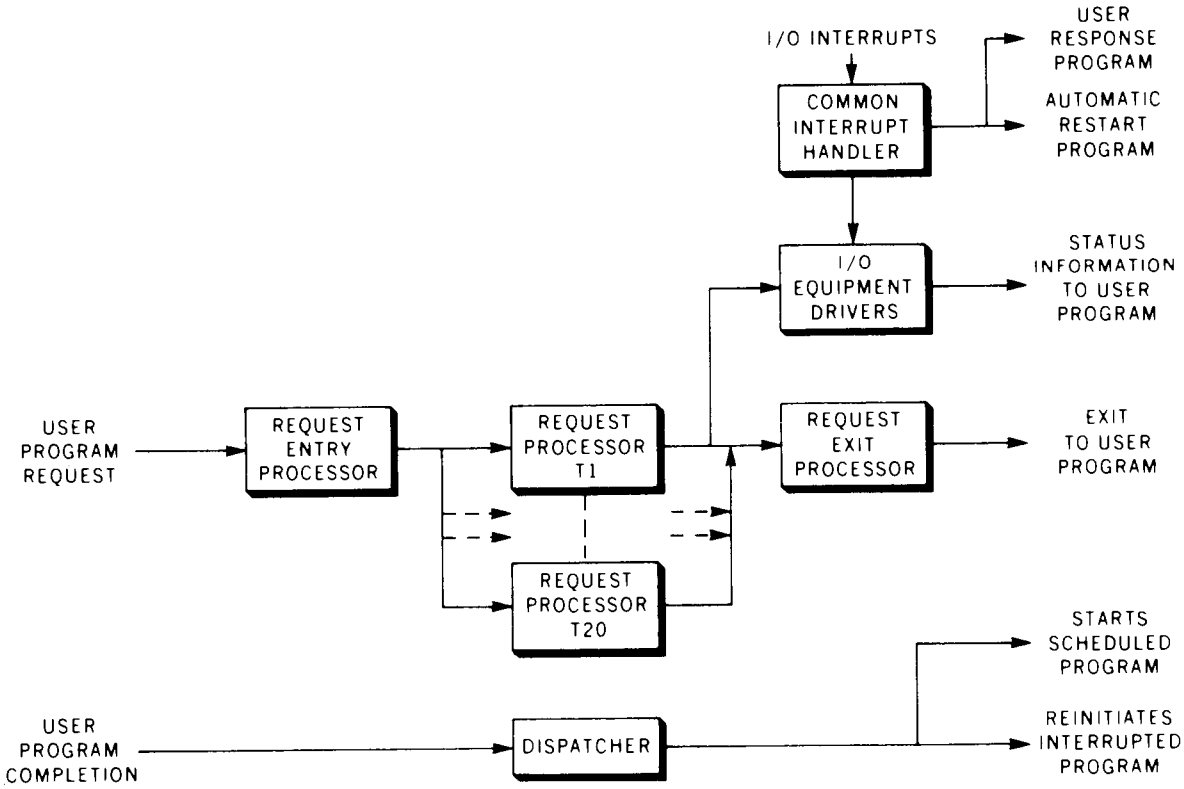


Fig. 9. Executive monitor modules.

2.3.3 Common Interrupt Handler

Hardware interrupts from peripheral equipment, e.g., system clock, paper-tape input-output, mass memory, and internal error interrupts, transfer control to the common interrupt handler which stores information pertinent to the program being interrupted. The information includes the register contents, memory index pointer, priority level prior to the interrupt, return location, and the interrupt line number.

The new priority level, established by setting a value in a hardware mask register, inhibits interrupts by lower-priority equipment or programs.

If the interrupt is generated by a memory parity, protect violation, or a system stall, all interrupts are inhibited, and control is transferred to an automatic restart program which reestablishes system operation. If the interrupt is the result of an input-output device or a manual interrupt from an operator, the interrupt processor will transfer control to the device driver or to a user response routine.

2.3.4 Request Modules

All programs make requests to the monitor by standard calls that include the parameters to execute the request. The monitor saves the contents of the registers at the time of the request and transfers control to a request processor. When the monitor receives requests that use input-output drivers, programs scheduled for execution, or requests to allocate the core, the requests are held in a priority queue for processing. The system request modules are as follows:

Read/write	transfers data between a specified input-output device and core memory,
Fread/Fwrite	format read and write for records with specific formats,
Status	determines the status of an input-output operation,
Indir	allows indirect execution of any request by indicating the location of the desired request,
Schdle	causes a program to be added to a scheduler stack on a priority basis,
Timer	causes a Schdle request after a prescribed time delay,
Space	causes space to be allocated in the core for the operation of mass memory resident programs,

Releas	used to return core storage acquired by a space request to the system,
Exit	used by unprotected programs to signal completion of a program and to return control to the dispatcher,
Core	used to determine the bounds of the unprotected core,
Gtfile	used to access permanent files in the program library.

2.3.5 Dispatcher

Termination of a program causes control to go to the dispatcher. This module then selects the program of highest priority from either the scheduler or interrupt stack and transfers control to that program. If there are equal priorities, the interrupt stack has preference. Entries are placed on the scheduler stack by priority level in response to a user request. Entries are placed on the interrupt stack by the common interrupt handler.

2.3.6 Input-Output Drivers

Each piece of input-output equipment has an operating program associated with it. These are usually called "driver routines" and are composed of two subroutines: the initiator, and the continuator.

The initiator subroutine is entered from a request processor or the continuator. The request queue is checked to find the next request for the device, and, if one exists, the program sets up the equipment and jumps to the continuator.

The continuator subroutine is entered from the initiator or the interrupt handler. If no errors are present, the driver causes the data to be transferred as requested. A typical input-output driver configuration is shown in Fig. 10. When a driver detects a device failure, control is passed to the alternate device handler which will seek an operative alternative device; and, if one is available, it will transfer requests from the failed to the alternative device. If no alternative device is found, the operator is informed by a message on the teletypewriter and corrective action is requested.

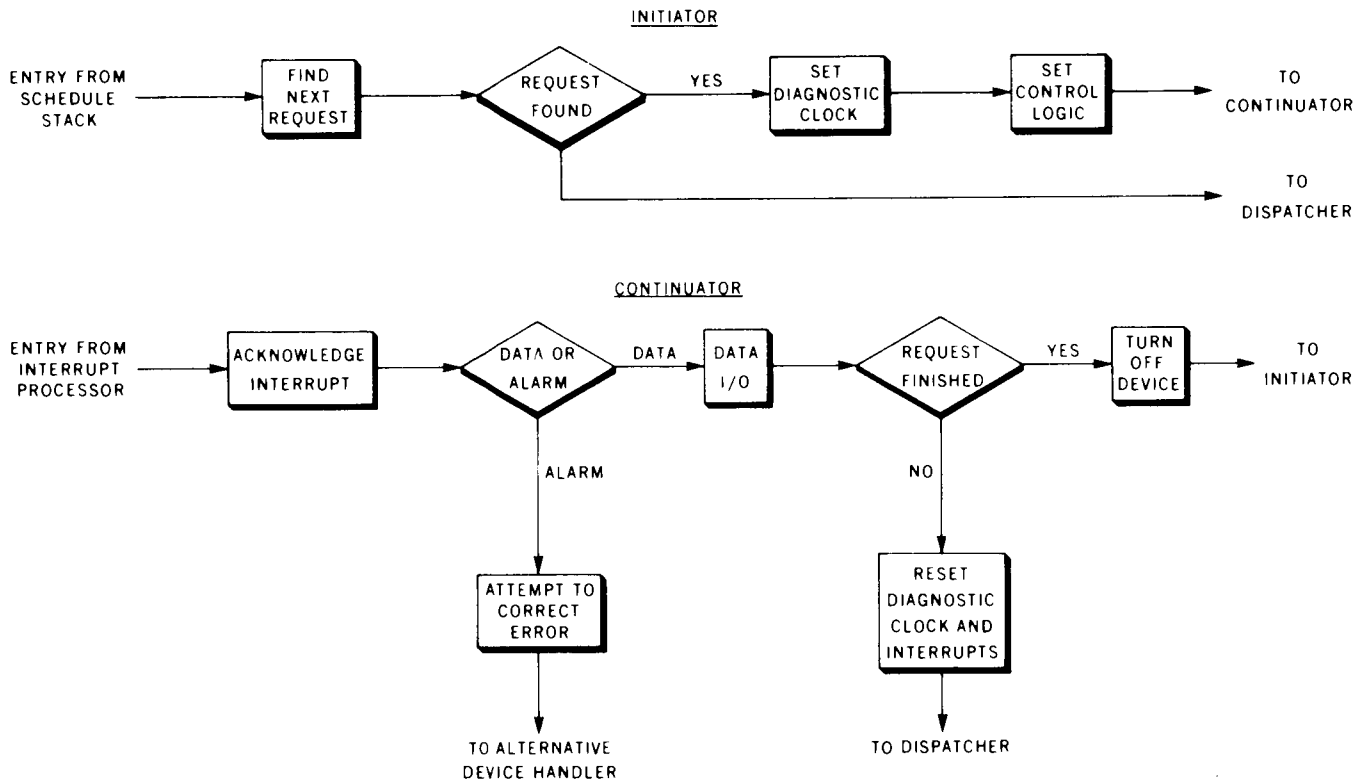


Fig. 10. Basic input-output driver functions.

2.3.7 Job Processor

Job processor routines initiate and control off-line programs in the unprotected core. These programs are initiated by a manual interrupt processor when the operator interrupts the system and types a job processor (JP) statement. Communication with the JP is primarily through typed statements on the teletypewriter, but control may be transferred to another input device, e.g., a paper-tape reader.

All JP statements are preceded by an asterisk and followed by a carriage return. The acceptable statements are as follows:

*U	Directs the JP to read all subsequent control statements from the teletypewriter
*V	Directs the JP to read all subsequent control statements from the binary input device (paper-tape reader)
*P	Causes the relocatable binary loader to be brought into the core
*L, ℓ	Causes the loader to load a relocatable binary program from a specified input device, ℓ
*X	Causes the loaded program to begin execution; the loader detects any needed subroutines from the program library and loads them into core as part of the job prior to execution
*T	Terminate the job
*K, Iℓ, Mℓ, Pℓ, Lℓ, Sℓ	Allows the operator to change the logical units assignment of various physical devices, e.g., change the binary input device from paper tape to drum, where I binary input device M system library unit P binary output unit L system listing device S system scratch area ℓ logical unit (device) number
*Rℓ	Informs the operating system that a failed device has been repaired or restored to service
*Z	Informs the system that the JP is no longer needed and marks the end of job processing
*LIBEDT	Causes the JP to read the library editing program into the core and to initiate execution

*ASSEM	Causes the JP to read the macroassembler into the core and to initiate execution
*FTN	Causes the JP to read the Fortran compiler into the core and to initiate execution.

2.3.8 Relocatable Binary Loader

This module loads relocatable binary programs from peripheral devices or the program library into available unprotected core. The loader itself is stored in the system library on mass memory until called into the core by a *P statement. Input to the loader consists of relocatable binary formatted records up to 60 words in length from any system input peripheral device. This input is single buffered, and, prior to each input operation, the accuracy of the previous input is checked for input-output errors. If an error is found, loading is terminated and the operator is notified through a typed diagnostic.

2.3.9 Assembler and Compiler

The macroassembler provides a symbolic language for writing source programs which it translates into machine instructions. The source programs are written with symbolic machine instructions, pseudoinstructions, and macroinstructions. Macro-definitions may be within the source program or placed in a separate macrolibrary. The basic instruction repertoire is shown in Table 1 of Vol. 2.[7]

A Fortran compiler provides a superset of ASA Fortran and includes many features of ASA Fortran. The source programs may be written entirely in Fortran or in a mixture of Fortran and machine language instructions. Floating-point quantities occupy 32 bits, and the range is $10^{-39} < n < 10^{39}$. The fractional part of a number is represented by 23 bits. Integer quantities are limited to $0 \leq n \leq 2^{15} - 1$.

Source input for both Fortran and assembly may be from paper tape, a teletypewriter, or the mass memory; the binary output is on paper tape or the mass memory. The listing and error diagnostic messages are on the teletypewriter or the Selectric logging typewriter.

2.3.10 Source Program Editor

A source editing and filing system (EDIT) was written at ORNL to allow the programmer to develop and modify source programs utilizing the teletypewriter, paper-tape input-output, and the system mass memory.[8] A source program can be read into the system, under EDIT control, from paper tape or from a previously stored file in the mass memory. New programs can be developed by typing them with the teletypewriter.

The EDIT program is described in detail in Sect. 2.26 of Vol. 2.[7]

2.4 Process Scan Software Description

The process scan programs are defined as those that relate to the tasks of acquiring and handling data from the process and those that are involved in process control or surveillance functions. In most dynamic systems these tasks are repeated on a time base sufficiently short to observe the changes of interest. The program group comprising this series of procedures is termed the "scan loop." The programs that define the HFIR 1-sec scan loop are shown in Fig. 11 where the name of each program and the task each performs are shown in a separate block and the sequence of execution is represented by connecting lines.

A large fraction of the process scan software consists of a group of programs that multiplex the analog process signals through the analog-to-digital converter, convert the resulting digital values into engineering units, and test the values for out-of-limit conditions. These programs, jointly developed by ORNL and CDC personnel, are

SCAN34	Low-level analog scan control
A-D Driver	Low-level analog read and convert
CNVT	Convert millivolts to engineering units
ORALC	Alarm limit comparison
ORMI	Alarm message initiator
HOOKUP	High-level analog scan control
RDPT	High-level analog scan input.

Each of these programs is described in greater detail in the Appendix, Sect. 4.2.

ORNL DWG NO. 72-864

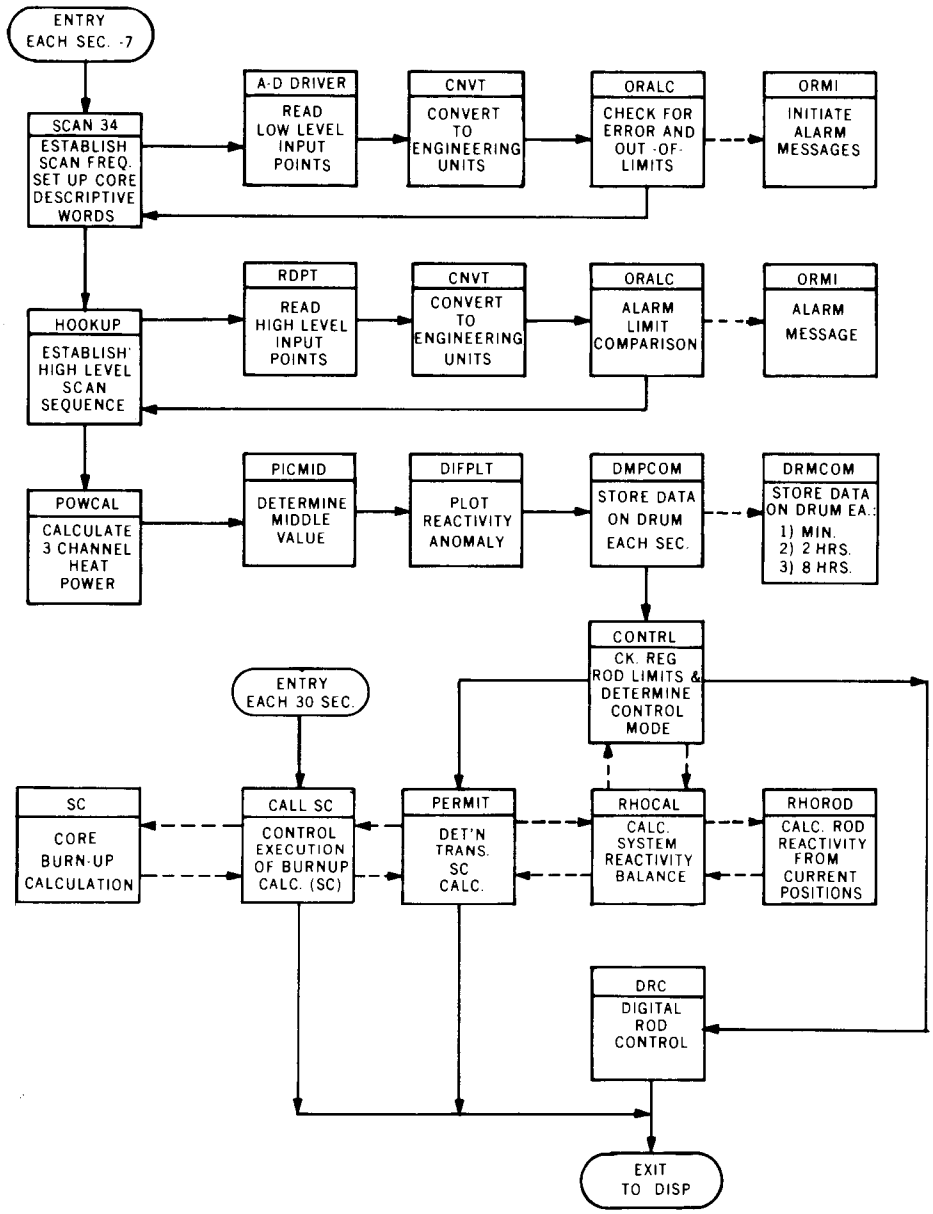


Fig. 11. Basic scan loop modules.

The process variables that are digitized and converted to engineering units are stored for use by other programs in a table called VALU.

The remaining programs in the scan loop, developed at ORNL, are described in detail in Vol. 2 and are shown in diagram form only in the Appendix, Sect. 4.3. Development considerations and a summary of these programs are discussed in the following sections.

2.4.1 POWCAL

Process scan program POWCAL calculates the power level of the reactor and the primary-coolant flow rate. The reactor inlet and exit water temperatures and flow venturi pressure drops from each of the three regulating channel sensors are stored at the beginning of each scan cycle by SCAN34. The primary flow rate in each channel is calculated by the equation $F = K (P)^{0.5}$, where K is the flow venturi calibration constant and P is the venturi pressure drop.

The heat power of each channel is calculated by the equation $HP = (C) (F) (\Delta T)$, where F is the coolant flow rate, ΔT the reactor core differential temperature, and C a conversion constant. Each flow and heat power is stored in the VALU table to enable ready access to the data. The calculated flows and power levels are typed as a part of the process log, and each value can be displayed on the control room console. A logic block diagram of program POWCAL is shown in Fig. 12.

2.4.2 PICMID

PICMID selects the middle value of heat power from the three heat powers calculated by POWCAL and stores it in the VALU table. This median power value is defined as the average reactor power for the burn-up and reactivity calculations. This technique is used in lieu of averaging the three channels, because one regulating channel is occasionally removed from service. Thus, averaging the three channels when one value is zero would cause erroneous heat power calculations. Since two channels must be operational for the reactor to operate, selecting the middle power value ensures a proper power calculation. Figure 13 is a logic block diagram of PICMID.

ORNL DWG NO. 72-843

Fig. 12. Logic block diagram of program POWCAL.

ORNL DWG NO. 72-845

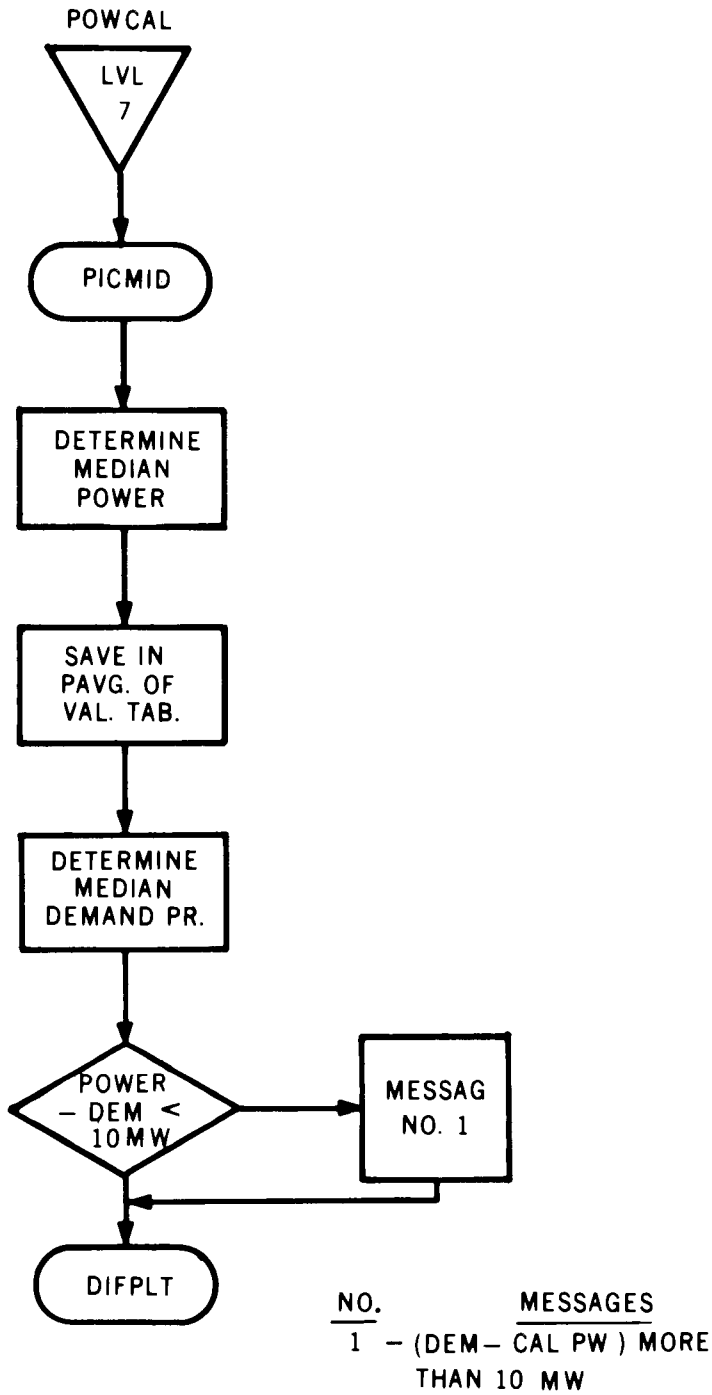


Fig. 13. Logic block diagram of program PICMID.

2.4.3 DIFPLT

The next program in the scan sequence is DIFPLT, which drives a strip-chart trend recorder in the control room. The parameter plotted is Rhodif, the reactivity anomaly, as calculated and defined in program RHOCAL (Sect. 2.4.6). The Rhodif values for the three previous 1-sec scan cycles are averaged to provide a smooth output. The recorder is calibrated for ±1.00 dollar full scale. A logic block diagram of DIFPLT is shown in Fig. 14.

2.4.4 DMPCOM and DRMCOM

DMPCOM stores the current scan data in a mass memory file sufficiently large to hold data from the previous 10 min of operation. The following 12 parameters are stored each second: five control-rod positions, median heat power, coolant temperature at core inlet, neutron flux level, Rhodif, dynamic reactivity, control rod reactivity, and burn-up reactivity.

Another data saving routine, DRMCOM, is scheduled for execution every minute by DMPCOM. The same 12 parameters are stored each minute in a drum file containing all the values from the previous 6 hr of operation. DRMCOM also stores protected common every 2 and 8 hr. The 2-hr records cover the previous 24 hr of operation, and the 8-hr records cover the entire 23-day fuel cycle. The logic block diagrams of DMPCOM and DRMCOM are shown in Figs. 15 and 16.

2.4.5 CONTRL

The primary function of CONTRL is to select the appropriate computer control action from an evaluation of current information. The three basic responses are (1) energize the Auto-Shim permit relay; or (2) transfer control to program DRC for direct rod control; or (3) deenergize all computer controlled relays to prevent any computer control. The first two actions represent the two control modes selectable by the operator, and the third response is initiated automatically during the following error conditions: illogical combination of regulating-rod limit-switch positions, negative reactivity anomaly, positive reactivity anomaly in the absence of a rod drop,

ORNL DWG NO. 72-837

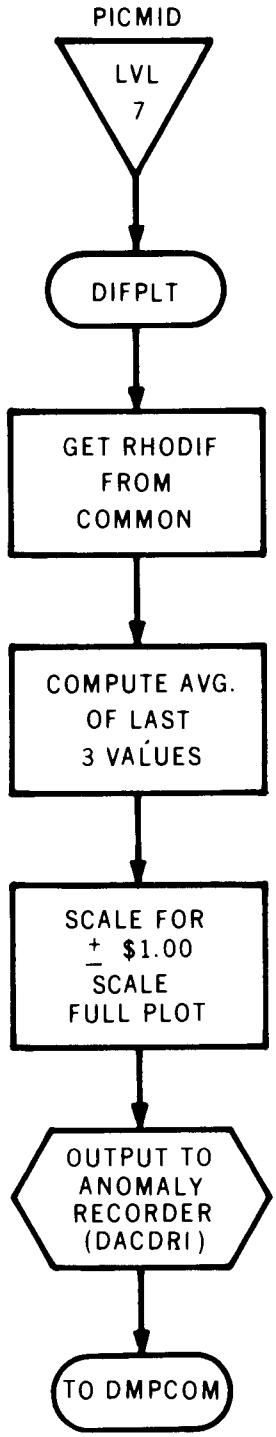


Fig. 14. Logic block diagram of program DIFPLT.

ORNL DWG NO. 72-834

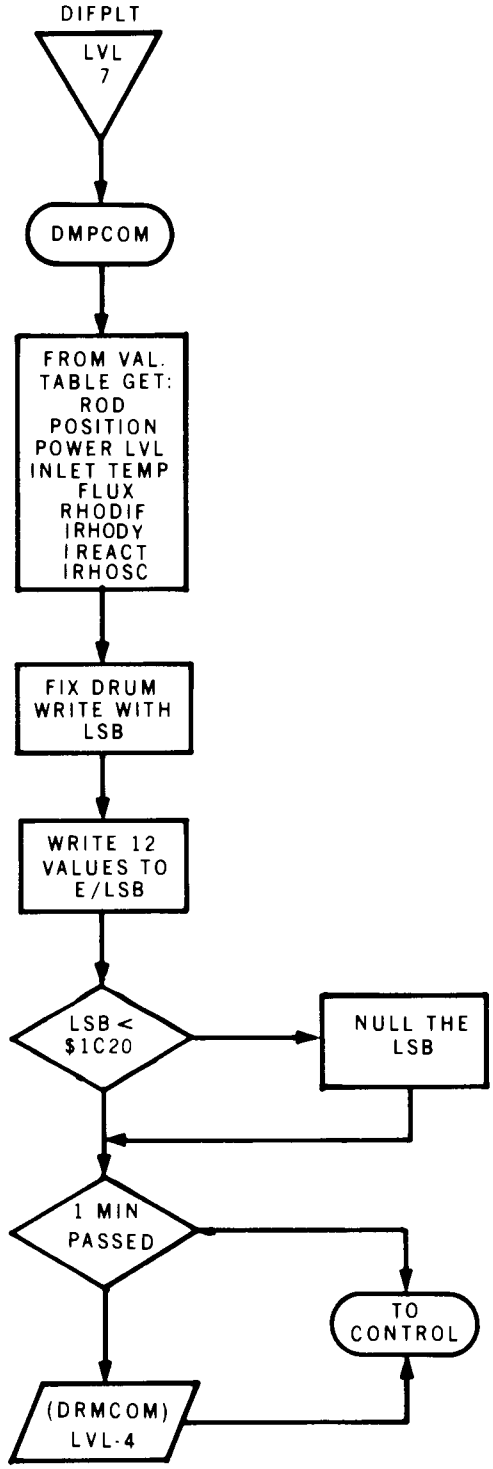


Fig. 15. Logic block diagram of program DMPCOM.

ORNL DWG NO 72-828

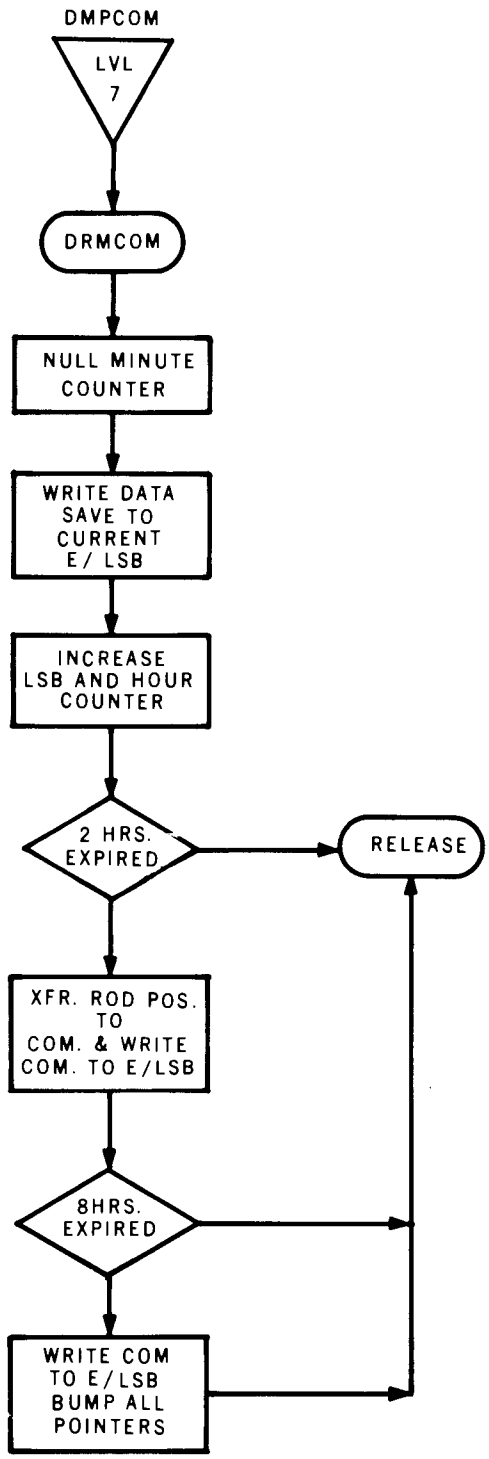


Fig. 16. Logic block diagram of program DRMCOM.

error in the analog-to-digital converter, and dynamic reactivity greater than maximum. The experience prompting development of the error condition logic is discussed in Sect. 3.3.

The reactor operator is provided with a four-position selector switch which is labeled Off, Shim, Rods, and Demand (Fig. 17). When the switch is manipulated, program CONTRL detects the change in position via a digital input signal derived from the switch positions. When the switch is in the Shim position, the program will, in the absence of errors, actuate the Auto-Shim permit relay to cause automatic shimming of all control rods whenever the regulating rod is above the Automatic Shim Withdrawal (ASW) limit switch position. Whenever the switch is in any other position, this function is bypassed. When the Rods position is used, the same error tests must be completed before control is transferred to the digital rod control program DRC. Since the selector switch (block switch) also controls the relays that connect the computer signals to the reactor control circuits, the logic in CONTRL will not allow the control program to be turned off unless the switch position is also off. A secondary entry point, FLAGER, controls the DRC on-off lamp located on the computer console in control room. This loop is entered whenever the console button is actuated, and as shown in the logic diagram of CONTRL in Fig. 18, the DRC lamp will not be turned off if the Rods position is selected.

To update the reactivity balance, the CONTRL program uses RHOCAL as a subroutine either directly or indirectly via PERMIT. The reactivity balance calculation is performed with a series of programs operating as subroutines, described as follows.

2.4.6 RHOCAL

RHOCAL calculates the current reactivity anomaly (Rhodif) which is the signed value of the excess reactivity deviation from the predicted value. Rhodif is defined as ρ_a (reactivity anomaly) $\equiv \rho_{sc}$ (cal. reactivity change) $- \rho_r$ (meas. reactivity change). The reactivity parameters are time dependent and are referenced from the beginning of the fuel cycle. Therefore, ρ_{sc} is the current value of the calculated change in excess reactivity since the start of the fuel cycle, i.e., the integrated burnup:

PHOTO 0576-72

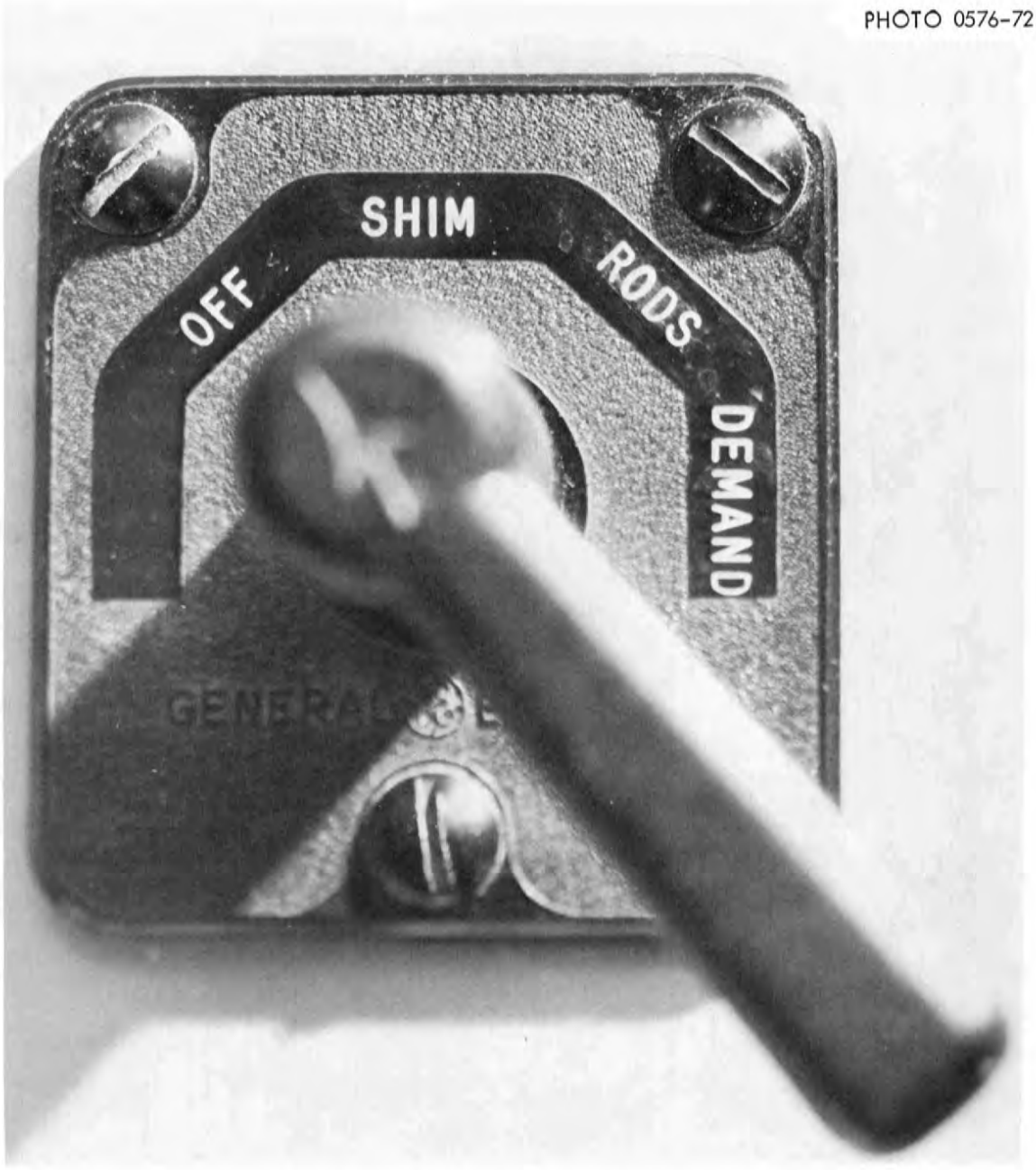


Fig. 17. Operator mode-select switch.

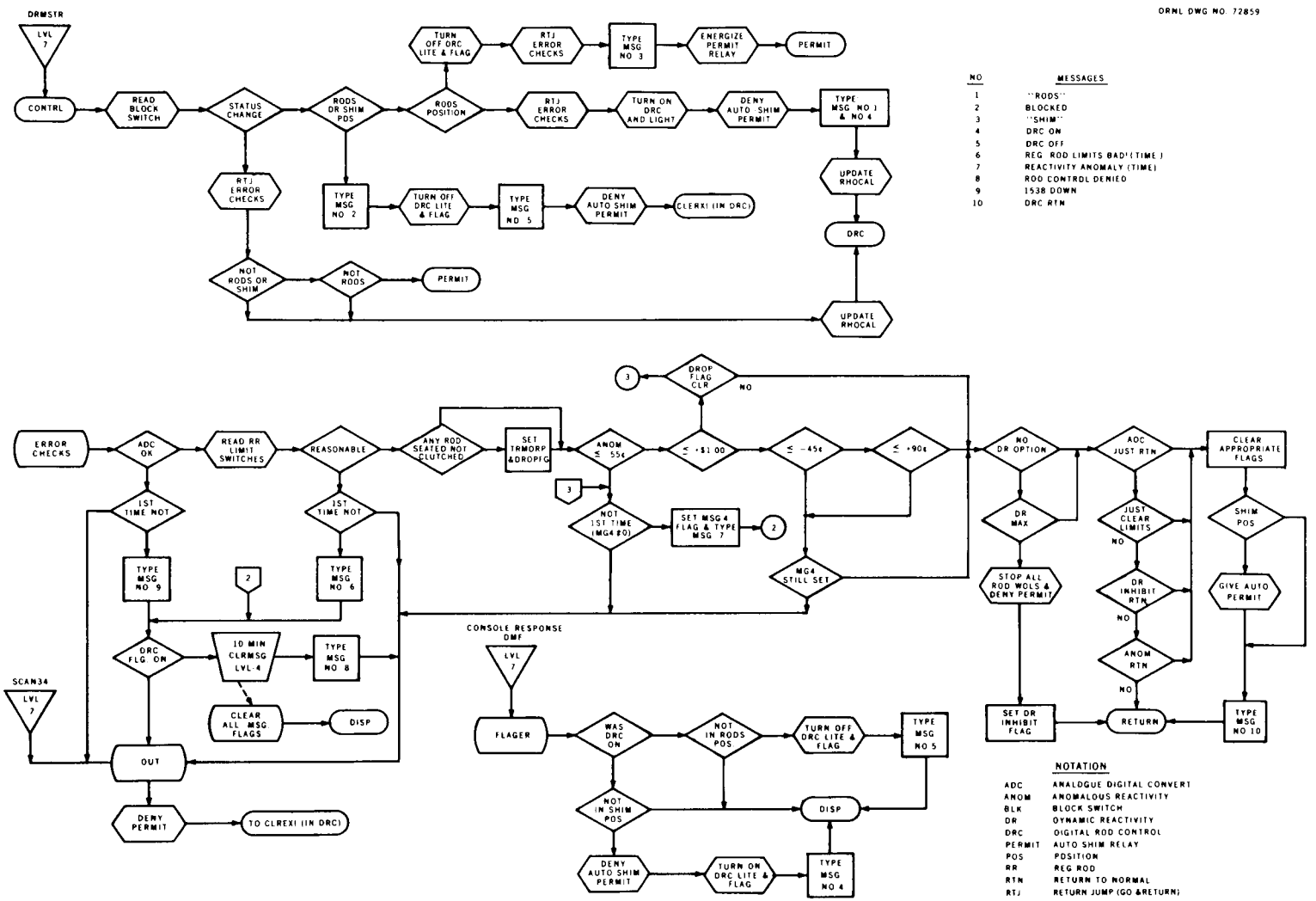


Fig. 18. Logic block diagram of program CONTRL.

$$\rho_{sc}(t) = \int_0^t \left(\frac{\Delta\rho}{\Delta T}\right)_{burnup} dt.$$

The measured excess reactivity (ρ_r) is determined from the current position of the control rods and the rod-worth algorithm as explained below.

The above equations assume that the reactivity measurements are made while the reactor is critical. A kinetic reactivity correction is added when the inverse-kinetics program[9] DR is operating.

RHOCAL uses a flag in initial common to detect the first operation of the program at the beginning of each fuel cycle. During the first scan cycle, the reactivity measured from RHOROD is defined as the initial core excess reactivity (ρ_o), and the position of rod 5 is defined as the initial symmetric critical value. Both values are stored in protected common for use throughout the fuel cycle. The logic block diagram of RHOCAL is shown in Fig. 19. The parameters for the anomaly calculation are generated by the following programs.

2.4.7 RHOROD

RHOROD is a subroutine used by RHOCAL to compute the combined reactivity worth of the five control rods, based on the rod positions stored in the VALU table by the scan programs discussed previously. Therefore, when critical, this worth is the measured total core excess reactivity. The value is returned to the calling program in the A-Register, with units of cents.

The integrated rod-worth at any time τ is usually obtained by integrating the differential rod-worth curve from the current position $X(\tau)$ to the fully withdrawn position. For five rods the total value is given by

$$\rho_r(\tau) \equiv \sum_{i=1}^{5} \int_{X(\tau)}^{X\,max} \left(\frac{\Delta\rho}{\Delta X}\right)_i dX.$$

By subtracting the reactivity value at any time (τ) from the value at the beginning of the cycle ($\tau = 0$), the measured change in reactivity as indicated by the rod positions is calculated from

ORNL DWG NO. 72-853

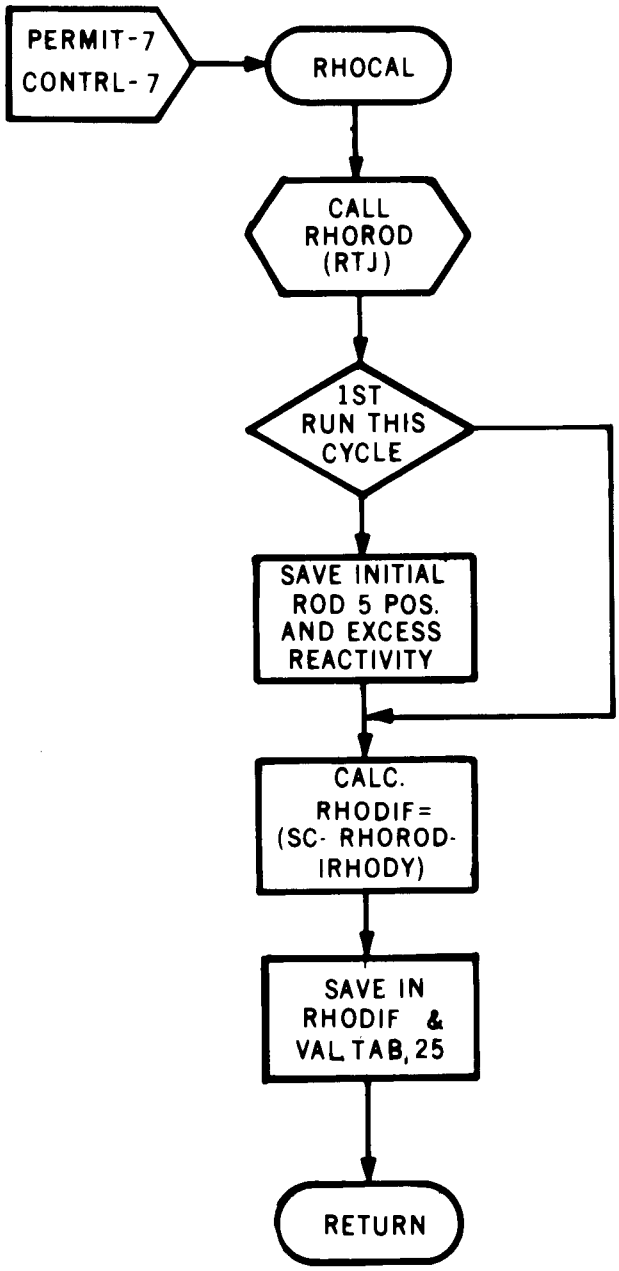


Fig. 19. Logic block diagram of program RHOCAL.

$$\rho_r(\tau) = \sum_{i=1}^{5} \int_{X\,crit}^{X\,max} \left(\frac{\Delta\rho}{\Delta X}\right)_i dX - \sum_{i=1}^{5} \int_{X(\tau)}^{X\,max} \left(\frac{\Delta\rho}{\Delta X}\right)_i dX;$$

where X crit is the position of the rods at the start of the cycle. The first term is ρ_o, the initial core excess reactivity discussed under RHOCAL (Sect. 2.4.6). To reduce the computation time, the differential rod worth curves were prefitted to a fifth-order polynomial using a least-squares method, and the polynomial was integrated to give an analytic function for the integral rod worth as a function of position. Straight-line segments represent the lower portions of the integral worth curves. The worths of rods 1-4 are about equal, and the curves are shown with the describing equations in Fig. 20. The measured change in reactivity is calculated by

$$\rho_r = \rho_o - f(x),$$

where f(x) is the integrated reactivity calculated from the appropriate sixth-order polynomial. The logic block diagram of RHOROD is shown in Fig. 21.

2.4.8 CALLSC and SC

SC is the burnup calculation algorithm based on a one-region, one-energy model of the HFIR.[11] The calculation consists of first determining the core averaged concentration of each of the isotopes that contribute appreciably to the reactivity balance and then assigning a reactivity change to each increase or decrease in concentration. The equations to determine the core averaged concentrations are shown in Table 3.

Table 3. Equations that describe core averaged concentrations

$$\frac{d\,{}^{235}U}{dt} = -\phi\,{}^{235}U_o\sigma_U ,$$

$$\frac{d\,{}^{10}B}{dt} = -\phi\,{}^{10}B\sigma_B ,$$

$$\frac{d\,{}^{135}I}{dt} = \phi\gamma_I\sigma_f\,{}^{235}U - \lambda_I\,{}^{135}I ,$$

$$\frac{d\,{}^{135}Xe}{dt} = \phi\gamma_{Xe}\sigma_f\,{}^{235}U + \lambda_I\,{}^{135}I - (\lambda_{Xe} + \sigma_{Xe}\phi)\,{}^{135}Xe ,$$

$$\frac{d\,{}^{147}Nd}{dt} = \phi\gamma_{Nd}\sigma_f\,{}^{235}U - \lambda_{Nd}\,{}^{147}Nd ,$$

$$\frac{d\,{}^{147}Pm}{dt} = \lambda_{Nd}\,{}^{147}Nd - \sigma_{Pm}\phi\,{}^{147}Pm ,$$

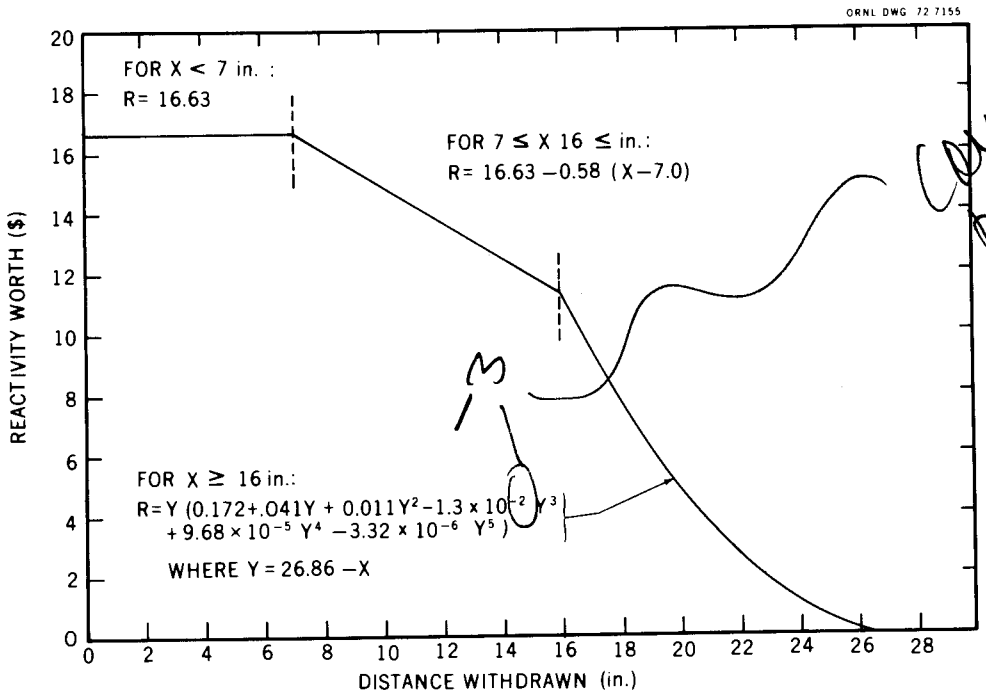
$$\frac{d\,{}^{148m}Pm}{dt} = .47\,\phi\sigma_{Pm}\,{}^{147}Pm - \left[\lambda_{148mPm} + \sigma_{148mPm}\phi\right]{}^{148m}Pm ,$$

$$\frac{d\,{}^{148}Pm}{dt} = .53\phi\sigma_{Pm}\,{}^{147}Pm - \left[\lambda_{148Pm} + \sigma_{148Pm}\phi\right]{}^{148}Pm ,$$

$$\frac{d\,{}^{149}Pm}{dt} = \phi\gamma_{Pm}\sigma_f\,{}^{235}U + \phi\left[\sigma\,{}^{148m}Pm + \sigma\,{}^{148}Pm\right] - \lambda_{Pm}\,{}^{149}Pm ,$$

$$\frac{d\,{}^{149}Sm}{dt} = \lambda_{Pm}\,{}^{149}Pm - \sigma_{Sm}\phi\,{}^{149}Sm ,$$

$$\frac{d\,FP}{dt} = \phi\gamma_{FP}\sigma_f\,{}^{235}U .$$



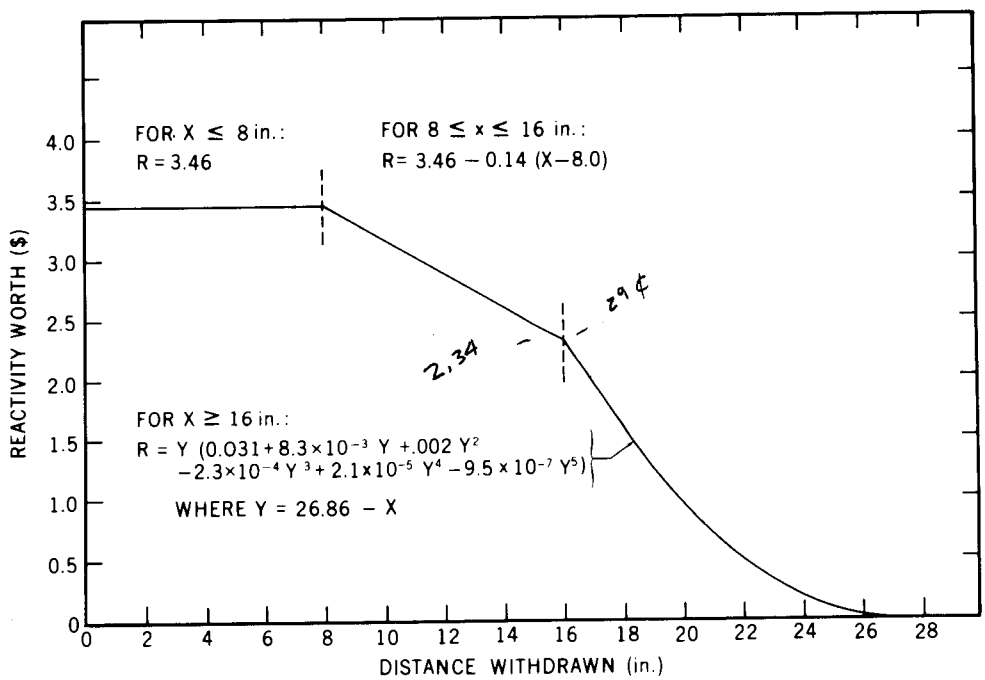


Fig. 20. Total rod reactivity worth vs position.

ORNL DWG NO. 72-802

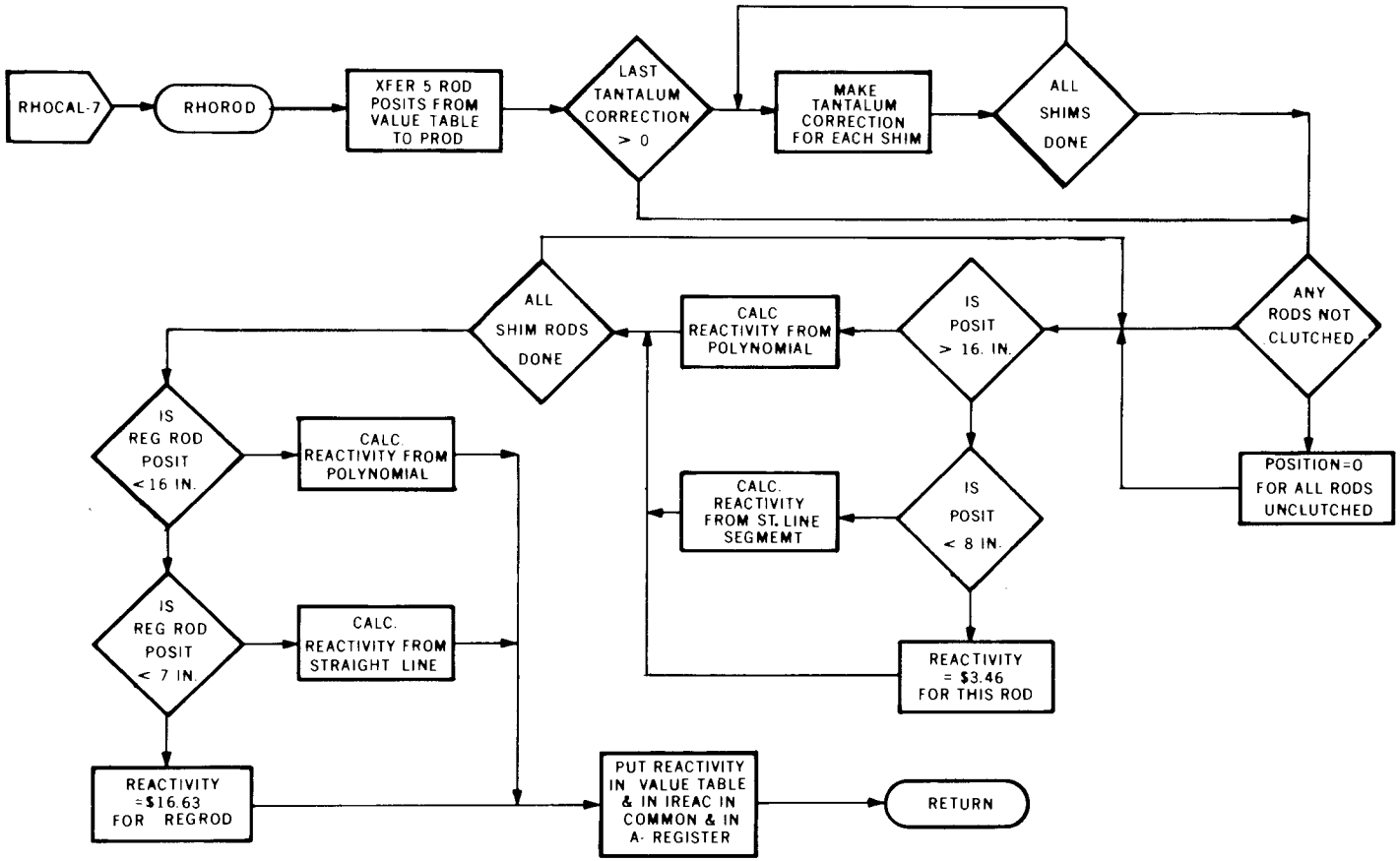


Fig. 21. Logic block diagram of program RHOROD.

The isotopic symbol represents a core averaged concentration, FP is a generalized fission product term, γ is the fission yield, σ_f is the fission cross section, λ is the decay constant, and σ is the neutron absorption cross section. The core average neutron flux ϕ is determined from the heat power (PICMID) and varies inversely with the uranium concentration. These equations are solved by a first-order Heun[12] method, and the calculated change in reactivity is then determined by the following equations:

$$\Delta\rho_{^{235}U} = K_U \frac{dU}{dt} C ,$$

$$\Delta\rho_{^{10}B} = K_B \frac{d^{10}B}{dt} C ,$$

$$\Delta\rho_{^{135}Xe} = K_{Xe} \frac{d^{135}Xe}{dt} C ,$$

$$\Delta\rho_{^{149}Sm} = K_{Sm} \frac{d^{149}Sm}{dt} C ,$$

$$\Delta\rho_{FP} = K_{FP} \frac{d\,FP}{dt} C ,$$

$$\Delta\rho_{total} = \Delta\rho_{^{235}U} + \Delta\rho_{^{10}B} + \Delta\rho_{^{135}Xe} + \Delta\rho_{^{149}Sm} + \Delta\rho_{FP} ,$$

where $C = \Delta t/\nu\sigma_f{}^{235}U$, K is the fitted reactivity parameter, and ν is the number of neutrons per fission. The behavior of each term contributing to changes in the reactivity is shown in Figs. 22-24.

Since the basic form of the burn-up algorithm had been developed before the computer was purchased, it was convenient to include a copy of the SC program in the bid request and require the prospective bidder to run it on his proposed machine. In this way the otherwise difficult questions of execution rate, compiler type, floating-point truncation, and accuracy were more easily answered.

Development of the program on a CDC-1604 indicated that a repetition rate of 2/min would provide good accuracy during periods of rapid changing reactivity. The CALLSC program is the calling routine for program SC.

72-5864

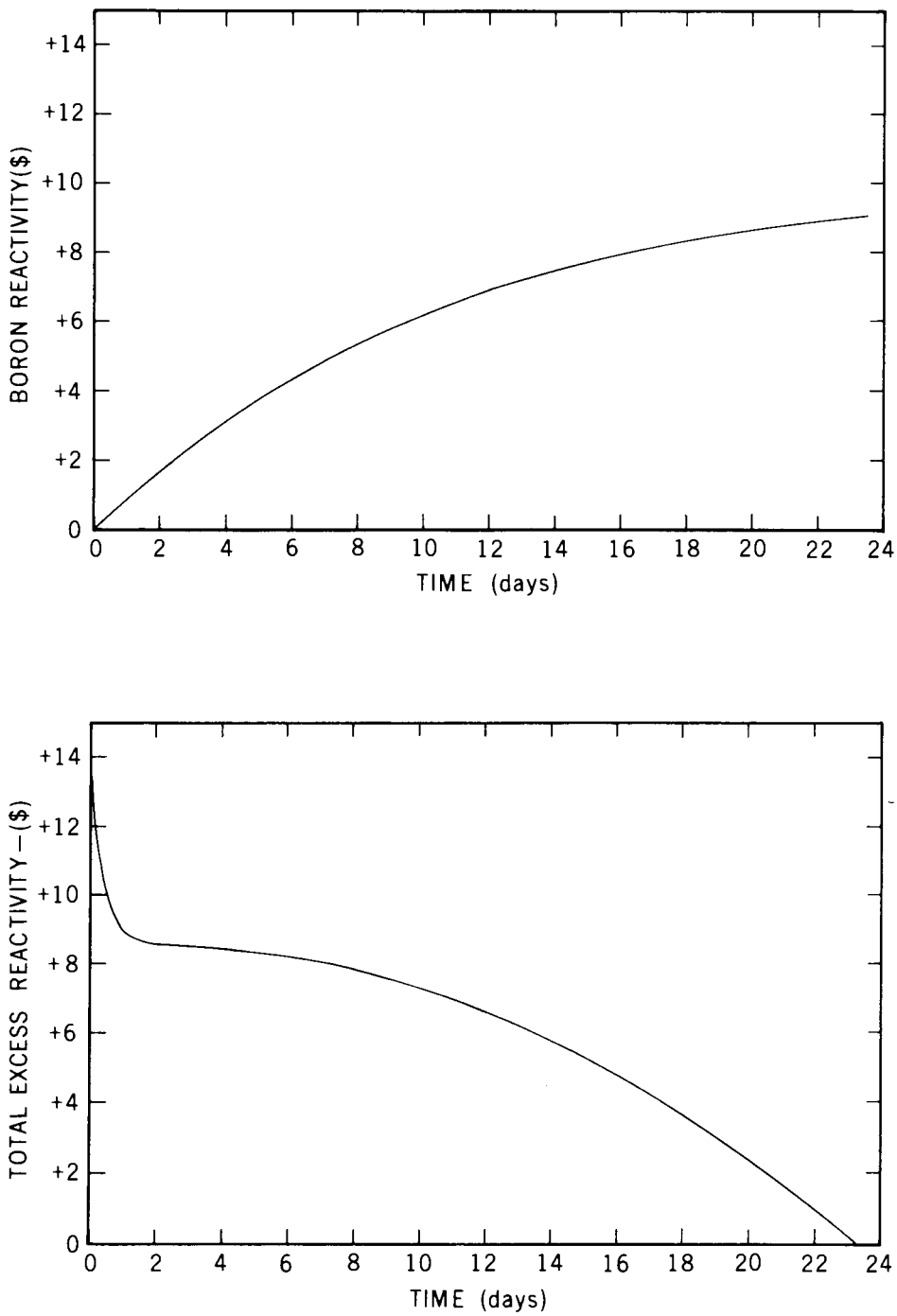


Fig. 22. Total excess and boron reactivity vs time in cycle.

72 5866

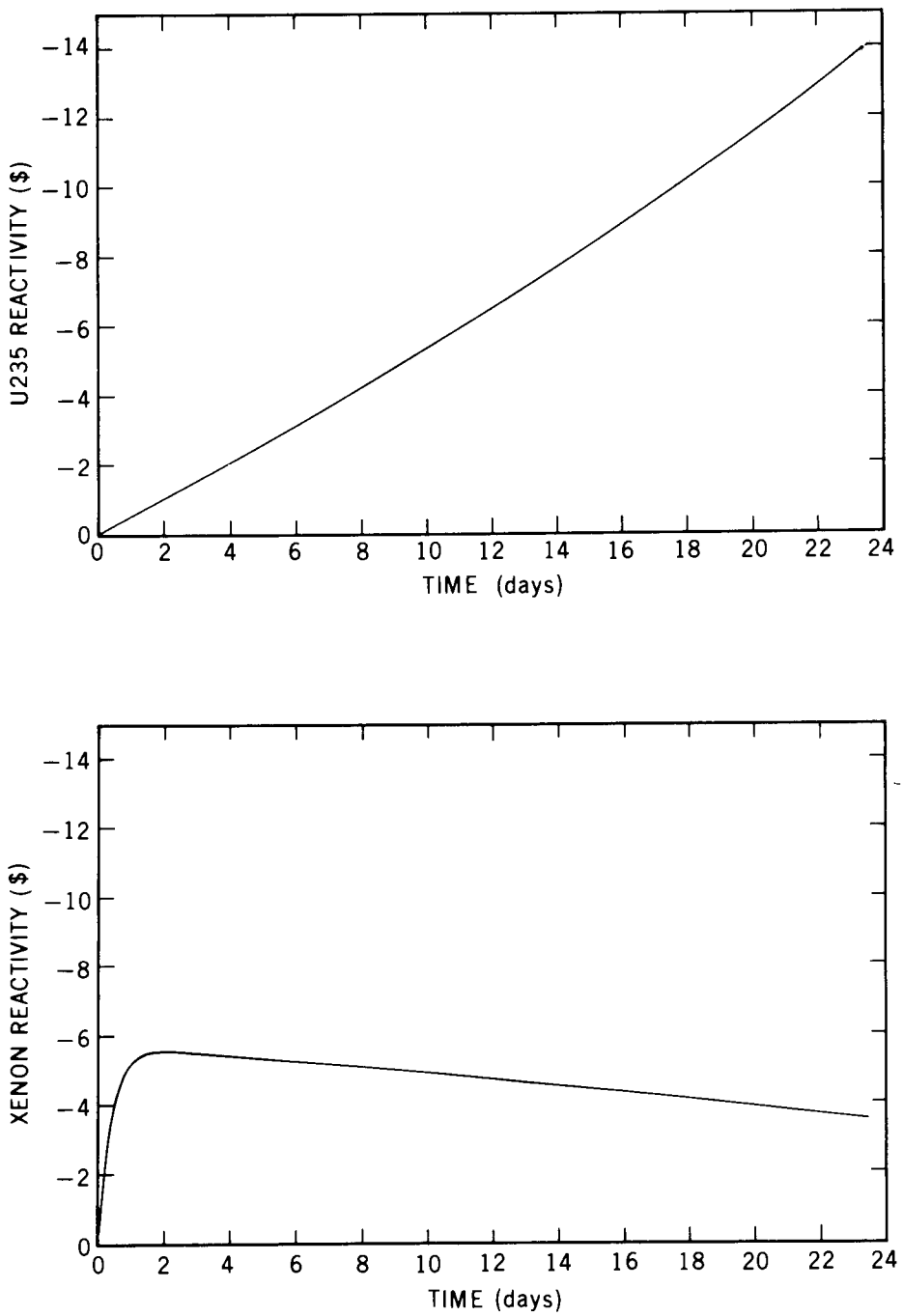


Fig. 23. ^{135}Xe and ^{235}U reactivity vs time in cycle.

72-5863

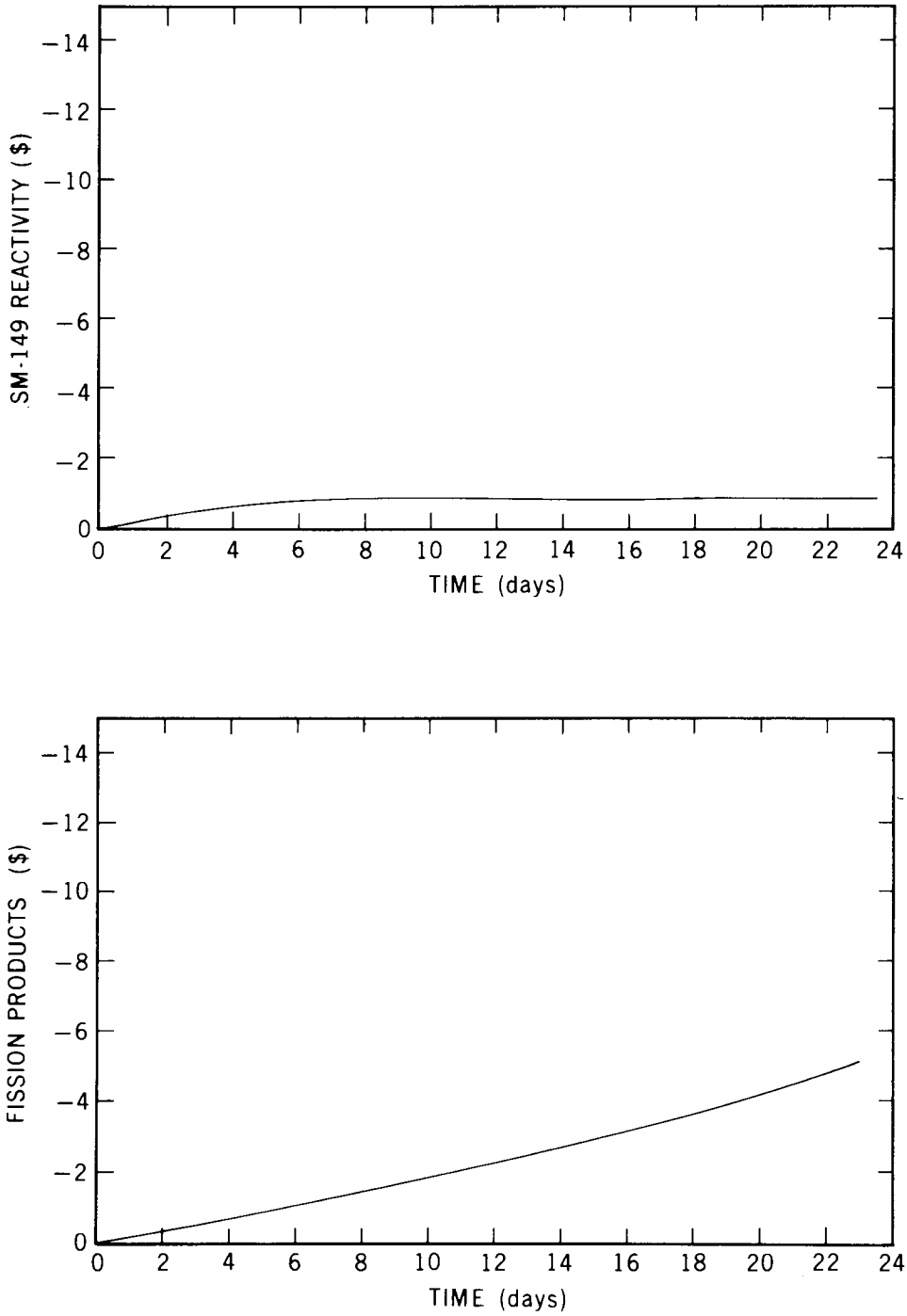


Fig. 24. ^{149}Sm and fission product reactivity vs time in cycle.

CALLSC generates the regular 30-sec timer call to provide for the routine execution of SC in addition to the transient entries. During conditions of system update, wherein the computer has been off for some period, the GIDUP program will repeatedly execute SC via CALLSC until the "lost" time has been accounted for. The PERMIT program will execute SC via CALLSC when a reactor power transient of more than 1.5 MW/sec is initiated or terminated.

2.4.9 DRC

The operating restrictions for the HFIR require that the power level be automatically reduced to 10 MW whenever any two rod positions differ more than 2.0 in. (6% of full travel). During steady-state operation, minor asymmetry can easily be corrected manually; however, when there is a gross asymmetric condition, such as a dropped rod, the operator must rapidly recover the symmetric pattern and return to full power before excessive Xe growth occurs. For a dropped rod, the operator will usually correct for the negative reactivity inserted by the dropped rod and maintain criticality at 10 MW by withdrawing the other shim rods. This state will therefore give rise to automatic shim-rod insertions when the operator withdraws the dropped rod to reestablish symmetry. Since the analog auto-shim insertions take priority over the manual withdrawal, all rods are inserted and the distance between the highest and the lowest rods is not reduced during the insert shimming actions.

To increase the speed and efficiency of reestablishing rod symmetry, the digital control algorithm (DRC) was written to interact directly with the control rods. The basic function of this program is to identify the highest and lowest shim rods and to actuate relays that will insert the high rod while simultaneously withdrawing the low rod. The program will recognize the priority of the regulating rod in maintaining criticality and the requirements for reactivity balance while digitally moving the shim rods.

DRC is initiated by operator request or program RODJOG. The operator can set the DRC On flag by moving the reactor console switch to the Rods position or by pressing the computer console button labeled DRC. If no error conditions are detected by CONTRL, the DRC algorithm will control the rods as necessary to meet the following criteria:

1. maintain all positions to within 0.05 in. of each other,
2. maintain the regulating rod near the intermediate limit position,
3. not withdraw rod 5 if it is withdrawn past the symmetric critical position or the position of any other rod,
4. not withdraw any rods if the dynamic reactivity (K_{eff}) exceeds the set point.

The first two requirements provide the necessary features for automatic shim control during routine operation, in addition to a rod-drop recovery. For example, during steady full-power operation, the regulating rod will gradually be withdrawn by the analog servo to correct for burn up. When the withdrawal causes more than 0.050 in. of symmetry, the computer will withdraw the lowest rod and thus cause the regulating rod to insert. This process will continue until all the fuel has been consumed and the rods are at the withdrawal limit. The operational experience with the DRC algorithm is discussed in Sect. 3.3. A logic block diagram of DRC is shown in Fig. 25.

As shown in Fig. 11, DRC is the last program in the 1-sec scan loop; therefore, DRC exits to the dispatcher.

2.5 Operational Surveillance Software Description

Included in this category are the programs developed at ORNL for the primary purpose of supplying the reactor operator with better information about the plant. Generally, programs in this category are executed only on request and are assigned to a console function number or pushbutton. The programs and demand function numbers are listed in Table 4. Logic block diagrams are in the Appendix, Sect. 4, and a description of each program is given in Vol. 2.[7]

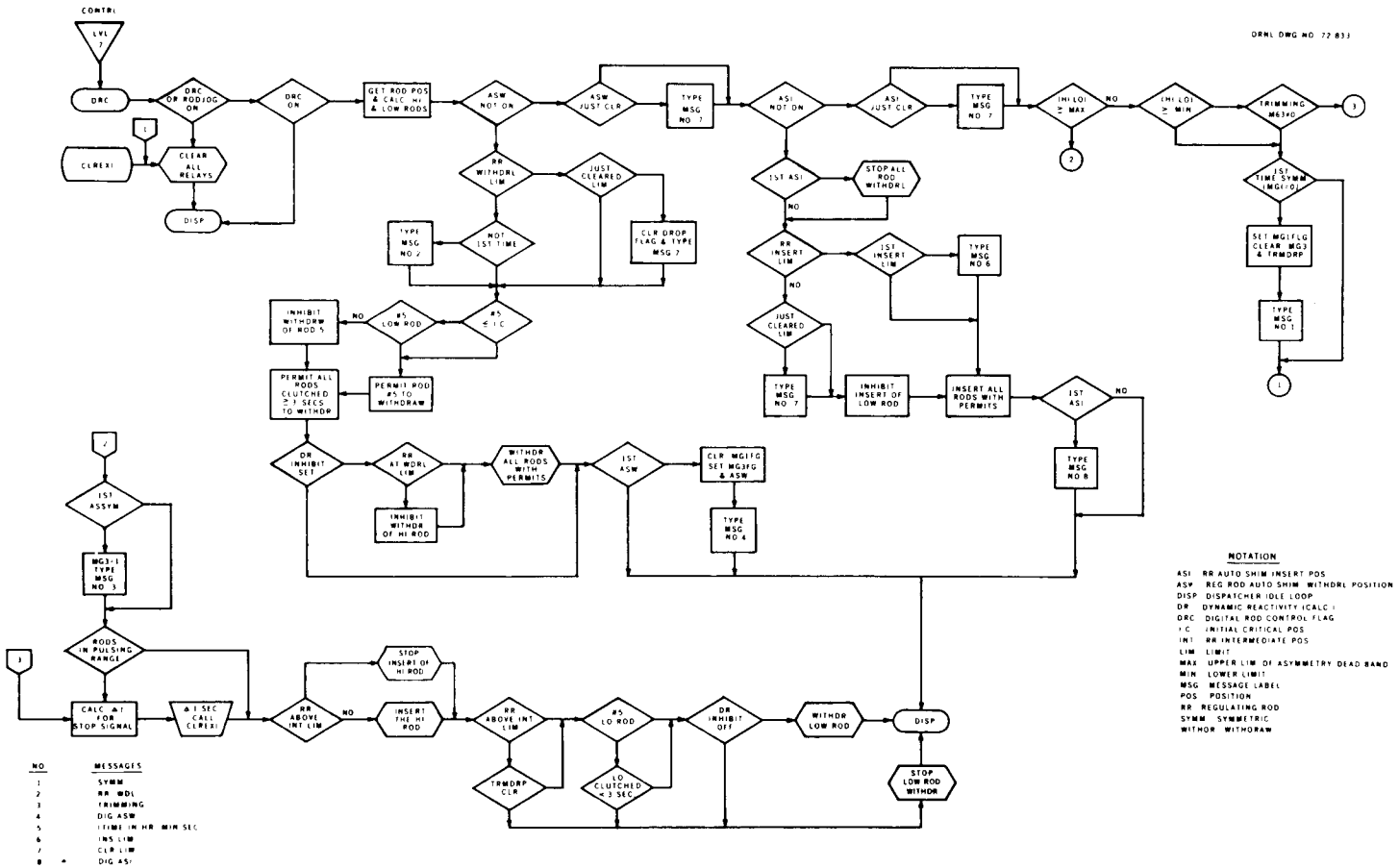


Fig. 25. Logic block diagram of program DRC.

Table 4. Operational surveillance software

Name	Function No.	Task Summary
ANNBLK	--	Contains message blocks on drum for ANNUNC and also tests for 2/3 safety trips in order to schedule GEOF.
ANNUNC	--	Scans all safety channel annunciators each 200 msec and lists sequentially the name and time for all alarms.
AVGDAT	12	Accumulates and lists the average values of primary flow rate, core inlet and exit temperature, and power level for each servo channel averaged over the requested time.
BULKRY	04	Uses MUXBUF to digitize and store fluctuating signal data on drum which is reduced to power spectral density vs frequency. The results may be subsequently listed and/or plotted using PSDIO (function 07).
DATPLT	13	Provides repeating zero and full-scale values to the x-y plotter for calibration.
DIGCLK	NA	Automatically calculates and displays the current time of day in the digital display window on the console each minute.
DPLAY	00	Displays selected reactor parameter from the VALU table (flow, temperature, power, etc.) in the digital display window on the console.
DR		Calculates the dynamic reactivity by inverse kinetics from neutron flux measurement for display on x-y plotter or digital display window.
DRMXFR	16	Saves updated reactor data by transferring the current values to a permanent save location on the drum.
DWNTIM	10	Extrapolates and lists the data and time for the end of the current reactor fuel cycle on the basis of integrated power.

Table 4. Operational surveillance software (continued)

Name	Function No.	Task Summary
FLTIME	08	Measures and lists the release, response, and flight times for each of the shim safety rods.
GADAT	NA	Calculates and displays the average reactor power using 30-sec average values of primary flow rate and core inlet and exit temperatures.
GEOF	NA	Saves current 1-sec data for last 10 min and 2-min data for last 2 hr by transferring it to a permanent save location and punching it on paper tape. Save is automatically initiated 8 min after an annunciator safety trip of two or more channels is detected by ANNBLK.
GETDAT	NA	Accumulates the 30 values of primary flow rate and core inlet and outlet temperatures used by GADAT for the 30-sec average power calculation.
LOG	01	Lists current values of all reactor parameters.
MUXBUF	04	Digitizes and stores fluctuating signal data using the 1572 high-speed timer for subsequent use by BULKRY.
PLOT	17	Plots on x-y recorder values digitized and stored on drum $A/0.
PLTCAL	11	Provides values to x-y recorder for calibration for use by use by PSDIO (function 07).
PLTCOM	14	Plots values on x-y recorder of reactor parameters currently in the updated data save area on drum $E.
	15	Plots past values on x-y recorder of reactor parameters saved on paper tape or in the fixed save area on drum $C.
PROJCT	02	Calculates the projected time to poison at the current power level or following a change in power, based on the burnup model of the core.
	03	Same as function 02 except that if poison occurs the time to restart (decay of Xe) is calculated.

Table 4. Operational surveillance software (continued)

Name	Function No.	Task Summary
	26	Calculates time to poison and time to restart for a new core for a specified power profile containing up to five power levels.
	27	Same as function 26 for the current core.
PSDIO	07	Lists and plots on x-y recorder the results of the power spectral density calculation of BULKRY (function 04).
REDBLK	18	Selects black or red ribbon on Selectric typewriter.
RODJOG		Automatically determines the differential rod worth for each of the five control rods in the HFIR by movement of the rods using rod control program DRC.
SCNFLO	(Hydraulic Scan Button)	Activates stepping switch to drive pneumatic multiplexer and calculates from the pneumatic data the flow rate in the beam tubes and reflector regions. Alarm conditions are listed.
SETOD	06	Sets the computer clock time of day.
SWENTR	(Average Heat Power Button)	Activates the 30-sec average heat power calculation using GETDAT and GADAT.

2.6 System Support Software Description

Programs designed to expand the capabilities of the executive system and non-process related function are classified as system support software. All these programs were written at ORNL to increase the efficiency of the HFIR computer system and to increase the total system availability. The EDIT and EDFILE programs greatly increase the rate of preparing and debugging new programs. Computer downtime due to intermittent failures is reduced by the automatic restart programs, PSYCHO and GIDUP, and downtime for system rebuild is minimized by program PATCH. The development and operating philosophy behind these programs are discussed in Sect. 3, and a

description of each program is given in Vol. 2.[7] Table 5 is a summarized description of the system support software. Logic diagrams are in Sect. 4.3 of the Appendix.

3. OPERATING EXPERIENCE

The computer system was installed at the HFIR in November 1967 and accepted in April 1969. Since then, the system has been used primarily for on-line reactor applications on a round-the-clock basis. The utilization is composed of routine reactor applications requiring calculation of average heat power every hour, listing of all safety channel annunciator alarms, measurement of scram rod release times and time of flight at the beginning and end of each fuel cycle, scan and alarm tests of the hydraulic facilities data every 15 min, projection of reactor burn-up calculations on request, and continuous calculation of the reactivity balance (or anomaly). Each of these functions has been in routine use for more than 2 years, and they form the basis of the reactor on-line computer operating experience at ORNL. In addition to the routine utilization, computer control experiments at the HFIR were among the first in the United States to demonstrate direct rod control at power. The following summary of observations is taken from the discussion to follow.

1. Reactivity balance is a valuable reactor parameter that can be observed and utilized with the aid of an on-line computer.
2. Direct rod control can be successfully performed with a process control computer.
3. Programmable systems require rigid quality assurance procedures, and the computer can aid considerably in complying with these requirements.
4. Computer system failures can be gradual as well as abrupt.
5. The computer hardware currently available can provide a high availability if supplied with a back-up system and good maintenance.

3.1 Software Operating Experience

3.1.1 Language Considerations

The question of whether or not a higher computer language, such as Fortran, is suited for a process control system was raised early in the development program. A

Table 5. System support software

Name	Task Summary
ASMFIX	Provides capability to assemble source language programs from mass memory
DACDRI	Subroutine which outputs values to all the digital-to-analog converter channels except channels 0 and 1
DECLCM	Declares the common variables for all the on-line programs
DIGCON	Subroutine which controls output to the power relay drives for movement of the HFIR shim safety rods
DRI72	Subroutine for operating the high-speed 200 kHz timer
EDFILE	Provides temporary storage space for source programs being tested or debugged
EDIT	Provides capability to modify statements in source language programs stored in mass memory
FTNFIX	Provides capability to compile Fortran source language programs from mass memory
FXCHRS	Used during system rebuild to absolutize the addresses in program CHRIS
GIDUP	Updates the burnup calculation and restarts the system programs following a computer outage
KILL	Terminates programs EDIT and CHRIS and reestablishes initial drum addresses in FTNFIX and ASMFIX
PATCH	Executed following system rebuild to automatically provide several required software changes
PSYCHO	Reestablishes the operating system following an interrupt caused by the stall alarm, protect, parity, a power outage, or a manual master clear
STATGO	Initiates the high speed data acquisition program MUXBUF for use by the neutron noise analysis calculation
STLCHK	Responds to the stall alarm interrupt, determines type of stall, and provides message
STLSET	Resets stall alarm watch-dog (timed dropout) relay every 0.067 sec
UPDWN	Responds to console light "Process Scan" by setting appropriate flags and initiating the process scan programs

significant amount of work had been done on the reactivity balance calculations prior to specifying the HFIR computer system. This work had been accomplished with Fortran programming, utilizing the ORNL computing center machines. To maintain continuity of this effort, a Fortran capability was specified for the HFIR machine. The choice of writing control programs in Fortran or assembly language was deferred until ORNL personnel could familiarize themselves with the idiosyncrasies of the particular computer delivered. In the time between the contract award and system delivery, block diagrams of the programs to be implemented were developed and the assembly language of the CDC 1700 was learned. During this time a philosophy finally crystallized: programs that required many floating-point calculations or were based on previous Fortran work would be written in Fortran; and programs that included many bit-by-bit logical operations, input-output operations, or integer manipulations of process data would be written in assembly language. There were, of course, some areas of overlapping requirements, and in these cases the language decision was based simply on the skill and preference of the control engineer writing the program. This approach proved successful, and is recommended for future systems having the dual capability. The least satisfactory system would be one having only the high-level language. Since it is important that the operation of the computer be fully understood by the control engineer, a system having only a high-level language would tend to preclude the required familiarity.

From operating experience, we believe that systems of this type should be supplied with a text editor type language to simplify the task of writing new programs or correcting existing programs. When it is known that many new programs will be prepared and debugged on the system, the capability for trial assemblies or trial compilations of "soft copy" should be specified. The term "soft copy" refers to programs that have been written on-line and stored in image form on bulk memory. This scheme will allow the user to locate (and correct) errors detected by the assembler or compiler without having to make hard copies of the intermediate versions which still contain errors.

The HFIR system was originally supplied with a compiler and assembler that would only accept source images from the paper-tape reader, requiring repunching of the program being debugged before each attempted compilation. Since many errors

were not detected by the compiler before preliminary errors were removed, frequent repunching was not uncommon. The difficulty was corrected by writing a text editor program to store ASCII image source material on drum and by modifying the compiler and assembler to read programs from this image. EDIT is the name of the program editor routine, and a complete list of commands and program capabilities is given in Vol. 2.[7]

Besides a text editor, the operating experience revealed a need for a method of modifying individual binary words in core or on drum. This method would be most useful when corrections are needed to a given program and time or circumstances will not permit a full system rebuild to install a corrected program. Some programs have constants which can best be established with operating experience. Generally these constants can be changed on-line, if a method for doing so is provided. This type of program is usually called a binary editor. Clearly, the hazard of incorrect usage of this type editor must be recognized, and its use should be under strict control. The operation of a computer system with undocumented program changes is like operating a hardware system with unknown wiring alterations.

The binary editor program CHRIS was used extensively during the first years of the operation, mostly to fixed addresses or known errors in the system following each rebuild. This method was later replaced by a more speedy process based on a special rebuild program called PATCH. The average rebuild time using PATCH is 1-2 hr compared with 4-6 hr with CHRIS. PATCH is a Fortran program with assembly instruction overlays, and it contains ample print statements to clearly document the rebuild statistics.

The rebuild procedure is shown in Sect. 4.4 of the Appendix along with a typical listing generated by the system initializer program (SI). The SI program is terminated by the *T command (step 22), and the remainder of the rebuild is done by using PATCH and CHRIS.

During the first 2 years of system operation, rebuilds were made about every 3 months, and later the frequency dropped to once per year. Operating personnel need a program like PATCH to guide the rebuild procedures when the rebuild frequency is very low, since it is not possible to remember all of the details of a complex process when it is infrequently practiced.

3.1.2 The Burnup Model

As discussed in the preceding section, the requirements of the burn-up model were developed before delivery of the on-line computer. The basic plan was to formulate the algorithm and attempt to fit the cross sections by a method of least squares such that the model described the reactor as predicted by the reactor design codes. As operational data became available, the constants were adjusted to more accurately describe the actual system. The Xe constant was established by observing the reactor during Xe transients like those shown in Figs. 26 and 27. The Sm term was determined by observing an extended transient that caused the Sm concentration to reach the peak value while the Xe decayed to zero concentration. The constants for uranium and boron were adjusted to describe the total excess reactivity change for steady-state burnup during the 23-day cycle, as shown in Fig. 22.

The main program for the reactivity balance calculation is the burn-up program SC. The basic form of program initially installed is similar to the test program included in the purchase specification document shown in the Appendix, Sect. 4.1. The subroutine EQUA was later incorporated within the main SC program. The initial philosophy specified that SC reside in fixed core; however, as operating experience was accumulated, it became obvious that large programs that run only twice each minute should be drum resident. The CALLSC program is employed to smoothly handle the operation and release of the drum resident SC.

Several operational features had to be incorporated to handle initial conditions for both: routine starts at the beginning of a fuel cycle, and updates (or restarts) at some time into a cycle. The initial conditions for a new core consist of an initial common array, stored on drum at $D/$7F00. The atom densities for U-235 and B-10 are fixed at the core design values, and the remaining terms are zero. Program UPDWN is used to identify the start of a new fuel cycle whenever the scan program is initiated by the operator. When a new cycle is identified, the initial common is transferred from drum to core COMMON and is used for the remainder of the fuel cycle.

The difference between the calculated change in reactivity and the measured change in reactivity is called the reactivity anomaly (Rhodif) and is defined in Sect.

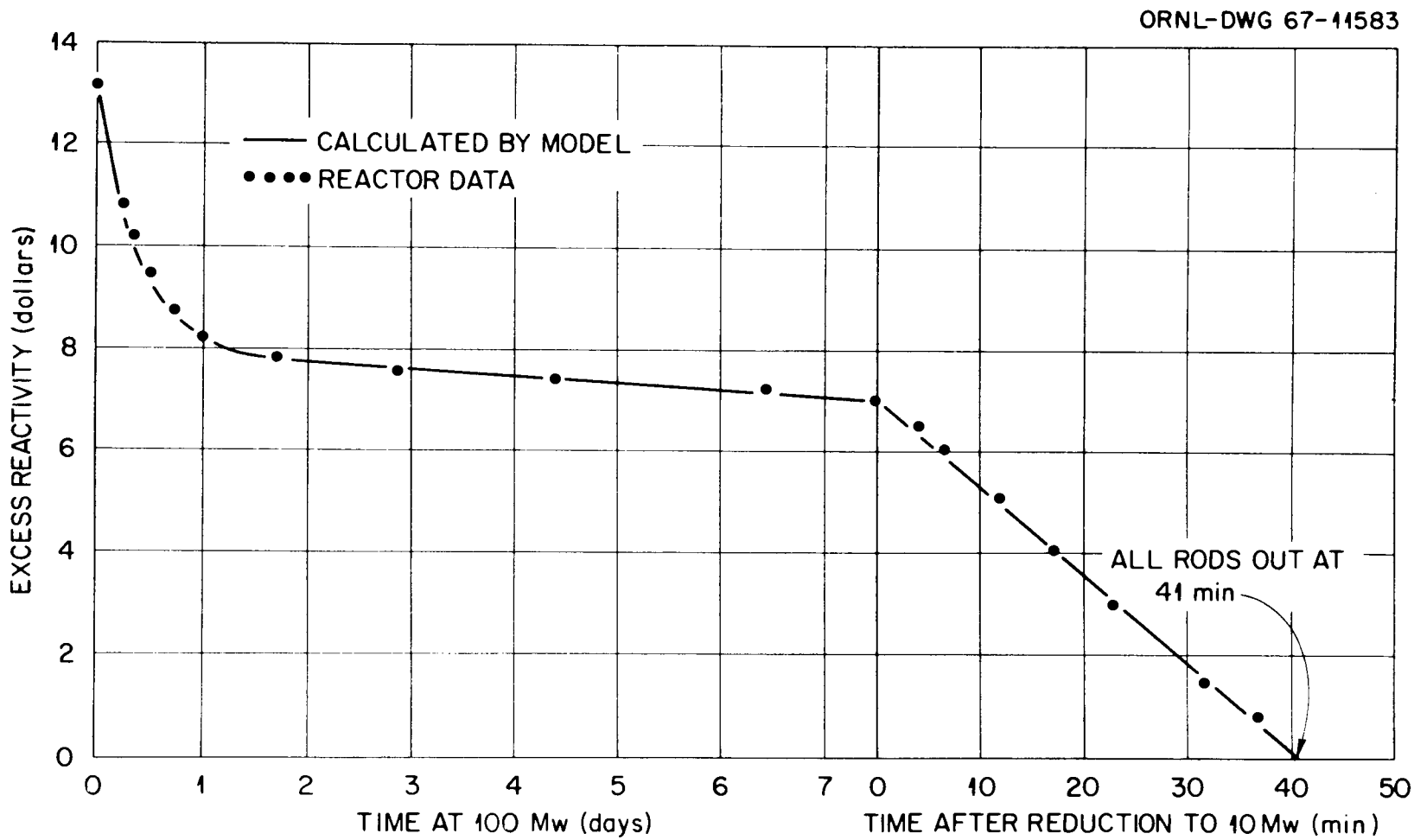


Fig. 26. HFIR reactivity transient following a power reduction to 10 MW.

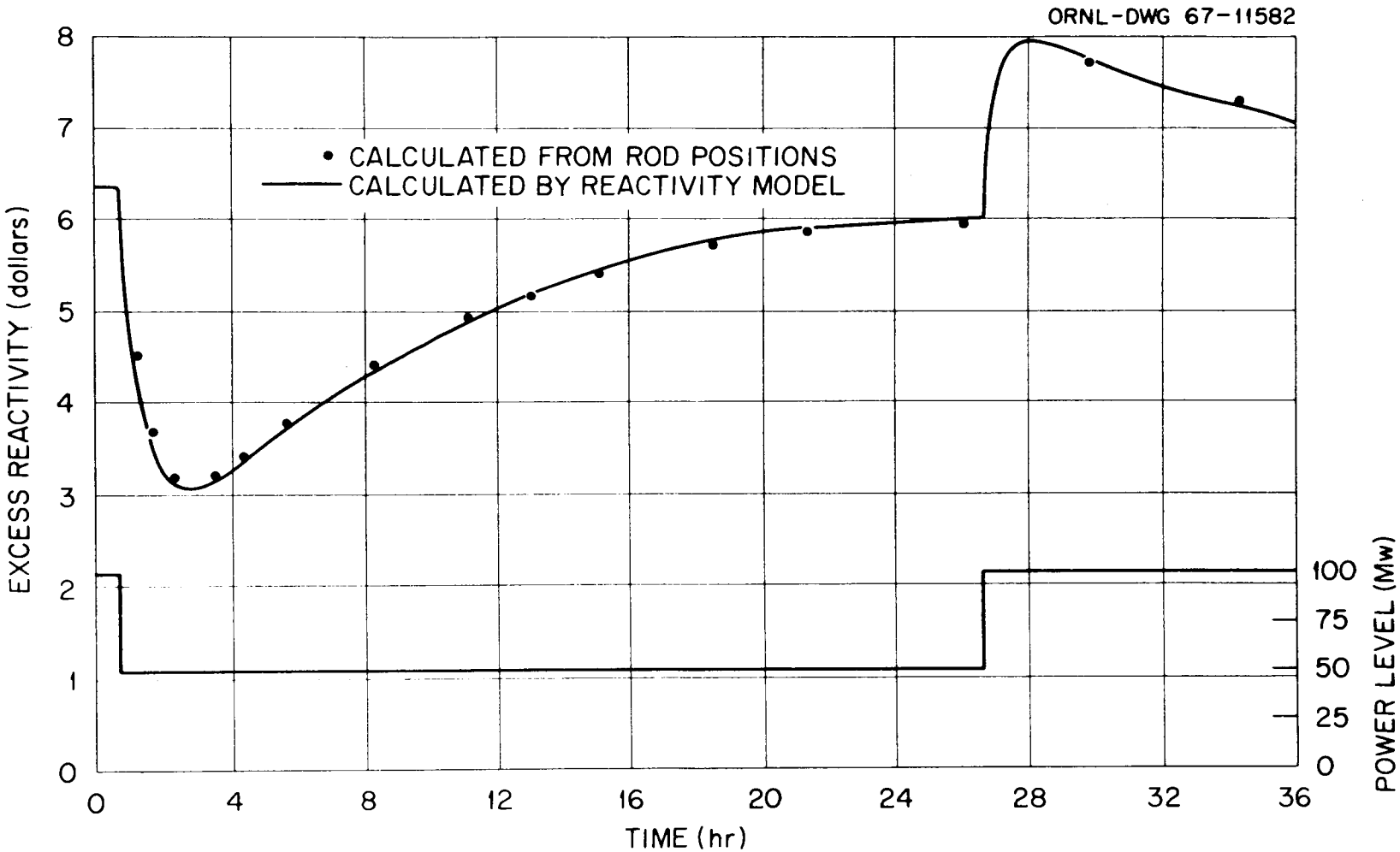


Fig. 27. HFIR reactivity transient following a power reduction after 7.8 days at 100 MW.

2.4.7. Rhodif is calculated each second, stored in core common as a control parameter for use by other programs, and is plotted continuously for use by the operator.

Since Rhodif is ideally a continuous variable throughout each reactor cycle, an important consideration was that of deciding how to reestablish or update the value following a system outage. The same question applied to the recovery problem following any inadvertent destruction of the core image (memory). Since the burn-up calculation is a running integration, the loss of the current values prevents continuation of the calculation.

The first method devised to handle this problem was an updating program which was executed after recovery from any outage of sufficient duration to produce significant errors in the burn-up calculation. It was decided to store all variables of a volatile nature (including time) in the common region or core, so that a paper-punch copy could be made each day during a fuel cycle. This update program starts with the last "good" values from the punched tape and integrates the burn-up equations over the time interval needed. Because additional tasks needed to be performed following such restart conditions, a major development was undertaken to resolve the restart problem.

3.1.3 Automatic System Restart

A large number of components in process control computers can result in an occasional high incidence of momentary failures. This observation was reported by Pearson[13] concerning the early applications of digital computers in Canadian reactor systems. Since a recovery method is needed for failures wherein it must be assumed that any or all of the core image is damaged, the recovery technique was extended to include power outage, memory parity error, stall alarm, and any dead-loop hang-ups. Except for the latter, all the aforementioned difficulties will cause an interrupt which can be used to branch to an automatic recovery program. If the trouble is related to or has altered the core memory containing the autorestart program, the system will require manual intervention, as will the dead-loop. To reduce the number of manual operations required for restarting the system, the paper copy of the common area was replaced by a drum copy. This is accomplished by automatically writing a copy of common (containing the current time and date) onto drum storage each second. Except in rare instances

in which this image is also destroyed, the restart program needs only the current time and date to update the burn-up calculation. This feature was further automated for outages of short duration; a battery-powered clock automatically starts when the computer is master cleared and runs until it is read and reset by the restart program. The automatic restart program attempts to perform the same tasks required manually, i.e., master clear; read new core image from guarded drum; read current time and read last saved common; and execute the update program, which also starts the system scan.

Restart programs GIDUP and PSYCHO are shown in Vol. 2.[7]

3.1.4 Software Quality Assurance

The ease with which a computer program can be altered may deceive the inexperienced user into thinking that it is as simple as making the change. Repeated examples of unsuccessful attempts to make a simple program change suggest the hypothesis that "a simple change is a rare occurrence in on-line process control systems." The user of programmable systems should recognize that altering program instructions is equivalent to changing wiring connections in conventional systems. Therefore, such anticipated actions should be reviewed and planned in detail to foresee the various ramifications.

Like conventional systems, programmable systems require a complete description of the installation, but, unlike the hard-wired system, the computer can provide much of the needed documentation. The program listing for a typical control algorithm contains all the details of the control function and is analogous to the "as wired" wiring diagram of the analog system. When the programmer is careful to include ample comments throughout the listing, it may also serve as a functional description document for design review purposes. The compiler listing contains the program instructions as written by the designer and are therefore not subject to "wiring" errors during installation. Obviously this argument assumes that the compiler or assembler has been debugged prior to on-line applications. The speed of the digital system generally will result in program revisions being available sooner than the equivalent revised drawings issued in the conventional manner.

Operating experience with an on-line control computer at the HFIR has indicated that quality assurance of the software is a major consideration in planning and operating such systems. The following practices proved most helpful in maintaining the integrity of the HFIR software:

1. Complete listings of all system programs are maintained in a clearly indexed file or notebook. This file includes a list of the previous version of any revised or corrected program.
2. A copy of all system program source images (cards, tape, etc.) is maintained, clearly labeled by name and revision date or number.
3. A logic block diagram of all complex programs showing their relationship to other programs in the system is maintained. Both the serial logic and the parallel logic forms described below are used for the more complex programs.
4. Appropriate designer comments are required in the comment fields of each program to clearly document the intent of the adjacent computer instruction groups.
5. A software change procedure was established and rigidly enforced. This procedure includes:
   a. a designation of person(s) responsible for reviewing and approving change requests,
   b. a designation of person(s) responsible for making the change,
   c. a description of the method or test required to verify the anticipated effects of the change,
   d. documentation changes, including new program listings, new logic diagrams, and new source copy.

Many of the techniques specified above are reasonably obvious, which, no doubt, many computer operators have recognized and implemented. However, the construction of both serial logic and parallel logic block diagrams is probably unique with ORNL and merits further consideration.

A serial logic diagram is a conventional logic diagram usually drawn by computer programmers showing yes-no decision blocks and the functions being performed as the logic flows in a manner roughly synchronous with the program instructions. The term "parallel logic" is applied to a logic diagramming technique developed for analog control system design; this diagramming technique has been used for many years at ORNL.

In the past this technique was used extensively for showing the design objectives in control systems based on relay logic. Recently, however, the technique has been successfully applied to the construction of block diagrams of complex, computer program logic. The principal advantage of the parallel logic diagram is illustrated in Fig. 28 in which a simplified algorithm displays two identical logic functions in both the serial and the parallel logic diagram form. This display makes it apparent that the parallel scheme is ideally suited for ascertaining the multiple paths or conditions to a given end state; whereas the serial logic diagram defines the programming steps more exactly and is consequently more useful than the parallel diagram in verifying that the actual program instructions will perform the desired functions.

When applied to complex programs like the DRC program shown in Figs. 25 and 29, the two techniques are a very efficient and compatible method for lucidly documenting the design objectives, the significant variables, and the overall control system logic. Documenting these important factors in a clear, simple manner provides the only meaningful information base on which detailed design, high quality reviews, and operational debugging can be planned, because, in software systems, as in conventional system design, the basic man-to-man interface is also a major deterrent to the assurance of high quality.

3.2 Reactivity Balance

Operating experience with the on-line reactivity balance has been very good. The data from 3 years of operation clearly indicate that reactivity anomaly (or balance) is a workable practical control parameter for automatic control systems using an on-line computer. The only abnormal experience approximating a "real" anomaly situation was handled by the system in the expected manner. This event occurred during a test startup in which the automatic symmetry control program DRC was given control of the rods. The experience provided an unscheduled demonstration of the proper operation of the reactivity anomaly as an automatic control system parameter. The events were summarized in a postexperiment review and are paraphrased as follows:

During the startup tests at the beginning of HFIR cycle 52, the rod-drop recovery program did not function as anticipated. The associated messages are shown in Table 6.

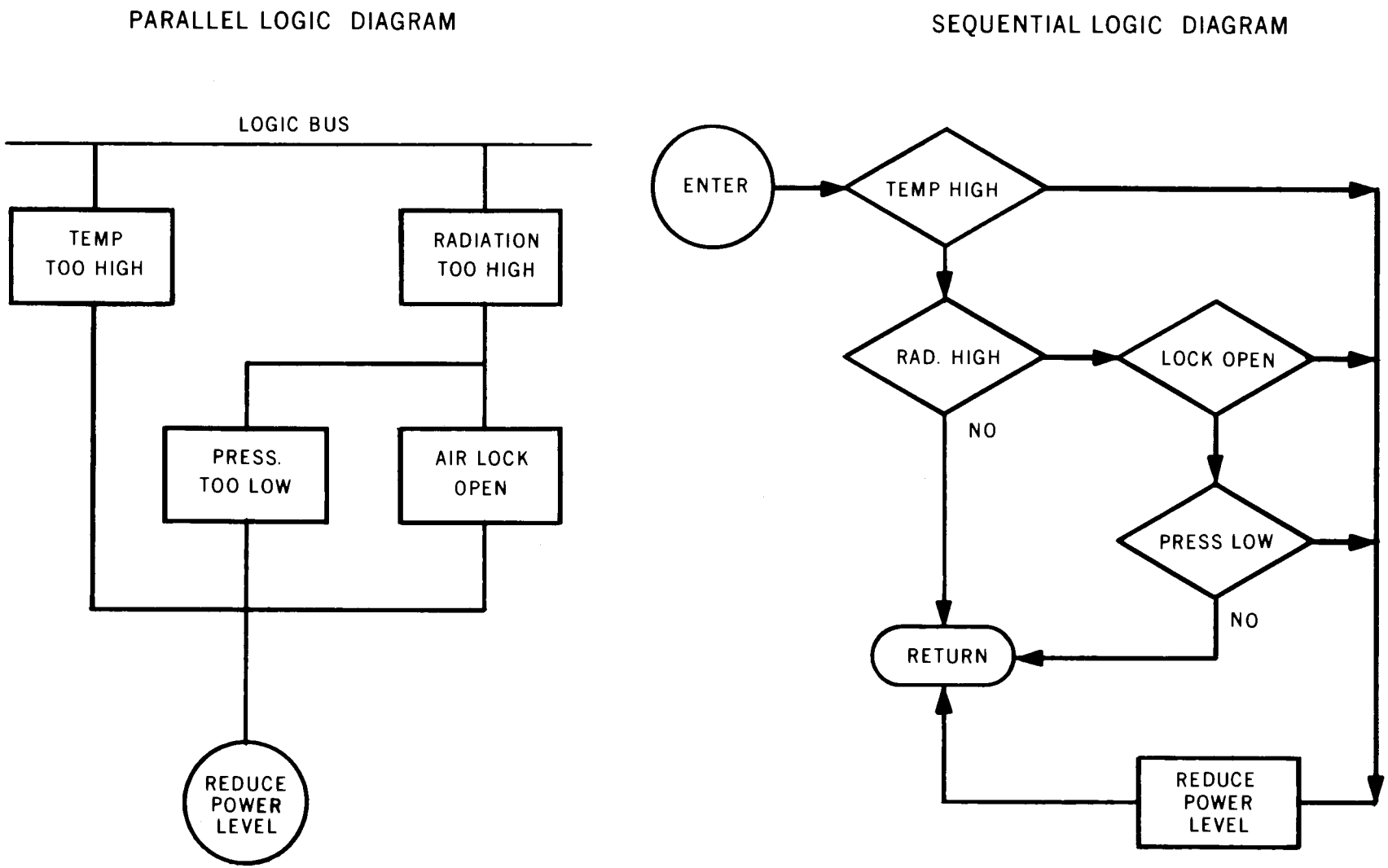


Fig. 28. Simplified parallel and sequential logic diagram.

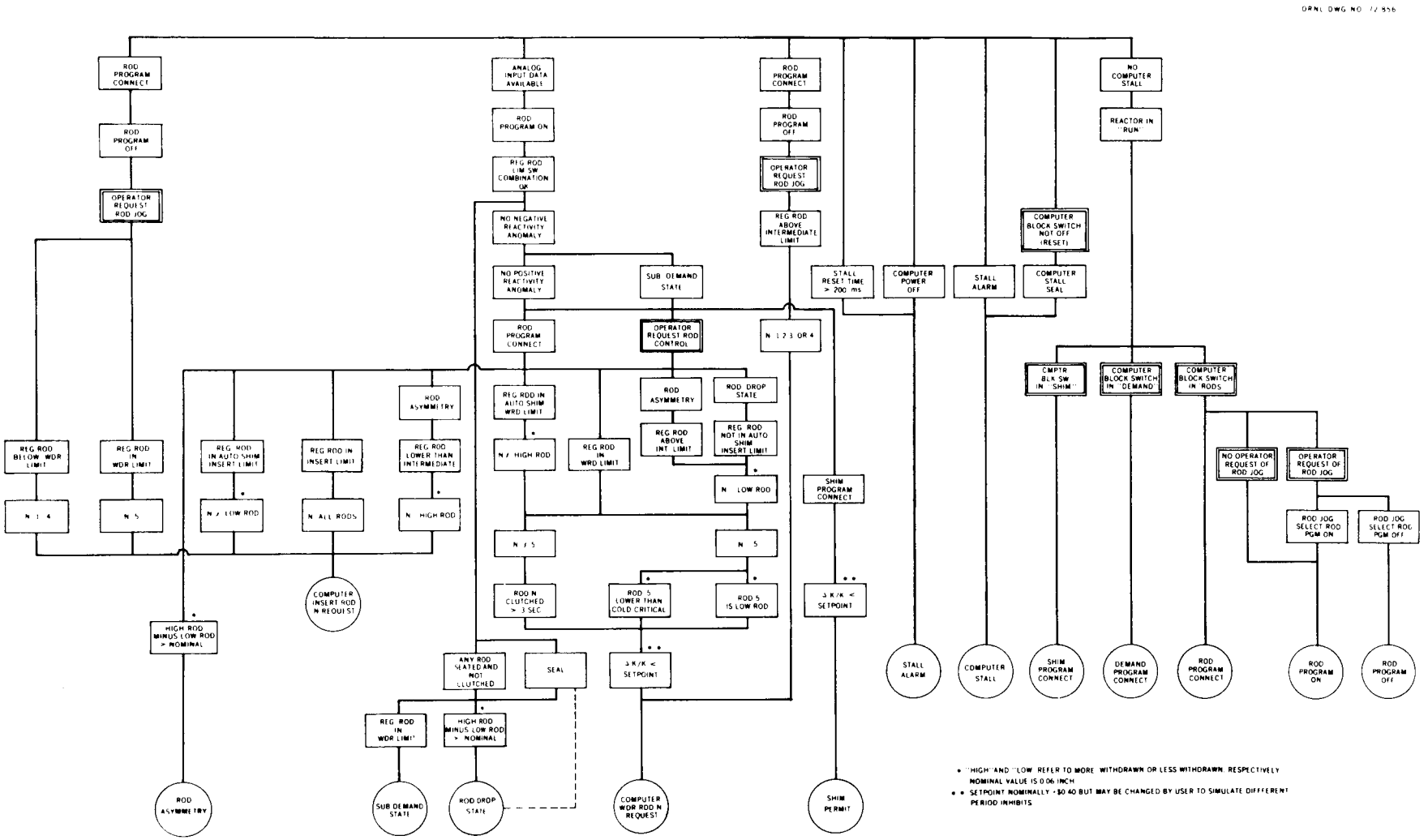


Fig. 29. Parallel logic diagram of program DRC.

Rod 1 was released at 01:10:38; this properly resulted in "Rod Asymmetry" 4.08 sec later (line 2). The reactivity anomaly at 01:10:39 (line 3) occurred because the rod was in flight, and line 4 ("DRC RTN") indicates that the digital rod control had returned to normal after the usual 1-sec delay. The "RODS" program was selected at 01:10:48 (line 5), and the regulating rod reached the withdrawal limit at 01:10:49, as shown by line 7. The computer properly shimmed the rods from the time "RODS" was selected at 01:10:48 until 01:11:14, when lines 8 and 9 were typed: "Reactivity Anomaly," and "Rod Control Denied." At 01:11:42 the "DRC OFF" (line 10) was actuated, and the operator manually withdrew the rods until rod 1 reached 4.90 in. At this time the anomaly had reduced to <50 cents, and the digital rod program was turned back on (line 11). The automatic symmetry recovery proceeded normally thereafter, with "Symmetry" reached at 1:15:29 (line 12). The test was repeated with basically the same results.

Table 6. Alarm Messages

		Rod Position x 100 (in.)				
Line No.		1	2	3	4	5
1	ANNUNC 0110:38					
2	Rod Asymmetry 0408					
3	Reactivity Anomaly 0110:39					
4	DRC RTN 0110:40	1678	1776	1777	1779	1824
5	"RODS" 0110:48	1329	1776	1777	1779	1824
6	RR WDL 0110:48	1329	1776	1777	1779	1824
7	RR WDL 0110:49	1287	1780	1782	1784	1825
8	Reactivity Anomaly 0111:14					
9	Rod Control Denied					
10	DRC OFF 0111:42	0000	1892	1894	1891	1820
11	DRC ON 9112:57	0490	1910	1912	1905	1809
12	Symmetric 0115:29	1780	1779	1779	1779	1779

Following a study of the programs, the only possible conclusion was that a reactivity anomaly did, in fact, exist during the tests. This conclusion was substantiated when it was learned from the weekly report following the startup that control plate 5 had been replaced during the preceding shutdown. Since the reactivity calibration for a new control rod had not been entered into the computer, the algorithm calculated an anomalous reactivity condition while shimming to compensate for the dropped rod and, consequently, properly stopped further shimming.

The operation of the balance calculation as a control parameter was successfully demonstrated throughout two fuel cycles by allowing program DRC to perform all the rod movements required for the entire cycle. These tests also demonstrated the use of reactivity anomaly as a control parameter for routine full-power operation. The value at initial critical is normally zero, and a gradual reduction to -0.30 dollar occurs during the first 20 days of operation. The value subsequently starts increasing to about +0.50 dollar during the last day of operation. The anomaly inhibit is set at -0.50 and +1.00 dollar unless a rod drop state exists. During a rod drop condition the +1.00 dollar is not valid because the large positive reactivity is normally expected.

Although the preceding limits may appear large, the criterion is established to prevent the control system from causing a reactivity excursion that would exceed the capability of the safety system. In the HFIR this value is near 2.00 dollars, which is well outside the -0.50 dollar limit. In applications requiring more accurate calculations, it will be necessary to include additional effects, such as power coefficients, etc. The values for the HFIR have been defined more accurately by this method.[14]

The incentive for very high accuracy is not based on the above criterion but rather on improving diagnostic capability. The Enrico Fermi malfunction detection analyzer[15] uses a similar reactivity balance technique to detect anomalous power coefficients at sufficiently low power levels to prevent excessive fuel heating. This type of parameter would also be adapted for use as a control variable during automatic shim rod control.

3.3 Direct Control of Shim Rods

The operating experience with direct control of the HFIR shim safety rods demonstrated the value of programmable controllers in applications requiring rapid and complex manipulation of the controls. Automatic recovery from a single-rod-drop condition was successfully demonstrated by numerous on-line experiments. The direct rod-control algorithm (DRC) was initially developed for this explicit function[16], but was later improved to perform all the shim rod movements for routine operation at 100 MW.

A typical automatic rod-drop recovery sequence is shown in Fig. 30, wherein the reactor was operating at 10 MW and rod 2 was dropped. The -2.50 dollar reactivity inserted by the dropped rod was rapidly nulled by withdrawing the remaining four rods. The demand power was reestablished within 40 sec, and rod symmetry was regained in 3 min 50 sec. This value may be compared to a typical manual response during an unscheduled rod drop on June 30, 1969, when the operator required 10 min to reestablish rod symmetry and consequently allowed the reactor to be poisoned by Xe-135. The average time to recover from Xe poisoning in the HFIR is 70 hr if the Sm increase does not permanently poison the core. In addition to its speed, the control algorithm for rod symmetry control utilizes the reactivity balance parameter; therefore, it will not move the control rods if an error or a reactivity anomaly exists.

The philosophy of separate control and safety systems is strongly enforced at ORNL. The single computer controller was classified, therefore, as a nonredundant control device.

Since the HFIR safety system is designed primarily for control system failures, it is the primary backup system to the computer controller.

3.3.1 Safety Considerations

The fundamental basis for the safety analysis of the computer installation is the separation of the control system from the safety system. Rigid separation of safety systems from control systems is required at ORNL, which dictates that a separate and highly reliable safety system must compensate for any malfunction of the control system. In this context, the digital computer is defined as a control device, and

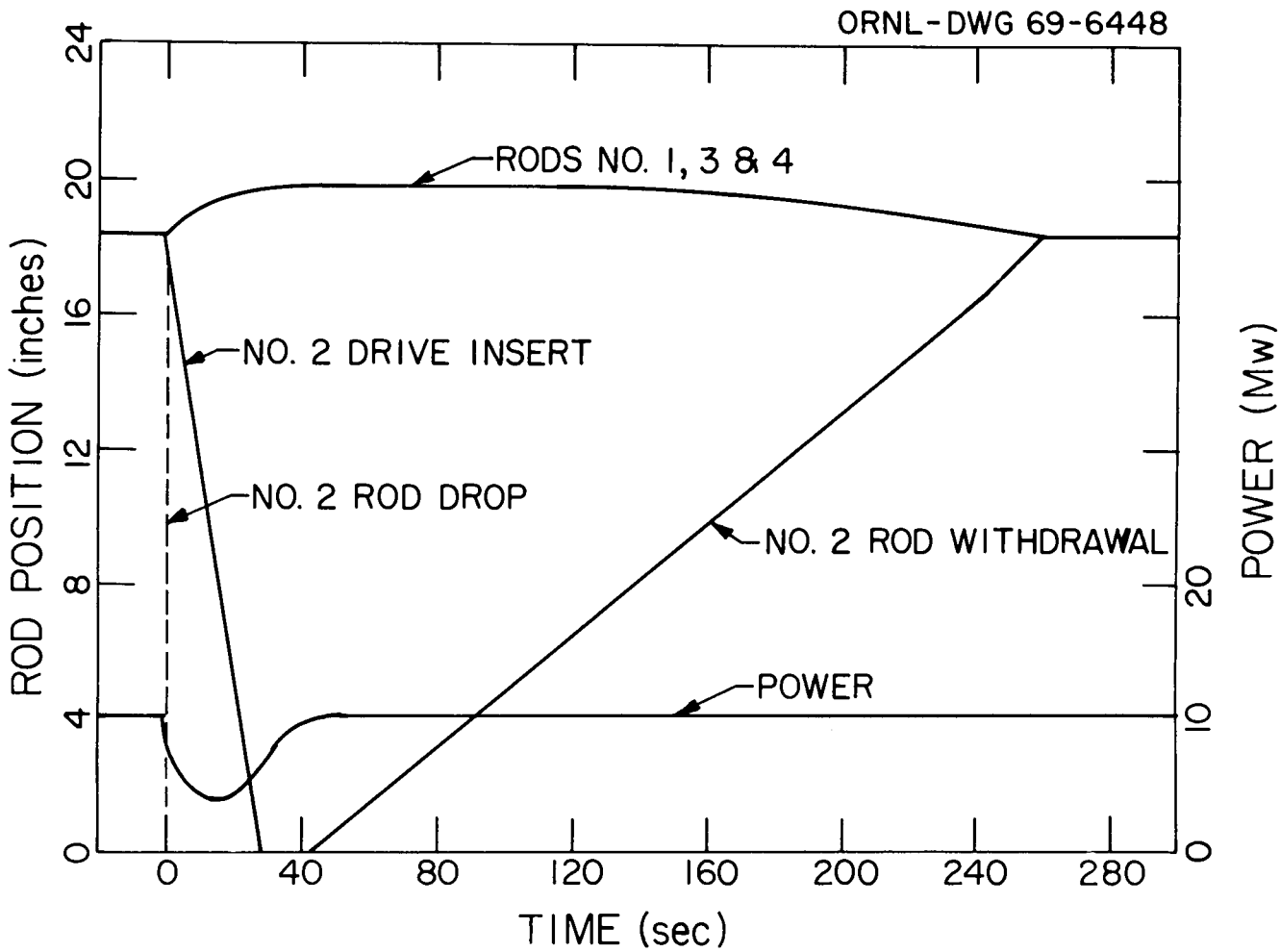


Fig. 30. HFIR computer controlled rod drop recovery.

safety-grade reliability requirements are not imposed for this application. The design of the HFIR computer is such that the computer will in no way be connected directly to the safety system. This is important to ensure that computer, control system, or operator failures will not prevent proper action of the safety system. In addition, features such as rod reverse, period inhibits, power reverse, etc. are retained in the analog control system for intervention in the event of control component failures. Some of these are shown in Fig. 1.

The area in which computer connections come closest to safety system components (but still separated by two levels of isolation) is the computer scan of safety annunciators. The computer receives information from contacts on the annunciator relays. The annunciator relays are actuated by contacts on relays in the fast-trip comparators in each channel of the safety system. Thus, there are two relay coil-to-contact isolations of the computer from each channel of the safety system (Fig. 31). Failure of either or both of these relays does not affect the protective capability of the safety system.

The computer is connected to the control and process systems to receive information and to give commands to the rod drive system based on its input information and the control algorithm being used. The computer has a total capacity of 80 input signals and 32 output signals.

The input signals from relay contacts consist of 31 inputs from annunciator contacts (both safety and process annunciators), 4 rod seat contacts, 4 rod clutch contacts, and 5 regulating-rod limit-switch contacts. Other input signals are 9 low-level analog inputs (10-50 mV signals) and 13 high-level analog input (0-5 V signals). The low-level signals consist of differential pressure and temperature signals, and the high-level signals consist of rod position and reactor power signals. None of these signals is taken directly from the safety system.

The digital output signals from the computer consist primarily of relay driver signals. Ten of these signals command insert and withdrawal of the five control rods, and one signal is needed for the automatic shim withdraw permit.

A manual control switch allows the reactor operator to remove all computer output signals from the reactor control system. When the switch is in the Off position,

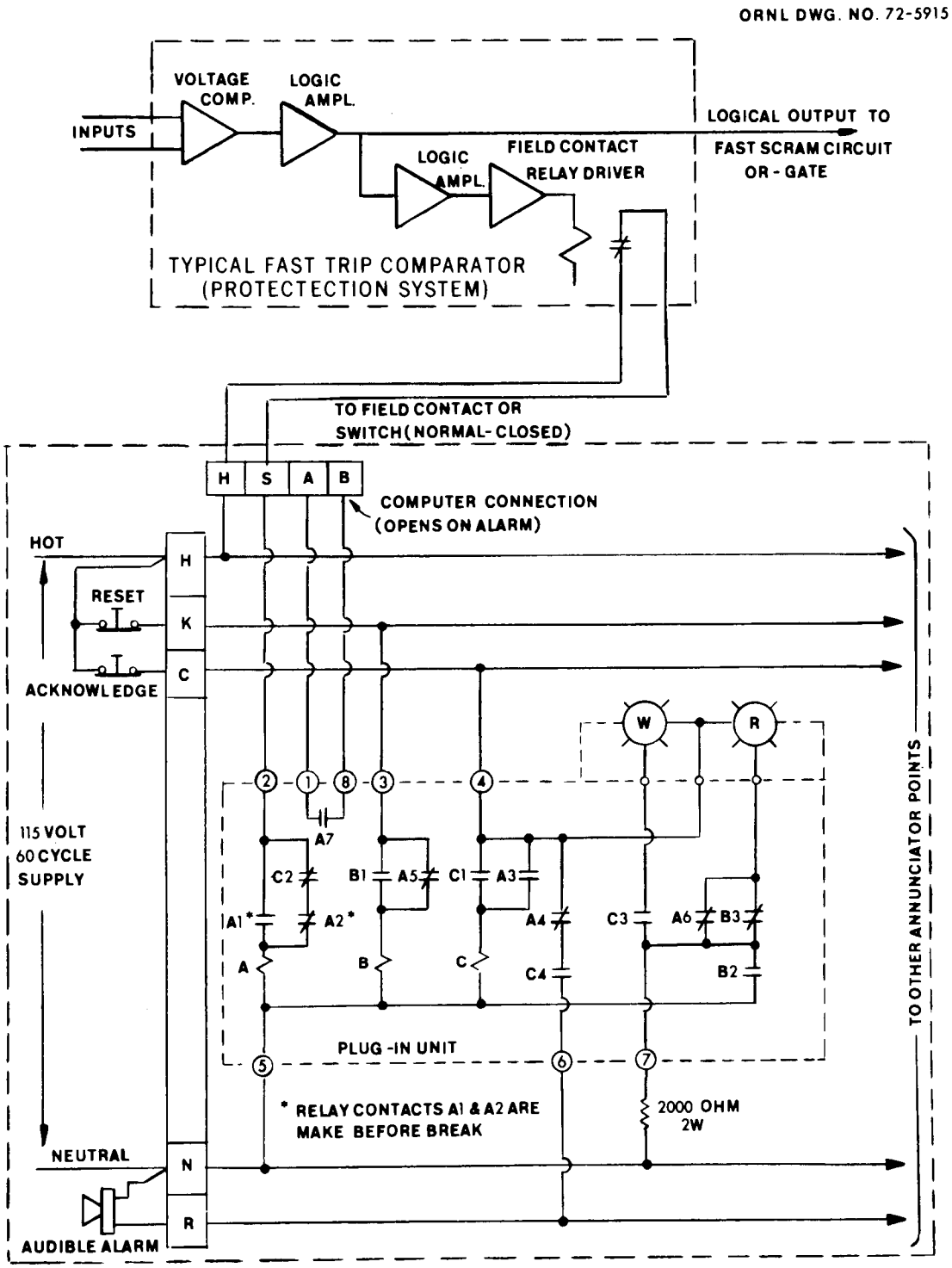


Fig. 31. HFIR safety system annunciator system.

the computer control signals are isolated from the reactor. The other switch positions include Shim (for permitting control rod shimming), Rods (for the rod symmetry program including shimming), and Demand (for power demand). The latter position has not been connected but will be used in future tests.

The computer receives three-phase 480-V power from an emergency bus supplied from diesel generator 2 during a power outage. However, the bus is deenergized for a few seconds following a power outage, and, at this time, a relay disconnects the computer from the control system regardless of the position of the computer control switch. The operator must take action to reconnect the computer to the control system following the resumption of normal power.

During planning of the computer installation, several postulated computer failure modes and effects of false information to the computer were analyzed. With an assumption that the computer would be effectively isolated from the safety system, the most serious consequence of a computer malfunction would be the simultaneous withdrawal of all control rods at the time of maximum rod worth in the reactor cycle. The time of greatest combined rod worth would occur at the beginning of the reactor cycle and would be equal to about 3 dollars/in. Since the shim rod velocity is 0.096 in./sec, the maximum rate of insertion of reactivity caused by withdrawal of all rods at this speed and at the most reactive position would be about 29 cents/sec.

The HFIR Safety Analysis Report[17] shows that a reactivity excursion of 70 cents/sec would not induce reactor damage even if the safety system did not function to limit the initial excursion. The temperature coefficients would be sufficient to limit the initial excursion and prevent core damage. In the actual case, a reactor reverse by the control system at 110 MW, a scram by the safety system at 130 MW, or a power rate scram at 20 MW/sec would be sufficient to safely terminate any control system malfunction. Although a reactor excursion caused by the computer would be undesirable, such an excursion would not damage the reactor.

3.3.2 Misoperation During Early Tests

A misoperation occurred July 22, 1968, during a test to demonstrate the ability of the computer to restore rod symmetry and normal operation, following a single-rod drop. At the time of the occurrence it was assumed by some observers that the computer had initiated a disturbance that the control system could not quell effectively. A subsequent study of the misoperation revealed that, although the computer was certainly involved, the principal difficulty was caused by poor performance of the control system and by untimely application of a jumper as an effort to correct the control system deficiency. This poor control system performance is significant for both manual and automatic rod manipulations, and it is noteworthy that the computer exposed the shortcoming. A description of the test is given below.

A single shim-safety rod was deliberately dropped to allow the computer to demonstrate its automatic shim-symmetry program. The resulting sudden power reduction caused the servo system to withdraw the regulating rod at its maximum rate until the withdraw limit was reached. As the regulating rod passed through the Automatic Shim Withdraw (ASW) position, shimming was automatically requested by the computer. However, the shim withdraw request did not persist, because of misadjustment of the operating cam for the ASW switch. The regulating rod coasted considerably past the electrical withdraw limit because of its high speed, and the ASW switch actuator released the switch by rolling off the back side of the cam. This malfunction allowed only a momentary request for shimming by the computer. The computer engineers were alert to this trouble, having previously observed it, and attempted to simulate correct action of the ASW switch by jumpering contacts at the computer input as directed by a person watching the regulating rod position indicator. Proper shimming under computer direction ensued until criticality was restored at 10 MW following a minor overshoot to 15 MW. The regulating rod inserted as required to maintain 10 MW and cleared the ASW position. As it did so, the computer operator manipulated the jumper, simulating ASW switch action, to negate the ASW request, but due to a misunderstanding, he did this in such a way that ASW could not be reobtained through normal switch action. The computer automatic trim program was functioning normally, withdrawing the low (dropped) rod and inserting the high rod.

Since the dropped rod is of little reactivity worth initially, a net reduction in reactivity results, requiring withdrawal of the regulating rod to maintain criticality. The regulating rod passed through ASW to the withdraw limit, but automatic shim withdrawal did not occur because the misplaced jumper prohibited this action. The trimming continued to reduce reactivity, and the power sagged below 10 MW. The reactor operator blocked the computer and manually restored criticality again. Under the mistaken assumption that ASW was then working properly, computer control was reinstated; and, after a short period of trimming as before, the power again sagged.

During the power sag, the computer operator recognized the wrong application of the jumper and changed it to request ASW. The computer obliged by withdrawing shims, and shortly thereafter the reactor surged to what is now believed to have been about 55 MW before the servo became effective in restoring order. As the regulating rod was rapidly inserting and nearing the insert limit, the reactor operator blocked the computer, stopping withdrawal of shims.

The following information was derived from records and observations:

1. The transient is presumed to have been caused by withdrawal of all shim rods. One shim rod had been dropped and was still near its insert limit and ineffective.
2. The regulating rod was in its withdraw limit at the beginning of the transient, and the reactor power had sagged to about 1 MW (and was probably subcritical). The power was not less than 300 kW, because "Run" mode was maintained.
3. The peak power of the transient was recorded as 50 MW, with good agreement by at least nine recorders in the safety and control systems. These were D'Arsonval movements with first-order response $(1 - e^{-t/T})$. Their time constants were $\sim T = 1.25$ sec.
4. A potentiometric recorder monitoring a gamma chamber and picoammeter showed a peak power of about 40 MW. This recorder had a limiting slew rate rather than a first-order lag and required 12 sec for full-scale travel. Neither recorder had sufficient chart speed to show the shape of the transient.
5. Rate trip did not occur, indicating that the rate of change of power level did not significantly exceed 20 MW/sec.

6. That the transient was fairly slow was borne out by the close agreement of the two recorder types even though there was a large difference in their speed of response.
7. No power level safety trips were obtained (120-130 MW), but at least two observers reported seeing "Reverse" annunciators illuminated after the transient, which would imply a power level of 110 MW. This observation was inconsistent with the other findings and, if accurate, the "Reverse" indications may have been caused by a sympathetic response of annunciators due to noise. Several other annunciators were actuated at the time of the transient, but all were reset before being accurately recorded.

A follow-up study was conducted, and a kinetics model of the reactor was simulated on an analog computer. In addition to the normal fast recorders, the two types of recorders were used; their responses were equivalent to those at the reactor. The conditions of the reactor transient were duplicated, particularly with regard to reactivity rates, initial conditions, and the observed recorder readings.

The significant findings of this study demonstrated that no power overshoot beyond about 15 MW was possible if the servo worked correctly. A considerable delay before regulating rod insertion must be postulated to allow any significant power transient. There was some prior and subsequent corroborating evidence to indicate that the regulating rod may have been overdriven into the withdraw limit so as to require extra torque (and time) to accomplish insertion. Possible electronic malfunctions cannot be ruled out as a cause of the delay (estimated $\sim$0.5 sec) after the power crossed 10 MW.

During initial checkout phases of the HFIR, the regulating rod had stuck in both the insert and withdraw mechanical limits when the electrical limits had been badly misadjusted. After the July 22 transient, while the ASW cam was being adjusted, the regulating rod did overtravel to the mechanical limit when withdrawing at full speed. No sticking was observed on this occasion.

No transient could be postulated which would reach a level of 110 MW (sufficient to cause reverse) without exceeding both the observed recorder readings

and the rate trip level. It is highly probable that either the observation of "Reverse" was erroneous or the annunciator response was spurious.

A variety of transients could be postulated which would satisfy the observed conditions. The most likely case began at 2.50 dollars subcritical, at about 1 MW, and withdrawal of all shims but one. Servo action was delayed for about 0.5 sec, or until the power reached about 40 MW. Both recorder traces were typical, and an indicated rate of 20 MW/sec was just reached. The real peak power was 50-55 MW, with a peak transient period of 250 msec. The indicated peak period (counting channel) was about 5 sec.

Additional computer studies show that the count-rate period circuits have sufficient response to use for period inhibit of shim rod withdrawal in the Run mode. Period inhibit of shim withdrawal could substantially lessen the consequences of transients such as the one described, without interfering with normal operation. Presently, period inhibit and period reverse are effective only in the Start mode.

From these studies we recommend the following (after first stating that the operating cam for the ASW switch, the basic underlying cause of the disorder, has already been readjusted to prevent misoperation):

1. The possibility of applying dynamic braking to the servomotors to reduce coast should be investigated.
2. The application of period inhibit of shim-rod withdrawal in the Run mode should be considered. The relay drop-out times necessary to accomplish rod withdraw inhibit are longer than desired, even for the Start mode. A rearrangement of wiring to eliminate one or more relays from the chain appears possible. In addition, the surge protectors used on the relays can be modified to reduce the individual drop-out times. These modifications should improve performance in the Start mode, as well as in the Run mode.

It is appropriate to reemphasize that the digital computer was not responsible for the misoperation. Indeed, the computer initiated the chain of events that led to the misoperation, but it is highly likely that manual operation could have resulted in similar circumstances. The computer was therefore instrumental in revealing control system weaknesses in need of correction.

After these problems with the HFIR control system equipment were experienced, the rod control program was modified so as to test key hardware items before actuating the control rod drives. These test features were finally incorporated into a single program named CONTRL, which is entered each scan cycle prior to executing DRC.

3.4 Hardware Operating Experience

The performance of hardware in a system configuration is perhaps best characterized by the maintenance records for each major item of equipment. Table 7 summarizes the maintenance record of the low-speed input-output equipment of the HFIR computer system. The record clearly shows that the logging and alarm message typewriter required the most maintenance in this group, although the major work load was on the teletypewriter. Generally, since the operating system was not shut down when these items were being repaired, no downtime data is shown in the table.

The maintenance record of the mass memory unit is shown in Table 8. The major problems encountered with the drum were solved by adding a cooling fan and reworking a portion of one drum track. These problems seem to have been early design type errors that were satisfactorily remedied in later models. The excessive downtimes attributable to the drum unit were due (in part) to delivery delays and a need for factory trained repairmen.

Maintenance required for the computer mainframe is listed in Table 9. The major problem was core memory parity errors. Parity errors that require maintenance activity usually were not momentary because the autorestart system was designed to recover from such errors. The downtime shown was usually related to attempts to adjust the memory margins (or voltages) to correct the trouble. The actual time required to replace a memory module was less than 5 min.

Since January 1, 1969, the total system availability was ~97%, which was determined from a clock that operated only when the stall alarm did not indicate an outage. The time required for all system maintenance, completion of field change orders, software debugging, and testing was included in the percentage availability.

Table 10, which shows total availability vs time in operation for consecutive 6-month periods of operation from January 1969 to June 1972 shows a gradual improvement in availability after the frequent early problems were resolved.

Table 7. Low-speed input-output equipment maintenance record

Date	Indication	Diagnosis	Action or Equipment Replaced
		A. Paper-Tape Punch	
11/22/68	Feed holes tearing	Punch pins worn	Punch head
1/23/69	Constant alarms	Unnecessary ground	Removed strap
12/17/69	--	--	FCO[a] installed
9/24/70	--	--	Installed new part
9/25/70	Feed holes tearing	Spring needed adjustment	Adjusted
9/28/70	Feed holes tearing	--	Adj. eccentric lever
2/11/72	--	--	Installed FCO
		B. Paper-Tape Reader (Photoelectric)	
11/5/69	Alarms	Poor alignment	Aligned
12/3/69	--	--	FCO[a] installed
3/3/70	Stopped	Motor failure	New motor
7/23/70	Stopped	Belt broken	New belt
11/23/70	Stopped	Motor failure	New motor
8/19/71	Stopped	Belt broken	New belt
8/25/71	Stopped	Belt broken	New belt
8/27/71	Alarms	Clutch pressure	Adjusted
9/16/71	Alarms	Clutch pressure	Adjusted
9/22/71	Alarms	Load switch failure	Replaced
12/7/71	Stopped	Motor failure	New motor
1/7/72	Noisy	Capstan bearing	Replaced
8/1/72	Satisfactory		
		C. Teletypewriter (Model 35)	
7/23/69	Keyboard lock inoperative	Bad latch	New latch and spring
12/3/69	--		FCO[a] installed
12/3/69	--		Replaced
2/3/71	Extra space typed	Weak spring	Replaced
2/23/71	Stopped	--	Repaired
10/14/71	Typing errors	Weak spring	Replaced
12/1/71	Typing errors	Weak spring	Lubricated and adjusted
12/7/71	No break light	Contacts and light	Replaced
12/13/71	Typing errors	Delay adjustment off	Adjusted
8/1/72	Satisfactory		

Table 7. (continued)

Date	Indication	Diagnosis	Action of Equipment Replaced
	D. Logging and alarm message typewriter		
11/11/68	Motor stopped	Seizure	Replaced
11/25/68	Motor stopped	Seizure	Replaced
1/22/69	No red type	Ribbon shift	Adjusted
3/4/69	Bad type	Bad ball	Replaced
7/2/69	Bad type	Needs shift adjust	Adjusted
11/24/69	Noisy motor	Loose	Tightened
11/26/69	Hanging up	Bent tab	Adjusted
12/1/69	No red type	Shifter broken	Replaced
4/21/71	No red type	Shifter broken	Replaced
4/29/70	Motor stopping	Shifter jamming	Adjusted
5/1/70	--	New motor & shifter	Installed
5/25/70	Motor stopped	Started o.k.	
5/26/70	Motor stopped	Bad capacitor	Replaced
7/16/70	Shift locked		Springs adjusted
7/17/70	Stopped; no shift		New belt
8/2-4/70	Stopped	Needs overhaul	Replaced
8/10/70	Replacement wouldn't space		Old unit installed
8/25/70	--	--	New unit installed
8/28/70	--	--	Old unit installed
9/30/70	--	--	New unit installed but didn't work again
11/24/70	Constant interrupt	Bad card	Replaced
12/30/70	Misprinting	Needs overhaul	Replaced
12/31/70	No carriage return	New unit not o.k.	Replaced
1/4-8/71	Misprinting	--	Reworked
1/25/71	Stopped printing	Broken belt	Replaced
2/1-23/71	Hangup	Needs overhaul	
2/23/71	Hangup	Need new unit	Installed
4/8/71	Stopped spacing	--	Adjusted
4/15/71	Stopped spacing	--	Clip replaced
6/4/71	Spacing errors	--	Old unit installed
9/8/71	Noisy	Loose clutch	Tightened

Table 7. (continued)

Date	Indication	Diagnosis	Action of Equipment Replaced
		D. (continued)	
11/9/71	Hangup	Space adj. off	Set to 30 msec
11/11/71	Hangup		Replaced
1/17/72	Stopped printing	Broken drive tape	Replaced
4/17/72	Stopped printing	Broken belt	Replaced
4/19/72	No carriage return	Spring hose	Replaced
5/4/72	Hangup	--	Adjusted
5/8/72	Hangup	--	Adjusted
8/1/72	Satisfactory	--	Adjusted

[a]Field change order.

Table 8. Mass memory (DDC-drum) maintenance record

Date	Indication	Action or Equipment Replaced	System Downtime (hr)
1/28-31/69	Drum errors track-7D	Replaced head plate	60.0
4/10/69	PE[a]	Controller card-HA79	3.0
5/25/69	System halted drum rejects	Controller card-HA47	2.5
7/11-13/69	Solid PE	Controller card-K22	41.0
8/4/69	No write track 4B	Needs factory work	--
9/3/69	Intermittent PE	--	1.0
12/2/69	FCO[b]	DS 12888	0.5
4/30/70	Dropping bit 4	Card HA37 replaced	0.3
5/8/70	Timing errors	Timing track replaced	2.0
5/21/70	Scheduled repair	Replaced drum	3.0
5/21/70	FCO	DS 12654	1.0
5/26/70	Timing errors track 88	Adjusted logic voltage	
6/3-4/70	Timing errors	Card HA79	22.0
7/10/70	Constant interrupt	Card K22 replaced	5.0
3/8/71	Drum getting overloaded	Guard switch error corrected	0.5
7/4/71	--	Card HA47	2.0
2/11/71	Low pressure	Adjusted	--
6/5/72	FCO	DS 13084A	
6/5/72	Helium flow	New bottle	0.5
7/25/72	Helium leaking	New regulator	0.5
8/1/72	Satisfactory		

[a]Parity errors.

[b]Field change order.

Table 9. 1700 mainframe and core memory maintenance record

Date	Indication	Action or Equipment Replaced	System Downtime (hr)
10/28/68	PE[a]	Sense amplifier	--
11/26/68	Would not autoload	Q register failed	--
3/26/69	Nothing	16 cards replaced due to high voltage on 6-V bus	--
11/5/69	FCO[b]	--	--
11/19/69	PM[c]	PM and 2 FCO installed	7.0
1/21/70	Excessive noise	M-G replaced	39.0
2/2/70	PE bank 7	Repaired circuit connection	8.0
3/14/70	PE bank 0	Repaired circuit connection	6.0
5/22/70	Drum to core PE	Replaced module	0.2
5/28/70	FCO	New memory modules	6.0
6/1/70	PE bank 4	Replaced stack	4.0
7/27/70	PE	Checked margins	0.3
7/29/70	PE	Checked margins	3.5
7/31/70	PE	Interchanged banks	2.5
8/27/70	PE on restart from power outage	Restart not clearing, reset	2.0
9/24/70	PE	Margins test	1.0
9/28/70	PE	Margins test	4.0
10/19/70	PE bank 1	Ordered	3.0
10/20/70	PE bank 1	Replaced	1.0
11/2/70	--	Adjusted drive voltage	0.5
11/5/70	Frequent stalls	Margins tested, adjusted terminator supply	1.5
11/30/70	Frequent stalls	Adjusted power supply	2.0
3/6/71	PE bank 1	Adjusted drive voltage	3.0
4/21/71	FCO		1.5
8/19/71	Overheating	Air conditioning	9.0
8/20/71	PE bank 3	Overheated	20.0
8/25/71	PE bank 3	Replaced	1.0
9/1/71	PE	Test run	1.0
9/16/71	PE bank 6	Replaced	0.5
9/22/71	FCO	Installed	--
10/15/71	PE bank 1	Swapped to bank 3	1.0
10/19/71	PE bank 6	Replaced	1.0
12/13/71	High temperature light	Repaired	0.5
1/4/72	PE bank 0	Drive voltage adjusted	1.0
1/26/72	No autostart	--	3.0
2/24/72	Autorestart not giving memory reset	--	4.0
4/10/72	PE	Margins o.k.	1.0
4/11/72	PE core and drum	Drum test o.k.	1.5
4/13/72	PE many locations	Swapped cards	6.0
4/17/72	Scheduled PM	New bank	5.0
7/18/72	PE bank 4	Replaced	6.0
8/1/72	Satisfactory		

[a]Parity errors.

[b]Field change order.

[c]Preventive maintenance.

Table 10. Total system availability

Period	Total Availability
Jan-June 1969	94.2%
June-Dec 1969	94.5
Jan-June 1970	97.9
June-Dec 1970	96.8
Jan-June 1971	95.8
June-Dec 1971	97.2
Jan-June 1972	96.5

The average time-to-repair was 2-4 hr; we believe that the availability of a complete spare parts inventory has been a major asset in this regard. The high cost of a parts inventory is a primary incentive for purchase of an 8-hr-day maintenance contract from the computer vendor.

Although failure mode arguments are frequently based on an assumption that hardware systems failures are immediately obvious when they occur, the operating experience at the HFIR does not substantiate such conclusion. For example, a significant hardware failure resulted in a gradual reduction in the digitized low-level input signals. These signals were used primarily for reactor power-level determination by calculating the primary-coolant flow rate from a venturi differential pressure and multiplying by core differential temperature. Since all of these signals were being attenuated by the same defective component in the low-level preamplifier, the result caused a gradual but significant change in the calculated power level. The error was detected by an alert member of the operations crew before significant reactor power-level changes were made.

This type of incident can be guarded against by using a standard reference signal on the input system and routinely checking the standard input for maximum error. A check scheme of this type has been successfully used on other systems.

The need for a back-up system is reasonably apparent for applications needing a continuous on-line computer. With a unit availability >95-97%, the total availability for a two-unit installation should be adequate for most control applications. Such availability is not high enough for most plant protection system applications; however, it could be increased considerably by using a smaller system without a bulk memory unit but with simpler software structure.

3.5 Operator Control Console Experience

The primary channel of communication between the plant operator and the computer system should be through a console designed to suit human needs. The design features of this unit are very important in realizing a control system that gives the operator a "feel for the plant." Selection of the console display techniques should also be compatible with the computer outage or restart philosophy adopted. These stipulations will impose some special hardware and software requirements on the console design. For example, the console should be designed to recover from the same type of power outage that is designated in the mainframe recovery requirements. The display of currently operating programs or program status should be correctly reestablished after the system recovers from an outage due to power failure or after any other system restart; otherwise, the system cannot be considered as fully operational following the restart.

In addition to such obvious requirements as size, shape, and legibility of displayed quantities, the following factors should be considered when one designs or specifies an operator console:

1. recovery capability for all postulated perturbations for which recovery is desired;
2. positive response or indicators that reveal to the operator the disposition of any manual request, e.g., accepted, rejected, wait, go, etc.;
3. unambiguous response to multiple, redundant, or possibly contradictory requests (usually accomplished when well-defined functions and simple combinations of requests are employed);
4. convenient and simple method for entering numerical values or memory addresses (engineering units and familiar symbol terminology are preferred);

5. limit the information displayed directly to the operator to manageable rates and significant parameters (output capabilities of the smallest computer greatly exceed the maximum input rate of the most adept individual).

The design of the console at the HFIR (Fig. 32) originally specified only a few of these recommendations, largely because of a lack of experience with such systems.

The methods developed to reestablish the indications of program status following a system restart are as follows. The console originally installed enabled operator request entries by one of two methods. The first method required that the function be requested by dialing a number (00 to 99) on the two function select knobs at the right side of the console. After the function was selected, the Enter button was momentarily depressed to initiate the request. The button generated an interrupt that caused the console interrupter program to execute, using the digital input bus to read all console switch settings into a standard buffer table. The table could thus be used by the responding program to acquire any numerical values set by the operator with the digital input switches along the bottom edge of the console. A preconstructed table, indexed by function number, was referenced to locate the memory address of the requested program and then transfer execution to that address. When functions were requested by function number (as above), the operator was not provided with any indication of the proper execution except when some anticipated response was observed. However, the typewriter could be used to list an appropriate message when the requested program was executed.

The second method of entering requests from the console was by back-lighted pushbuttons for each designated function. The original intention of this feature was to continually indicate the status of these specific functions. However, the method by which the indicators were actuated and the actual program initiated proved to be inadequate. The operator was required to press the requested function button, which turned the light on, and then press the Enter button to cause the interrupt, which would start the program.

The button was also turned off by depressing it, and if Enter was depressed, the program function would be terminated. The deficiency in this technique is obvious: whenever the operator failed to depress the Enter button, the function indicator light

Fig. 32. Operator console and message typewriter.

incorrectly displayed the status of the program. The situation was further complicated if, at a later time, a function number was requested; then the Enter request applied to the function button status change before the function number request was accepted. To successfully execute the requested function number in that situation, the Enter button had to be depressed a second time because the lighted button change request was accepted when the Enter button was first depressed.

The deficiency was corrected by modifying the buttons of the important control functions, so that the light was actuated by the requested program rather than by the manual depression of the button. For example, the digital rod control program (DRC) is now requested by depressing the button so labeled and simultaneously depressing the Enter button. While the DRC button is being held down, the left half becomes lighted (Fig. 33). If the Enter button is simultaneously depressed and the program is successfully put into execution, the program contains a digital output instruction to turn on a light in the right half of the indicator button. The light remains on until turned off by the program or a system master clear. The Display Avg. Power button (Fig. 33) indicates that the program is operating. This technique provides a more positive indication that a request has been accepted and the program is actually running.

As discussed previously, the indicator lights are extinguished when the computer is mastered cleared, such as occurs during any sort of system restart. Since some functions ought to continue operating following autorestart operations, a scheme was required to automatically reschedule all such programs with the restart program.

The method developed for the HFIR was successful and should prove useful in similar installations. A console interpreter (DMF) constructed a record (bit pattern) showing the currently operating programs. This code word was saved with any additional information that was needed to resume proper operation following a restart. The core common area was used for this purpose, with locations $7F30 and $7F40 retaining the status of active button displays. The restart program interrogated these locations and proceeded to schedule the indicated programs as a normal part of the restart sequence. Since the individual programs turned the lights on when the programs were executed, the console status was fully restored during the restart procedure.

Fig. 33. Console switch light display.

An additional need for having each program actuate its own indicator light was evident when situations other than a direct request could initiate a program. For example, the program might be called into execution by other programs when certain plant conditions were detected. The HFIR computer control relays were wired in a way that allowed the operator to fully disconnect the computer from the reactor control system by turning a single SB switch. After some experience with the use of this arrangement, a decision was made that movement of this switch to the active position should also initiate the rod control program. This obviously required that the program activate the On indicator, and, by such a scheme, any mechanism that started the program also gave an On indication.

The method of entering numerical values on the operator console should be compatible with the type and quantity of values to be entered. The HFIR console has thumbwheel digiswitches (Fig. 32). Although this selection process is slower than one possible with a keyboard type console, the "set-switch" method permits verification of the number before it is entered and is most convenient when multiple entries are required. We recommend an equivalent capability in any entry station specification. The keyboard could have, for example, a numeric tally display, showing the values that will be entered when a "go" command is given. When engineering values are to be entered, a decimal place key is needed; hexadecimal symbols should be available if base 16 will be required. The conversion of a hexadecimal address to base 10 is very tedious and frequently is a source of errors.

Finally, a clear list of user directions should be prepared and conveniently located for the console operator. The list of the HFIR functions and their descriptive procedure is given in Table 11.

Table 11. List of console functions

```
**********************************************************************
FUNCTIØN NØ.                PURPØSE
************            *************


    00          DIGITAL DISPLAY ØF REACTØR PARAMETERS
                1. SET PØINT NUMBER IN L.H. DIGISWITCH.
                2. PRESS "ENTER".
                PØINT NØ.   PARAMETER          PØINT NØ.   PARAMETER
                ---------   ----------         ---------   ----------
                    1      RØD PØS #1             15      ØUTLET TEMP #1
                    2      RØD PØS #2             16      ØUTLET TEMP #2
                    3      RØD PØS #3             17      ØUTLET TEMP #3
                    4      RØD PØS #4             18      FLØW #1
                    5      RØD PØS #5             19      FLØW #2
                    6      DEMAND PØWER #1        20      FLØW #3
                    7      DEMAND PØWER #2        21      PØWER CALC #1
                    8      DEMAND PØWER #3        22      PØWER CALC #2
                    9      DELTA PRESSURE #1      23      PØWER CALC #3
                   10      DELTA PRESSURE #2      24      PØWER AVERAGE
                   11      DELTA PRESSURE #3      25      RHØ RØD
                   12      INLET TEMP #1          26      RHØ CALC
                   13      INLET TEMP #2          27      RHØ DIFF
                   14      INLET TEMP #3          28      ACCUMULATED PØWER
                                                  29    FRY FLUX (DR ØN)
                                                  30      DR LIVE (DR ØN)

    01          TYPEWRITER LØG ØF REACTØR PARAMETERS

    02          PRØJECTS THE TIME TØ PØISØN FØLLØWING A CHANGE IN PØWER
                LEVEL TØ X MEGAWATTS ØCCURRING T MINUTES FRØM NØW.
                1. ENTER THE NEW PØWER LEVEL (IN MW) IN THE RIGHT
                   HAND DIGISWITCH.
                2. ENTER THE TIME THAT THE PØWER CHANGE IS TØ ØCCUR
                   (IN MINUTES FRØM NØW) IN THE LEFT HAND DIGISWITCH.
                3. PRESS "ENTER".
                4. THE RUN MAY BE CANCELLED AT ANY TIME BY SELECTING
                   FUNCTIØN 09 AND PRESSING "ENTER".

    03          PRØJECTS TIME TØ PØISØN AND TIME TØ RESTART FØLLØWING
                A CHANGE IN PØWER LEVEL TØ X MEGAWATTS ØCCURRING T
                MINUTES FRØM NØW. THE REQUIRED INPUT IS THE SAME AS
                FØR FUNCTIØN 02 ABØVE.

    04          NØISE ANALYSIS PSD PRØGRAM

    05          DISPLAY CØNTENTS AT MEMØRY ADD. SHØWN IN L.H. DIGISWITCH

    06          SET TIME, SHØWN IN R.H. DIGISWITCH.

    07          PLØT PØWER-SPECTRAL-DENSITY

    08          MEASURE TIME ØF FLIGHT
```

Table 11. (continued)

FUNCTIØN NØ.	PURPØSE
26	PRØJECTS TIME TØ PØISØN AND TIME TØ RESTART FØR A CLEAN CØRE. A PØWER PRØFILE CØNSISTING ØF FIVE PØWER LEVELS AND CØRRESPØNDING TIME PERIØD AT EACH PØWER LEVEL ARE REQUIRED. 1. ENTER THE FIRST PØWER LEVEL (IN MW) IN THE RIGHT HAND DIGISWITCH. 2. ENTER THE TIME PERIØD FØR THIS PØWER LEVEL IN THE LEFT HAND DIGISWITCH, IN HØURS MULTIPLIED BY 10. EXAMPLE: 3.5 HØURS IS ENTERED AS 35. 3. PRESS "ENTER". 4. CØNTINUE ENTERING THE PØWER LEVELS AND CØRRESPØNDING TIME PERIØDS, PRESSING "ENTER" AFTER EACH PAIR IS ENTERED, AS IN STEPS 2, 3, AND 4 ABØVE, UNTIL FIVE PØWER LEVELS AND TIMES HAVE BEEN ENTERED. 5. AFTER THE FINAL ENTRY, A PRINTØUT WILL DISPLAY THE CØMPLETE PRØFILE REQUESTED. 6. THE RUN MAY BE CANCELLED AT ANY TIME BY SELECTING FUNCTIØN 09 AND PRESSING "ENTER".
27	PRØJECTS TIME TØ PØISØN AND TIME TØ RESTART BEGINNING WITH PRESENT CØRE CØNDITIØNS. A PØWER PRØFILE CØNSISTING ØF FIVE PØWER LEVELS AND CØRRESPØNDING TIME PERIØD AT EACH PØWER LEVEL ARE REQUIRED. THE INPUT IS THE SAME AS FØR FUNCTIØN 26 ABØVE. THE RUN MAY BE CANCELLED AT ANY TIME BY SELECTING FUNCTIØN 09 AND PRESSING "ENTER".
28	DRPØWC SPECIAL
29	DISPLAY EVERY SECØND THE PØINT VALUE SELECTED FRØM THE TABLE UNDER FUNCT. 00. (KILL WITH FUNCT. 09.
30-99	NØT IN SERVICE.

Table 11. (continued)

FUNCTIØN NØ.	PURPØSE
09	ABØRT FUNCTIØNS FØR: (A) TIME ØF FLIGHT (B) PSD DATA TYPEØUT AND PLØT (C) ALL PLØTTING RØUTINES (D) TIME TØ PØISØN (FCNS 02, 03, 26, 27)
10	PRØJECT END ØF CØRE LIFE (SHUTDØWN)
11	PLØT CALIB. RØUTINE FØR FCNS 7 AND 17
12	AVG. CØRE IN AND ØUTLET TEMPS.,FLØWS AND HEAT PØWER. 1. SET SECØNDS (NØ. ØF SCANS) IN R.H. DIGISWITCH. 2. PRESS "ENTER".
13	CALIB. FUNCTIØN FØR DATA PLØTS - FCNS 14 AND 15
14	PLØT REACTØR DATA - LIVE. (SEE INSTRUCTIØNS AT RIGHT)
15	PLØT REACTØR DATA - REPLAY FRØM SAVED DATA BUFFER. (SEE INSTRUCTIØNS AT RIGHT)
16	SAVE CURRENT REACTØR DATA IN DATA BUFFER FØR REPLAY.
17	PLØT FRØM DRUM A 1. SET X INCREMENT IN L.H. DIGISWITCH (VARIABLE 0-10). 2. SET CHANNEL TØ PLØT IN R.H. DIGISWITCH. 8 = CHANNEL 8 ØNLY 9 = CHANNEL 9 ØNLY ANY ØTHER NUMBER PLØTS ALL DATA. 3. PRESS "ENTER".
18	RIBBØN CØLØR CHANGE (+ IN RH DIGISW FØR BLK, - FØR RED)
19	DIGITAL REACTIVITY (INVERSE KINETICS CALC.)
20	ABØRT DIGITAL REACTIVITY CALC. AND RØDJØG FUNCTIØN.
21	CALIBRATE RECØRDER FØR DIGITAL REACTIVITY DISPLAY
22	NØT IN SERVICE
23	FIXES CURRENT HYDRAULIC READINGS AS STANDARD FØR ALARMING ALL FUTURE READINGS. (SET +99 IN R.H. DIGISWITCH).
24	READ AND TYPE ALL HYDRAULIC DATA.
25	TEMPERATURE BULB CALIBRATIØN AID.

REFERENCES

1. H. P. Danforth, "Reactivity Balance as an Anomaly Detector," Nucl. Safety 10(5), 411-14 (Sept.-Oct. 1969).

2. R. D. Cheverton and T. M. Sims, HFIR Core Nuclear Design, ORNL-4621 (July 1971).

3. G. R. Owens, Development of an Automatic Restart System for the HFIR Digital Control and Data Acquisition System, ORNL-TM-2787 (March 1970).

4. J. B. Bullock and H. P. Danforth, Annunciator Scan Program for the HFIR Digital Control and Data Acquisition System, ORNL-TM-2837 (Feb. 1970).

5. F. T. Binford and E. N. Cramer, The High Flux Isotope Reactor, ORNL-3572 (May 1964).

6. L. C. Oakes, A Second Generation of Reactor Control Systems as Applied to the High Flux Isotope Reactor, ORNL-TM-1259 (Sept. 1965).

7. J. B. Bullock et al., Reactor On-Line Computer Control Development at the HFIR, Vol. 2: Program Listings, Summaries and Logic Diagrams, ORNL-TM-3679, Vol. 2 (Feb. 1972).

8. G. R. Owens, Development of a Mass Memory Program Editor for the HFIR Digital Control and Data Acquisition System, ORNL-TM-2788 (Feb. 1970).

9. B. R. Lawrence and J. B. Bullock, An Algorithm for Calculating Reactivity from Neutron Flux Using An On-Line Digital Computer, ORNL-TM-1473 (April 1970).

10. D. N. Fry, On-Line Calibration of HFIR Control Rods Using the Rod Oscillation Technique, ORNL-TM-1961 (Sept. 1967).

11. B. R. Lawrence et al., A Mathematical Model for the High Flux Isotope Reactor Reactivity Calculation, ORNL-TM-1472 (Oct. 1966).

12. P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962.

13. A. Pearson, "Computer Control on Canadian Nuclear Reactors," Proc. Symp. Appl. On-Line Computers to Nucl. Reactors, Sandefjord, Norway, Sept. 2-8, 1968, European Nuclear Energy Agency.

14. J. W. Allen and W. H. Sides, Jr., Study of Reactivity Balance During Startup of the HFIR, ORNL-TM-3769 (June 1972).

15. E. R. Volk et al., "Installation, Checkout, Testing, and Planned Operation of the Malfunction Detection Analyzer for the Enrico Fermi Atomic Reactor," ANS Trans. Conf. Reactor Operating Experience 12 (Suppl.), 35 (October 1969).

16. J. B. Bullock and H. P. Danforth, "The Application of an On-Line Digital Computer to the Control System of the High Flux Isotope Reactor (HFIR)," IEEE Trans. Nucl. Sci. NS-16(1), 222-26 (February 1969).

17. F. T. Binford et al., The High Flux Isotope Reactor Accident Analysis, ORNL-3573 (April 1967).

4. APPENDIX

4.1 Specification SCS-1: HFIR On-Line Data-Acquisition and Process-Control Computer System

INSTRUMENTATION AND CONTROLS DIVISION

OAK RIDGE NATIONAL LABORATORY
Operated by
Union Carbide Corporation
for the
U. S. Atomic Energy Commission
Oak Ridge, Tennessee

HFIR On-Line Data-Acquisition and Process-Control Computer System

1. SCOPE

1.1 This specification covers the performance and component requirements for an on-line data-acquisition and process-control computer system. The system is to be used to complement the existing reactor control instrumentation system at the High Flux Isotope Reactor (HFIR) at Oak Ridge National Laboratory.

1.2 Under this specification the Seller shall engineer, construct, and test the system at the Seller's premises and shall supervise the installation of the system at the HFIR site. The Seller shall supply engineers and other technical personnel to assist the Company during startup and initial operation of the system, provide the initial programs for operation of the system, and train Company personnel in programming and operating techniques for the system.

2. APPLICABLE DRAWINGS AND DOCUMENTS

Manufacturer's Test Program Listing (attached).

3. REQUIREMENTS

3.1 Design

3.1.1 General Description of the System

a. As a minimum, the system shall consist of input-signal conditioning equipment, input-signal scanning equipment, an analog-to-digital converter, a digital-to-

analog converter, a computer with peripheral accessories, an operator console, a programmer panel, one input-output electric typewriter (Teletype model 35 or approved equal), one paper-tape punch and reader, one message typewriter, and equipment for producing analog and digital output signals.

b. The system shall perform three functions:

(1) On-line computing--calculate information in real time for reactor control.

(2) Data handling--scan, digitize, and convert to engineering units data from sensors, and log the data on a typewriter when called for.

(3) Time shared computing--perform general purpose computing on accumulated experimental data or execute other scientific programs in conjunction with the on-line process control program.

3.1.2 System Input Signals

a. The system shall handle a minimum of 12 low-level analog signals (0-50 mv dc) utilizing at least 11 bits plus sign with an accuracy of ±0.1% of full scale. The system shall be expandable to handle 125 low-level analog signals by the addition of modules and mounting racks.

b. The system shall handle a minimum of 12 high-level analog signals (0-5 v dc) utilizing at least 11 bits plus sign with an accuracy of ±0.1% of full scale. The system shall be expandable to handle 125 high-level analog signals by the addition of modules and mounting racks.

c. The system shall handle a minimum of 48 digital input signals consisting of sustained contact closures. The system shall be expandable to handle 500 digital input signals by the addition of modules and mounting racks.

3.1.3 System Output Signals

a. At least one low-level analog signal (0-10 mv) shall drive a recorder (not to be supplied by the Seller).

b. A minimum of 6 digital output signals, consisting of single-bit on-off solid-state circuits, shall supply at least 200 milliamperes at 48 v dc to load circuits.

3.1.4 System Operation

a. The analog input signals (with the exception in 3.1.4b below) shall be scanned and digitized at a minimum rate of 200 signals per second.

b. Two of the high-level analog input signals shall be scanned and digitized at a minimum rate of 5000 samples per second for each signal.

c. The digital input signals (contact sense) shall be scanned at a minimum rate of 5000 contacts per second.

d. The Seller shall demonstrate that the proposed system can compile and execute the attached Manufacturers Test Program (see paragraph 6.1.2). The program may be rewritten (Seller's option) in symbolic assembly language. The proposed system shall compute a value of DKSUM=-0.0766 corresponding to TDAYS=14 days within an execution time of less than 120 minutes.

3.1.5 System Control

a. All functions of the system shall be controllable both automatically and manually.

b. Automatic control shall be by instructions written into the magnetic core memory unit of the computer.

c. The program shall be changeable by three methods: (1) paper tape, (2) the typewriter keyboard, and (3) switches at the programmer panel.

3.1.6 System Failures

a. Failure of any major component of the system shall actuate an alarm on both the operator console and the programmer panel.

b. Diagnostic routines, automatic test routines, and manually controlled tests shall indicate probable impending failure and their location.

c. A momentary electrical power failure of 3 cycles or less (60 cycle power) or power transients of ±10% of 115 v ac shall not alter any information in the computer or disturb the continuity of operation of the system.

d. In the event that the electrical power of 115 v ac decreases by 10% to 103.5 v ac, a sensing circuit shall cause a battery backup power supply to be switched

on. The battery supply (to be supplied by the Seller) shall operate the computer system for a period of at least 3 minutes. If the normal power returns within this 3 minutes, the batteries shall be switched off and recharged (the Seller shall supply the charger).

e. In the event that normal electrical power of 115 v ac has not been restored within 3 minutes, a sensing circuit shall cause the contents of all registers to be stored in the nonvolatile core memory. When normal power is reestablished, the system shall automatically restart the computer system in accordance with predetermined program requirements, and the batteries shall be recharged.

3.2 Components

3.2.1 Computer

a. The computer shall be a general purpose, fully parallel digital computer. The core memory shall have a minimum of 12 K words addressable by instruction, directly and indirectly.

b. The computer shall have a minimum of 16 hardware-implemented levels of priority interrupt.

c. All words in the memory shall be protected, either as individual words or memory blocks of 1-K words, against accidental modification.

d. Parity checks shall be provided for all transfers between registers and into and out of core memory locations. Parity errors shall be indicated on both the operator console and the programmer panel.

e. All analog-to-digital and digital-to-analog converters and all signal conditioning equipment shall be included as an integral part of the computer system.

f. All computer main-frame active components shall be solid-state devices.

3.2.2 Peripheral and Input-Output Equipment

a. A remote operator console shall incorporate, as a minimum, the following items:

(1) A lamp bank shall give a 4-digit plus sign decimal readout of process variables requested by the operator.

(2) A keyboard shall enter data or change data in unguarded locations in the computer memory.

(3) Either a group of at least 16 request switches or a 16-position selector switch (Seller's option) shall enable the operator to request specific variables to be read out on the panel or to request some required action (such as to log data on the typewriter). These switches may (Seller's option) be tied in parallel to one of the priority interrupt lines.

(4) A digital data display shall consist of individual lights to indicate, as a minimum, parity error, watchdog timer run down, and computer malfunction.

(5) One request button shall operate in conjunction with the keyboard to request a readout of the information stored in a specific memory address selected from the keyboard.

b. This console will be located approximately 100 feet from the computer.

c. The programmer panel shall incorporate the various switches, indicators, and displays necessary for programming and maintenance.

d. The system shall be equipped with a real-time clock which shall be set and read under program control and shall interrupt the system operation when the preset time has expired. The clock shall indicate real time in hours, minutes, and seconds.

e. A ("watchdog") or stall timer shall be supplied which shall be set under computer control and shall sound an alarm on the operator console and programmer panel when the timer counts to zero.

f. One paper-tape punch and reader (either separately or as one unit) shall be furnished with a minimum speed of 10 characters per second. This unit may be a part of the input-output electric typewriter (Teletype model 35 or approved equal) specified in 3.2.2g.

g. One Teletype model 35 (or approved equal) input-output electric typewriter shall be furnished with full alphanumeric capability and a minimum speed of 10 characters per second.

h. One message typewriter shall be furnished; it will be placed at the operator console. It shall have full alphanumeric capability and a minimum speed of 10 characters per second.

i. A manually resettable timer shall be furnished which shall indicate days, hours, and minutes to determine the running time of the computer.

3.3 System Reliability

3.3.1 Operating Availability

a. The operating availability of the system, excluding the mechanical components of paper-tape punch and reader devices and typewriters, shall be 99.0% or better. The operating availability shall be determined over a minimum period of 180 days of operation (Continuity of Service Test).

b. The system shall operate continuously without any detectable failures 24 hours per day at all times except when the reactor is shut down for a core change (approximately 2 hours at the close of each 2-week period will be allowed for computer maintenance).

3.3.2 Mean Time Between Failures

a. The system shall operate with a minimum of 1500 hours mean time between failures (MTBF). The MTBF shall be determined at the end of the first 180 days of operation (Continuity of Service Test).

b. The system shall be considered as having failed if (1) any error in performance is detected, (2) any analog input signal is inaccurate by more than ±0.1% of full scale, or (3) if the total system error is greater than ±0.6% of full scale (0-10 mv). For this requirement only, the system shall include only input amplifiers, noise rejection equipment, analog-to-digital and digital-to-analog converters, the computer, timing devices, signal conditioning circuits, and output signals to terminal points.

3.4 Construction, Installation, and Assembly

3.4.1 Construction

a. All circuits shall be constructed as modular plug-in units. It shall not be necessary to alter the modules or to make wiring connections to remove or replace the modules. All test points shall be accessible without removal of the card or unit from the cabinet.

b. The Seller shall supply a cabinet (or cabinets) to contain the computer, signal converters, signal conditioning circuits, operator console, and all other equipment used for connecting the system to the external instrumentation and peripheral equipment.

c. Only field wiring connections shall be required for connection of electrical power, input leads from sensing elements, output leads to peripheral devices, and for interconnections between cabinets.

d. Stands shall be supplied for the operator console and reader-punch units.

e. All consoles, panels, and cabinets supplied shall be constructed of heavy-gauge steel (minimum 16 gauge), consistent with good industrial control-panel practice.

3.4.2 Installation

a. The system will be installed by the Company under the supervision of the Seller.

b. The Seller shall supervise the connection of input and output signal leads to the system.

3.4.3 Assembly

a. The Seller shall assemble all components of the system.

b. The Seller shall supply all terminal connections and panels for connection of signal leads to the system and all other equipment required for conditioning and modifying the input signals.

3.5 Software

The Seller shall provide at least the software described in the following paragraphs.

3.5.1 A Fortran II (or approved equal) language which shall:

a. Call Fortran subroutines;

b. Operate in an interruptible environment; branching into Fortran subroutines on the occurrence of one or more specified interrupts shall be programmable in Fortran language;

c. Transfer data to and from core memory;

d. Call subroutines written in symbolic assembly language;

e. Compile a mixture of Fortran and symbolic assembly language instructions;

f. Store data from analog-to-digital converters and digital inputs into core locations specified by Fortran variable names;

g. Output information from locations in core locations specified by Fortran variable names to specified output lines;

h. Control analog and digital sampling rates;

i. Supply standard Fortran library functions such as sine, cosine, natural logarithm, exponential, square root, raising to arbitrary (floating point) powers, etc;

j. Store the reading of the real-time clock and watchdog timer in core memory locations specified by Fortran variable names.

3.5.2 A symbolic assembly language shall perform all functions required by this specification and shall provide for double-precision computation.

3.5.3 An executive or monitor routine (as required by the Seller's proposed system) shall compile a subsidiary Fortran program on a time-shared basis with the on-line process control program.

3.5.4 Test programs shall check for proper operation of the proposed system and shall aid in troubleshooting. Diagnostic routines shall be applicable to on-line use with the process control program.

3.5.5 A memory dump routine shall dump the contents of specified portions of the core memory onto paper tape and the typewriter.

3.5.6 A routine shall provide a listing of variable names and their addresses.

3.5.7 A routine shall indicate the unused locations in the core memory.

3.5.8 The on-line process control program shall be the joint responsibility of the Seller and the Company. For the purpose of this specification the Seller shall provide one-man month of programming effort.

3.6 Environmental Conditions

The room where the system will be installed is air conditioned. The system shall meet or exceed all requirements of this specification while operating in an ambient temperature from 10°C to 45°C and in relative humidity to 90%.

3.7 Training of Company Personnel

3.7.1 The Seller shall furnish training facilities and shall train three Company personnel at the Seller's premises in all phases of the system, including programming and theory of operation.

3.7.2 The Seller shall train a minimum of three and a maximum of six Company operators at the installation site to operate the system.

4. INSPECTIONS AND TEST PROCEDURES

4.1 Preshipment Tests

The system shall be inspected and tested at the Seller's plant to show full compliance with this specification with the exception of paragraph 3.3.2. The system shall operate continuously for 72 hours without failure prior to shipment. The Seller shall notify the Company 10 days prior to commencement of testing. Company representatives will witness and participate in these tests. Certified inspection and test results shall be furnished to and approved by the Company prior to shipment.

4.2 Final Tests

4.2.1 The system will be inspected and tested in accordance with the tests specified in paragraph 4.1 by the Company at the installation site after the system has been installed.

4.2.2 If the results of tests specified in paragraph 4.2.1 are not accepted by the Company, the system shall be repaired or replaced by the Seller and a complete test of the system will be again made by the Company.

4.2.3 If the results of the Continuity of Service test are not as specified in paragraphs 3.3.1 and 3.3.2, the system shall be repaired or replaced by the Seller, and another 180-day Continuity-of-Service test will be made by the Company.

4.2.4 Final acceptance will be upon written approval by the Company of the results of tests specified in paragraphs 3.3.1, 3.3.2, 4.2.1, and 4.2.3, and upon written certification by the Seller that the system is ready for operation.

5. PREPARATION FOR DELIVERY

There shall be marking on all packages and shipping containers to permit identification of the materials or units contained therein. All shipping containers shall have a separate packing list enclosed showing the contents of the container, the purchase order number, the serial number, and the name of the Seller. Each shipping container shall be marked on the exterior with

a. The Seller's name and address,

b. The Company purchase order number,

c. The number of units contained therein and the serial numbers.

6. TECHNICAL INFORMATION AND ENGINEERING DRAWINGS

6.1 Information to Be Supplied with Bids

6.1.1 Technical Information

One copy of each of the following information shall be supplied with the Seller's bid:

1. Complete operating instructions for the basic machine,

2. Abstracts of all software to be supplied by the Seller,

3. A machine instruction list with mnemonics, octal equivalents, Fortran equivalents, and detailed definition of the function of each instruction,

4. A list of maintenance options available from the Seller.

6.1.2 Manufacturers Test Program

The Seller shall supply the following information regarding the Manufacturers Test Program.

a. All printed output generated by the system while compiling and executing the test program,

b. Compilation time,

c. Execution time,

d. The number of memory locations used for compiling and for execution.

6.2 Information Required Within Twelve Weeks After Receiving the Award

Four copies of each of the following information shall be furnished by the Seller within 12 weeks after receiving the award:

1. Utility wiring diagram and system interconnection cabling diagram,

2. Outline and mounting drawings of major system components,

3. Layout and installation drawings of the system,

4. Description of system operation.

6.3 Information Required Four Weeks Prior to the Test of the System at the Seller's Premises

6.3.1 Within 4 weeks prior to beginning the preliminary test of the equipment at the Seller's premises, the Seller shall furnish six copies each of the following drawings and engineering information; all drawings and engineering information shall be identified and marked by serial numbers or descriptive titles (or both) that indicate their relationship to the system and shall include drawings, data, manuals, and engineering information required to cover the complete description of the system and instructions for operation and maintenance:

1. Outline, mounting, and dimensional drawings of the major components, including input-output cabinet, computer, magnetic drum, and the operator console,

2. Floor plan drawings of the system, showing the installation,

3. Schematic drawings, showing all electrical components of the system (computer, typewriters, amplifiers, etc.),

4. Block diagrams, such as information flow diagrams of the system which show how the system is connected together,

5. Wiring, cabling, and interconnection diagrams, showing connections between major components,

6. Installation, operation, and maintenance instructions,

7. Parts lists, giving the name of the part and the manufacturer's part numbers,

8. Recommended spare parts list, including original manufacturer's type and model numbers,

9. Equipment performance specifications,

10. Program and computer instruction manuals.

6.3.2 Delivery of equipment shall not be authorized until the Company has approved in writing the engineering information and drawings.

6.4 Information and Drawings Required Four Weeks After Approval of Final Tests

6.4.1 Not later than 4 weeks after final acceptance (paragraph 4.2.4), the Seller shall correct all drawings and information required by paragraph 6.3.1. These drawings and information shall be corrected to the date of final acceptance.

6.4.2 The Seller shall supply 6 copies each of all corrected information and drawings.

```
      PRØGRAM TEST
      DIMENSIØN D11(1),D12(1),D13(1)
      DIMENSIØN B(11)
      CØMMØNF3,F7,F8,SA5S3,SA10S3,SF5S3,
     1SAIS3,SAXES3,SAPMS3,SASMS3,SALIS7,
     2SALIS8,SNAS7,SNAS8,AMDAI,AMDAXE,
     3AMDAPM,GI,GXE,GPM,GSM,ALPFP,ANBE7,
     4ANBE8,GAM,GAMA,AMA,CA,HFC,AA,AMC,CC,
     5TIN,TAUC,GA,GHC,GC,V3
    1 FØRMAT(2E13.5)
      SA5S3=6.836E-22
      SA53=6.836E-22
      USF53=1.196E-21
      SF5S3=5.770E-22
      FFF53=7.000E-01
      SFF53=5.770E-22
      SA10S3=4.017E-21
      SA103=4.017E-21
      SAIS3=0.0
      AMDAI=2.900E-05
      GI=6.100E-02
      SAXES3=2.540E-18
      SAXE3=2.540E-18
      AMDAXE=2.100E-05
      GXE=3.000E-03
      SAPMS3=2.500E-26
      AMDAPM=3.560E-06
      GPM=1.400E-02
      SAPM3=2.500E-26
      SASMS3=4.210E-20
      SASM3=4.210E-20
      GSM=0.0
      ALPFP=3.700E-23
      PCØN=3.100E+16
      PØWER=1.000E+02
      SALIS7= 9.450E-22
      SALIS8= 9.450E-22
      SALI7 = 9.450 E-22
      SALI8 = 9.450 E-22
      SNAS7= 6.000E-26
      SNAS8= 6.000E-26
      S2B7 = 2.000E-25
      S2B8 = 2.000E-25
      ANBE7 = 1.950 E +23
      ANBE8 = 1.950 E +23
      V3 = 6.200 E +04
      V7 = 8.100 E+04
      V8 = 2.900E+05
```

```
      ALPHA3 =-1.000E-05
      ALPHA4 = - 9.700E-05
      BETA3 = 0.0
      BETA8 = 0.0
      GAM3 = 5.000E-01
      GAM8 = 4.000E-01
      AMA=3.860E-02
      CA=1.000E+00
      HFC = 1.160 E +00
      AA=1.000E+00
      AMC = 5.590 E-02
      CC = 1.000 E +00
      GAM = 1.436 E -17
      GAMA=9.750E-01
      TIN = 1.200 E +02
      TAUC = 3.902 E - 02
      DFS = 2.750 E-03
      TFIN = 1.210 E+06
      DT=3.0E+1
      NI=1
      TØ = 0.0
      S=0.0
      SA53 =SA53/ FFF53
      SA5S3=SA5S3/FFF53
      SF5S3=SFF53/FFF53
      GA=AMA*CA
      GHC=HFC*AA
      GC=AMC*CC
      NC=0
      T=TØ
      B(1) = 3.890E+20
      B(2) = 2.060E+18
      GØ TØ 104
 104  B(3) =0.0
      B(4) =0.
      B(5) =0.
      B(6) =0.
      B(7) =0.
      B(8) =0.
      B(9) =0.
      B(10) = 2.380 E +02
      B(11) =1.540E+02
      GØ TØ 105
 105  DKSUM=0.0
      TDAYS=0.0
C     PRINT 1,TDAYS,DKSUM
      WRITE(4,1)TDAYS,DKSUM
    6 F3 = PCØN*FFF53*PØWER/(V3*B(1)*SFF53)
      GØ TØ 106
```

```
106  F7 = F3 / ( GAM3 + BETA3 *S )
     F8=(BETA8*S+GAM8)*F7
     GØ TØ 100
100  W1=SA5S3*F3*SA53-SA5S3*F3*USF53-GXE*SF5S3*F3*SAXE3
    1-GSM*SF5S3*F3*SASM3-ALPFP*F3*SF5S3- GPM*SF5S3*F3*SAPM3
     W2=SA10S3*F3* SA103
     W3=AMDAI*SAXE3
     W4=AMDAXE*SAXE3
     W5=SAXES3*F3*SAXE3
     GØ TØ 101
101  W6=AMDAPM*SASM3-SAPM3*AMDAPM
     W7=SASMS3*F3*SASM3
     W8=V7*SALIS7*F7*SALI7
     W9=V7*SNA S7*F7*SALI7
     W10=F8*SALIS8*SALI8*V8
     GØ TØ 103
 103 W11=V8*SNA S8*F8*SALI8
     W12=USF53*V3
     W13=S2B7*V7
     W14=S2B8*V8
     W19=SAPMS3*B(5)*F3*SAPM3
     GØ TØ 108
108  DK=(    W1*B(1)+    W2*B(2)-W3*B(3)+(W4+    W5)
    1*B(4)-W6*B(5)-W19          +    W7*B(6))*V3*F3**2
     DK=DK+F7**2*(B(8)*W8-ANBE7*W9)
     DK=DK+F8**2*(B(9)*W10-ANBE8*W11)
     D11= W14*F8**2
     D11= D11*ANBE8
     D12= W13*F7**2
     D12= D12*ANBE7
     D13= W12*F3**2
     D13= B(1)*D13
     DK=DK/(D13+ D12+ D11)
     DK=DK*DT
     W15=0.0
     W16=0.0
     W17=0.0
     W18=0.0
     DKSUM=DKSUM+DK
     DS=DK/DFS
     TM=T
     T=T+DT
     S=S+DS
     IF(NC-5760)9,11,11
  11 TDAYS=TM/8.64E+4
     WRITE(4,1)TDAYS,DKSUM
     NC=0
```

```
    9 CØNTINUE
      NC=NC+1
      IF(T-TFIN)7,8,8
    7 CALL DIFEQ(TM,T,B,11,NI)
      GØ TØ 6
    8 CØNTINUE
      END
      SUBRØUTINE EQUA(T,B,D)
      DIMENSIØN B(11),D(11)
      CØMMØN F3,F7,F8,SA5S 3,SA10S 3,SF5S 3,
     1SAIS 3,SAXES 3,SAPMS 3,SASMS 3,SALIS 7,
     2SALIS 8,SNAS 7,SNAS 8,AMDAI,AMDAXE,
     3AMDAPM,GI,GXE,GPM,GSM,ALPFP,ANBE7,
     4ANBE8,GAM,GAMA,AMA,CA,HFC,AA,AMC,CC,
     5TIN,TAUC,GA,GHC,GC,V3
      D(1)=-F3*SA5S 3*B(1)
      D(2)=-F3*SA10S 3*B(2)
      D(3)=GI*F3*SF5S 3*B(1)-(AMDAI+F3*
     1SAIS 3)*B(3)
      D(4)=GXE*F3*SF5S 3*B(1)+AMDAI*B(3)
     1-(AMDAXE+F3*SAXES 3)*B(4)
      D(5)=GPM*F3*SF5S 3*B(1)-(AMDAPM+F3*
     1SAPMS 3)*B(5)
      D(6)=GSM*F3*SF5S 3*B(1)+AMDAPM*B(5)
     1-SASMS 3*F3*B(6)
      D(7)=ALPFP*F3*SF5S 3*B(1)
      D(8)= -F7*SALIS7*B(8) + F7*SNAS7*ANBE7
      D(9) =-F8*SALIS8*B(9)+ F8*SNAS8*ANBE8
      D(10)=0.0
      D(11)=0.0
      RETURN
      END
      SUBRØUTINE DIFEQ(XA,XE,Y,N,I)
      DIMENSIØN Y(11),YI(11),AM1(11),AM2(11),AM3(11),AM4(11)
      FFFFF = I
      FI = FFFFF
C     FI=I
      H=(XE-XA)/FI
      DØ 1 J=1,I
      CALL EQUA(XA,Y,AM1)
      XA=XA+H*0.5
      DØ 2 K=1,N
    2 YI(K)=Y(K)+AM1(K)*H*0.5
      CALL EQUA(XA,YI,AM2)
      DØ 3 K=1,N
    3 YI(K)=Y(K)+AM2(K)*H*0.5
      CALL EQUA(XA,YI,AM3)
      XA=XA+H*0.5
      DØ 4 K=1,N
```

```
    4 YI(K)=Y(K)+AM3(K)*H
      CALL EQUA(XA,YI,AM4)
      DØ 1 K=1,N
    1 Y(K)=H*(AM1(K)+AM4(K)+2.0*(AM2(K)+AM3(K)))/6.0
     1+Y(K)
      RETURN
      END
NEXT!
```

4.2 SCAN Programs

4.2.1 SCAN34

The SCAN34 program (Fig. 34) establishes the scan frequency by calling itself to run again after a 1-sec delay. A check is made to ensure that, if the scan has been turned off during the cycle, there will be no attempt to move the reactor control rods without proper input data. Next, a check is made of a low-level scan fail flag. This flag is set if all low-level points are out of limits, indicating a major system malfunction. The result of such a malfunction detection is a complete automatic restart sequence. Experience has shown that this technique has been effective, since most failures have been temporary, e.g., an ac power transient to transducers, etc.

A set of descriptive words is assigned to each analog input point. This table was originally stored on the drum and read into the core as needed by the scan routines. However, operating experience resulted in the adoption of a core resident version. (See DUMBUL in Vol. 2.[7]) There are eight words per input point, or a total of 256 descriptive words. The data format is:

Word	
1	Point number and active flag
2	Multiplexer address and gain index
3	High-limit value
4	Low-limit value
5	Pointer to conversion routine
6	Last alarmed value
7	Point index number
8	Deadband and significant change limit

The SCAN34 program reads in the low-level descriptive words (128) and checks for points to be scanned. Active points have their descriptive word core address passed to the low-level A-D driver program.

The low-level signals are processed through a 200-pulse/sec reed relay multiplexer to an amplifier with a gain of 100 and then to an eleven bits plus sign A-D converter. The function of this driver is to obtain the value of the requested analog point in millivolts and pass it to a conversion routine to obtain engineering values. This is accomplished in two steps:

ORNL DWG NO. 72-866

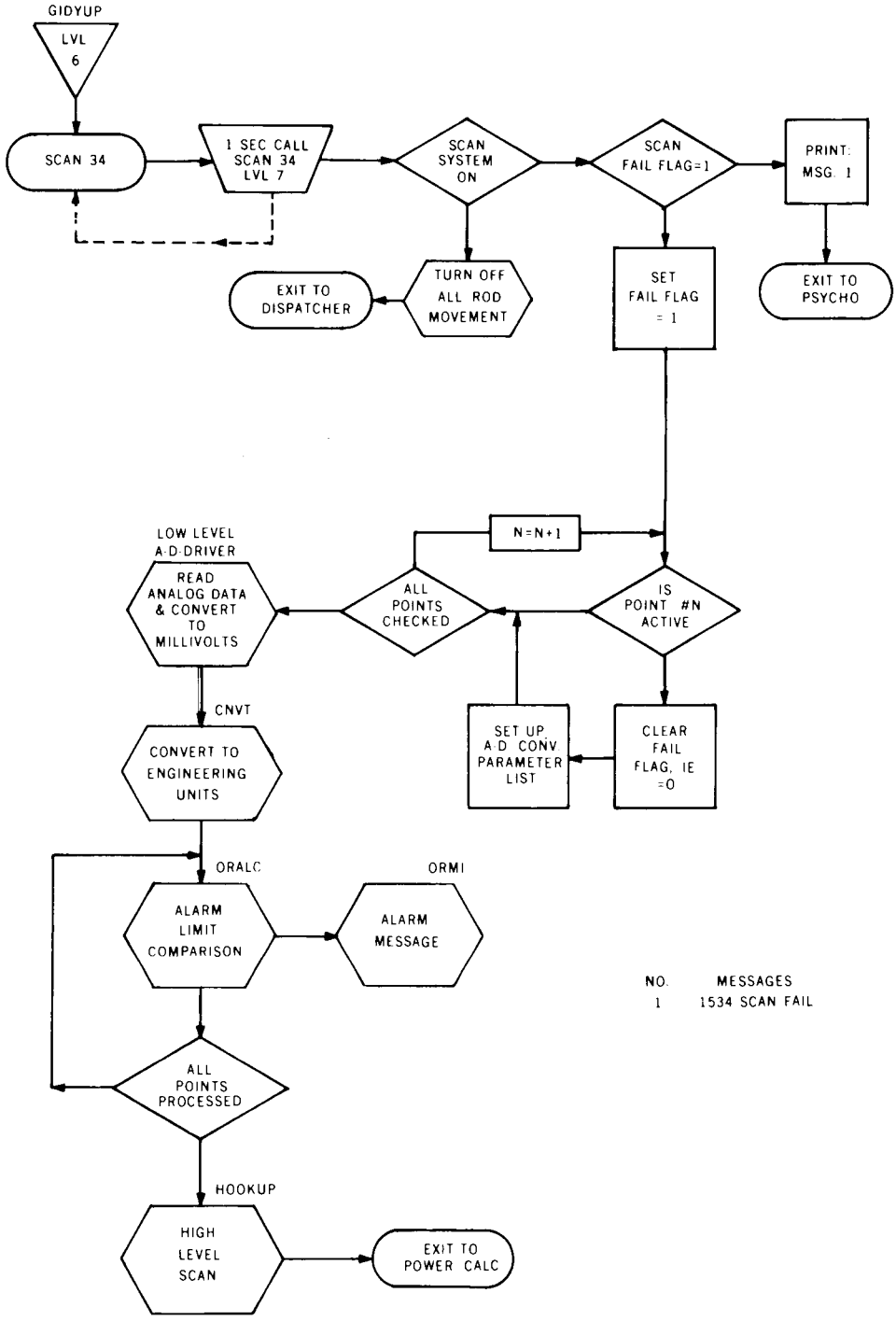


Fig. 34. Logic block diagram of program SCAN34.

1. The value of the last multiplexed point is read into the computer, and the requested point is multiplexed.
2. A-D conversion completion generates an interrupt; the requested point data are read into the memory and the next point is multiplexed. The values are then converted to millivolts.

The A-D Driver (Fig. 35) consists of two parts: the initiator, entered by a monitor request; and the continuator, entered from the interrupt processor. A system diagnostic clock is used to generate an error message if excessive time is taken between steps (1) and (2) above. This clock is turned on by the initiator portion of the driver and turned off by the interrupt response. The parameter list associated with the initiating call includes request priority, completion address and priority, logical unit number, number of input points, and starting address of an input-output data list. The data list, in turn, contains a word specifying gain and conversion units, descriptive word address, and a buffer to store input values. The initiator portion of the driver picks out the proper parameters from the data list to pass to the continuator and checks for outstanding requests.

The continuator sets up gain and function codes and reads in the requested data. The low-level A-D converter inputs counts into the upper 12 bits of the 16-bit A-register: thus, a right shift of four places is required to obtain the actual count. The input value (counts) is then converted to millivolts times gain and passed to the CNVT program for conversion to engineering units, e.g., pressure, temperature, etc.

Several error checks are made in both the A-D driver and the conversion routine. Errors such as input-output rejects, A-D converter failure, or converter input overrange are indicated by special codes which are passed to the alarm limit comparator.

4.2.2 CNVT

This program converts millivolt values to engineering units. The millivolt values are passed to this program from both the low-level and high-level analog input programs via the computer A-register. The Q-register contains the core address of the point descriptive words for the point value being converted. One of the descriptive words

ORNL DWG NO. 72-865

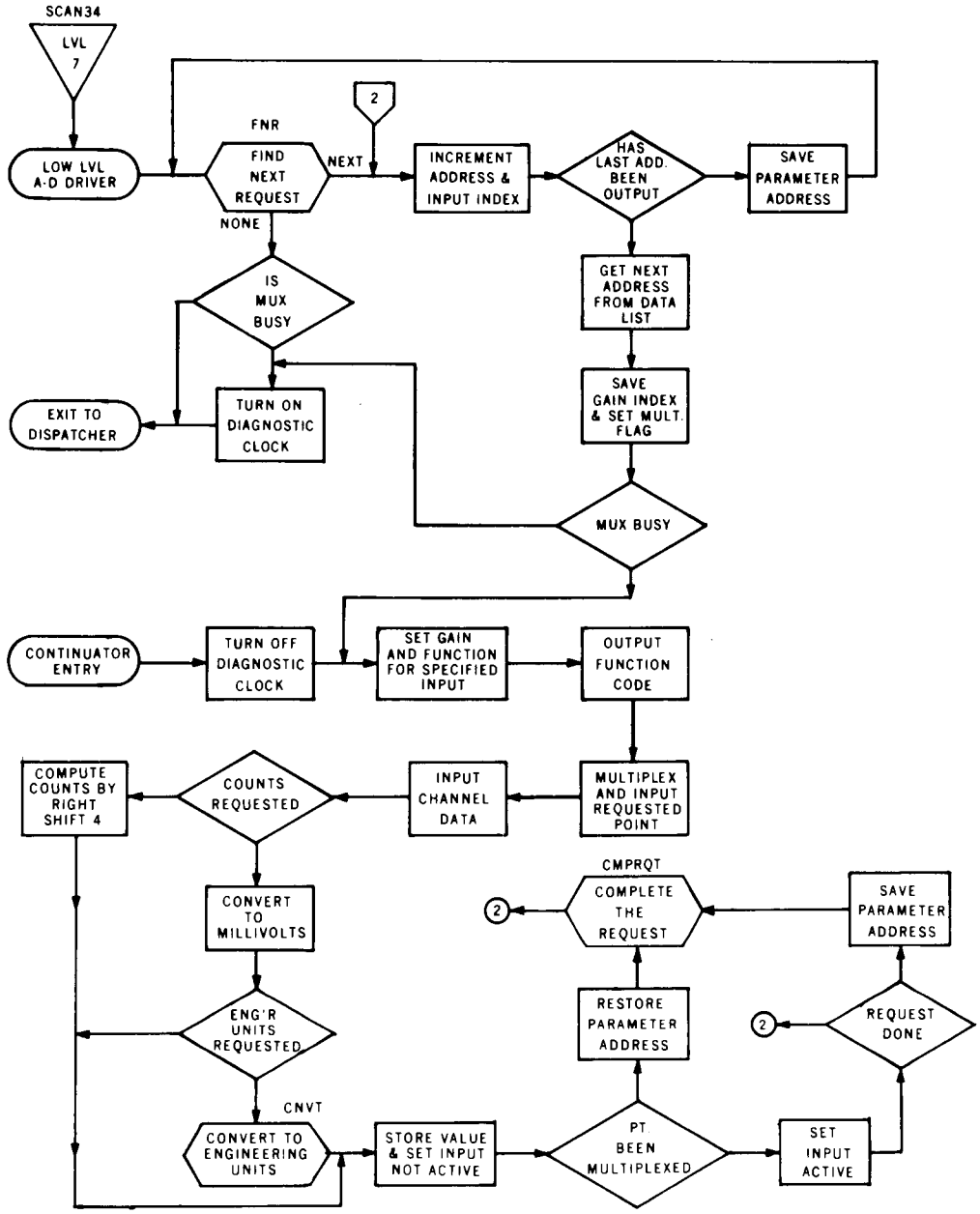


Fig. 35. Logic block diagram of program A-D driver.

contains two indices, one pointing to a particular conversion routine, and the other pointing to a particular set of constants for the chosen conversion equation. This method allows several process variables to be converted using relatively few conversion routines, e.g., $Ax^2 + Bx + C$, $Ax + B$, $A\sqrt{x} + B$, and sets of constants A, B, C.

The conversion routine index is also used as a pointer to a list of allowable maximum and minimum transducer millivolt limits. If the input value is within limits, CNVT returns to the calling routine, with the engineering value in the A-register and zero in the Q-register. An error causes a code corresponding to the type of error to be passed in the Q-register. Thus, a check to see if Q is zero assures proper utilization of the data passed in the A-register. After conversion, the engineering value or error code is passed to the alarm limit comparator, ORALC, by the scan control program.

4.2.3 ORALC

The alarm limit comparator (ALC) program (Fig. 36) has two functions: (1) interpretation of error codes passed to it by the A-D driver or conversion routine, and (2) if no error exists, to check operational limits by comparing the input value with predefined or calculated limits for that point. If errors or alarms are found, special codes are passed to a message initiator program, ORMI, which cause the appropriate message printout. Similarly, if a point returns to a normal condition, a message is also initiated.

Upon entry, the ORALC program sets up an internal ALC field index table to store parameters associated with a particular point being checked. The table is made up of eleven words.

ALC FIELD INDEX TABLE

Point Number
Error Field
Alarm Point Indicator
Dummy
Alarm State Indicator
Deadband Range
High-Limit Value
Low-Limit Value
Message Cutout Indicator
Significant Change Limit
Last Alarmed Value

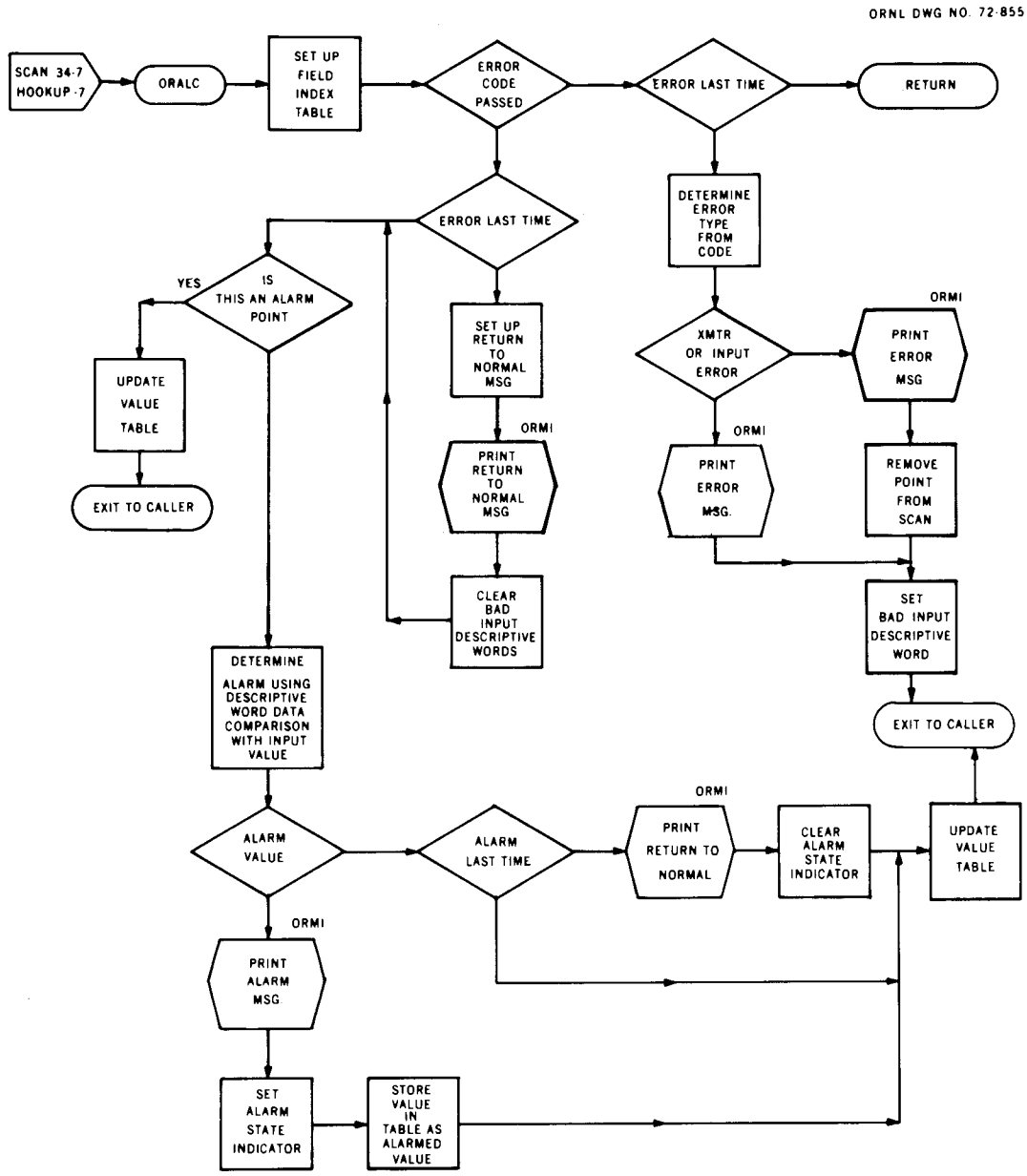


Fig. 36. Logic block diagram of program ORALC.

After setting up a portion of this table, the program proceeds with error code checking. Several error codes may be passed indicating:

a. bad input--a bad value is used in calculating a point value,

b. overrange--input is too high for A-D converter,

c. underrange--input is too low for A-D converter,

d. transmitter overrange--input exceeds field transmitter range,

e. transmitter underrange--input less than field transmitter range,

f. input error--an error in the A-D driver program.

Several cross checks are made to ensure that repeated typeouts of the same message do not occur if the error condition persists. A check is also made to see if the point has returned to normal. If an error condition exists, no alarm limit checking is made, and the program returns to the caller without changing the contents of the value table. This value table is a permanent core resident table that contains the current value of each scanned point in engineering units. All programs that use these system parameters do so by referring to the value table. Therefore, if an error occurs, the value table will contain the last good value that the alarm limit comparator placed into it.

After successful completion of error checking, the program proceeds with alarm limit checks, if desired by the system programmer. The point descriptive words contain an alarm point indicator, and, if this word is not set, the alarm limit checks will not be made. Several alarm limit checks are made, with some of the limit value, e.g., deadband and significant change limits, being calculated by the ORALC program. The alarm checks are

1. High Operating Alarm:

   Normal alarm value ≥ high limit

   Deadband alarm value ≥ high limit - deadband

   Significant change value ≥ last alarmed value + significant change

   value ≤ last alarmed value - significant change

2. Low Operating Alarm:

   Normal alarm value ≤ low limit

   Deadband alarm value ≤ low limit + deadband

   Significant change value ≤ last value - significant change

   value ≥ last value + significant change

As in the error checking portion, return-to-normal and persistent out-of-limits conditions are detected. If an alarm condition occurs, the appropriate code is passed to the message initiator. The current value is stored in both the value table and the descriptive words so that it may be used for significant change checks in the next scan cycle.

4.2.4 ORMI

The function of this program is to initiate the output of all messages associated with both the low-level and high-level analog scan. Error messages, alarm messages, and return-to-normal messages are set up as a result of special codes and parameters passed to this program from the alarm limit comparator. The messages constructed by ORMI are output on a Selectric typewriter in the reactor control room. Alarms and errors are printed in red; return-to-normal messages are black. If the typewriter fails, the messages are transferred to the teletypewriter in the computer room, which has no red-black option available.

An area of mass memory is set aside for point descriptions, a 19-word block containing an alphanumeric description of each analog point; e.g., 07 DEM POWER 2 is the description of analog input point No. 7, demand power No. 2. These are not the same as the eight-word blocks of point descriptive words previously discussed which contain packed fields of input point characteristics. A table of error and alarm messages is associated with the ORMI program, e.g., LO OPER. ALARM. This program merely picks the proper alarm message, using the code passed to it as an index, and places it into a message buffer. Next it reads in the appropriate point description from the drum into the buffer. The current time from the time-of-day package is also placed into the buffer as "hour-minute". A call is then made to the monitor to print the 72-character buffer. A typical printout of alarm and return to normal is:

***	1237	07 DEM POWER 2	LO OPER. ALARM
RTN	1238	07 DEM POWER 2	

4.2.5 HOOKUP

The function of this program is similar to that of SCAN34, i.e., scan control. The HOOKUP program (Fig. 37) is used for the high-level scan and determines the sequence and number of analog points to be read in. Since it is a part of the scan cycle, it is entered once each second from the SCAN34 program.

The point descriptive words for the high-level inputs are brought into the core memory from the drum. The input channels are sequenced and read in via the RDPT program, analogous to the low-level A-D driver. An error causes an alarm error message to be printed if it is the first time an error has occurred for that particular point. The raw counts are stored in a data table, and after all points have been read in, a conversion and checking loop is entered.

In this loop, each point value is converted to engineering units, using the same CNVT program utilized by the low-level scan. The CNVT program then passes control to the ORALC for alarm limit checking. No error codes are passed from RDPT to CNVT because the error message is initiated and completed in HOOKUP. After all values have been checked and stored in the value table by ORALC, the scan cycle control is passed to POWCAL, a reactor heat-power calculation routine (see Fig. 11).

ORNL DWG NO. 72-816

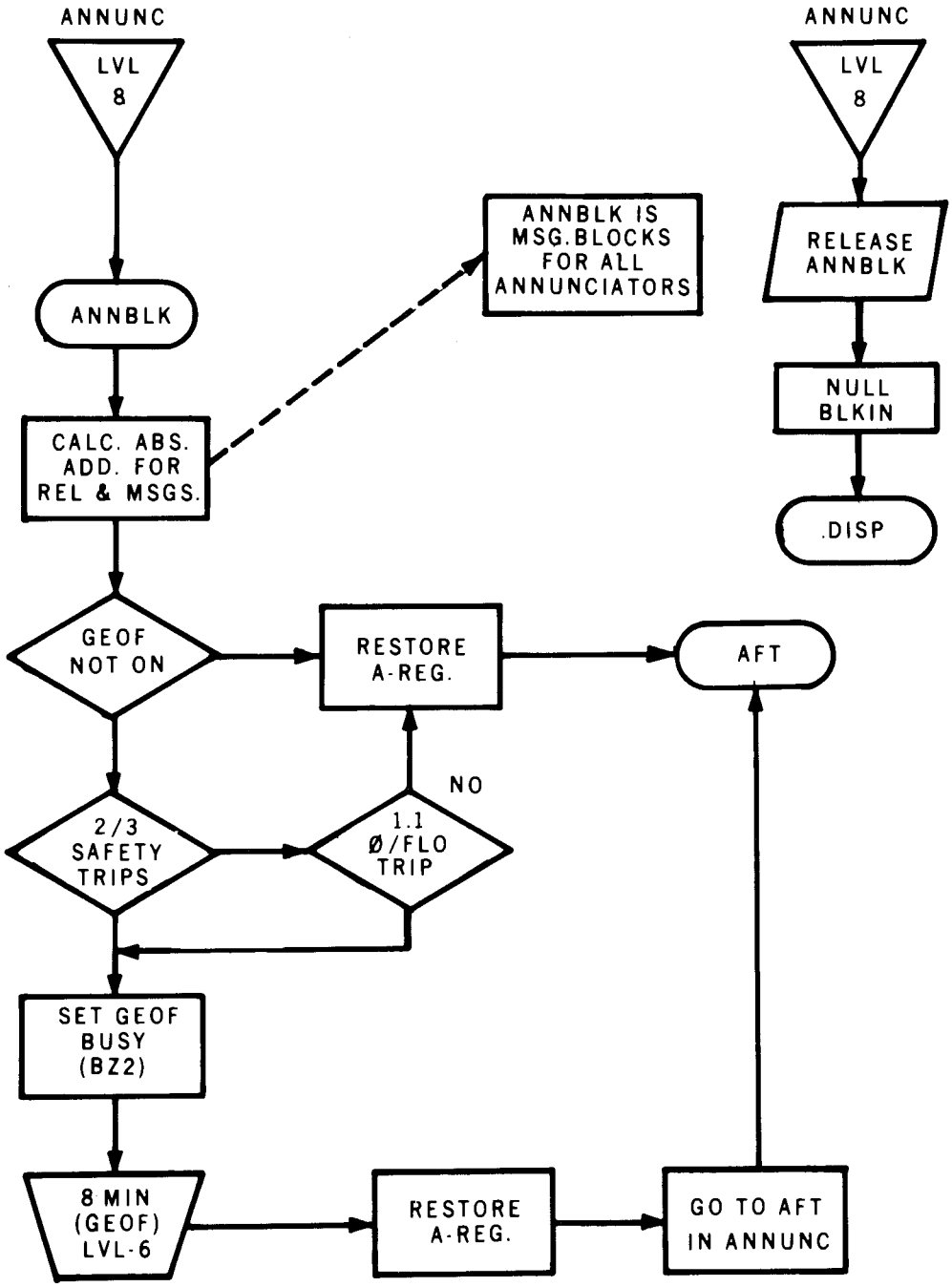


Fig. 37. Logic block diagram of program ANNBLK.

4.3 Logic Diagrams for Support Software

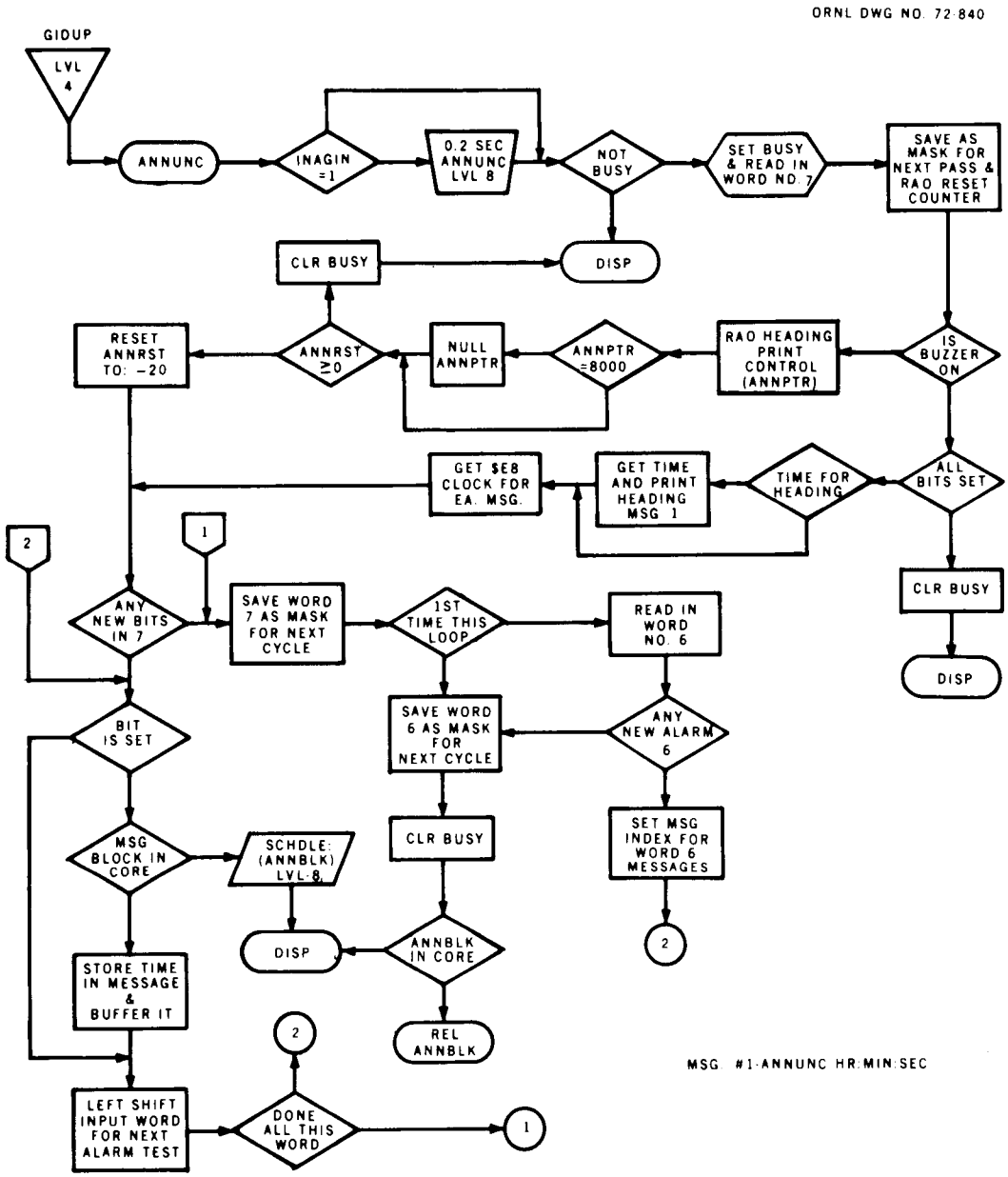


Fig. 38. Logic block diagram of program ANNUNC.

ORNL DWG NO. 846

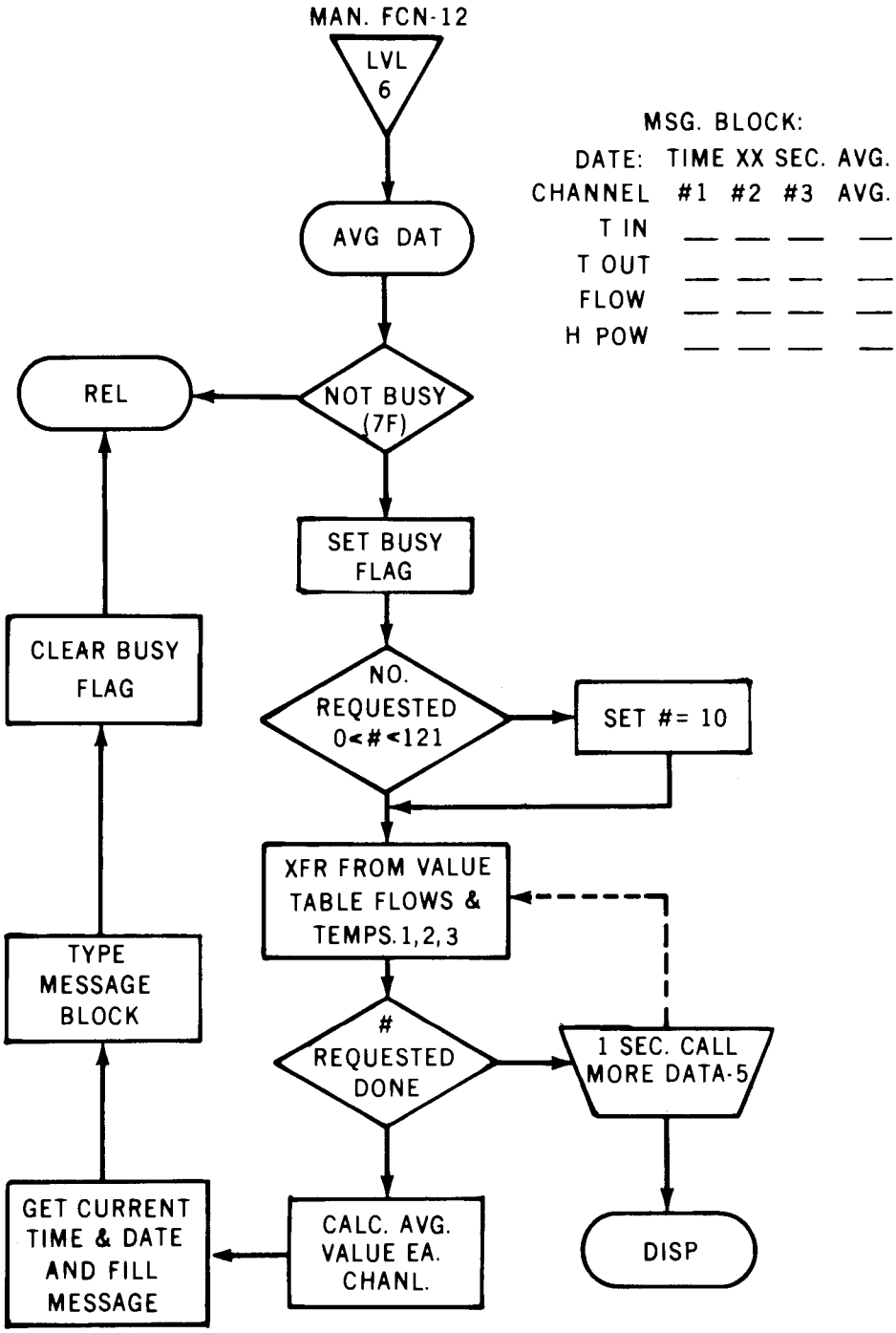


Fig. 39. Logic block diagram of program AVGDAT.

ORNL DWG NO. 72-862

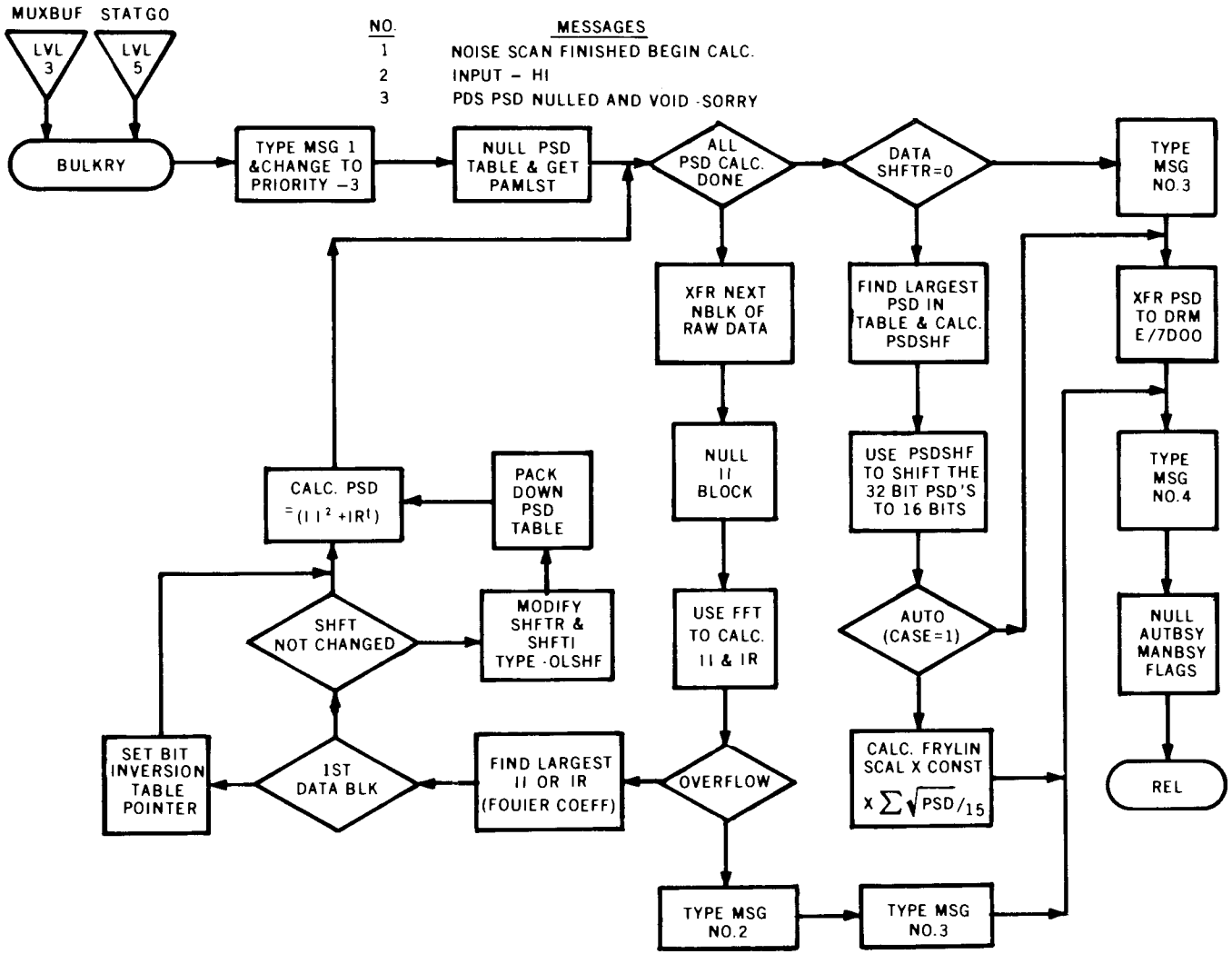


Fig. 40. Logic block diagram of program BULKRY.

ORNL DWG NO. 72-809

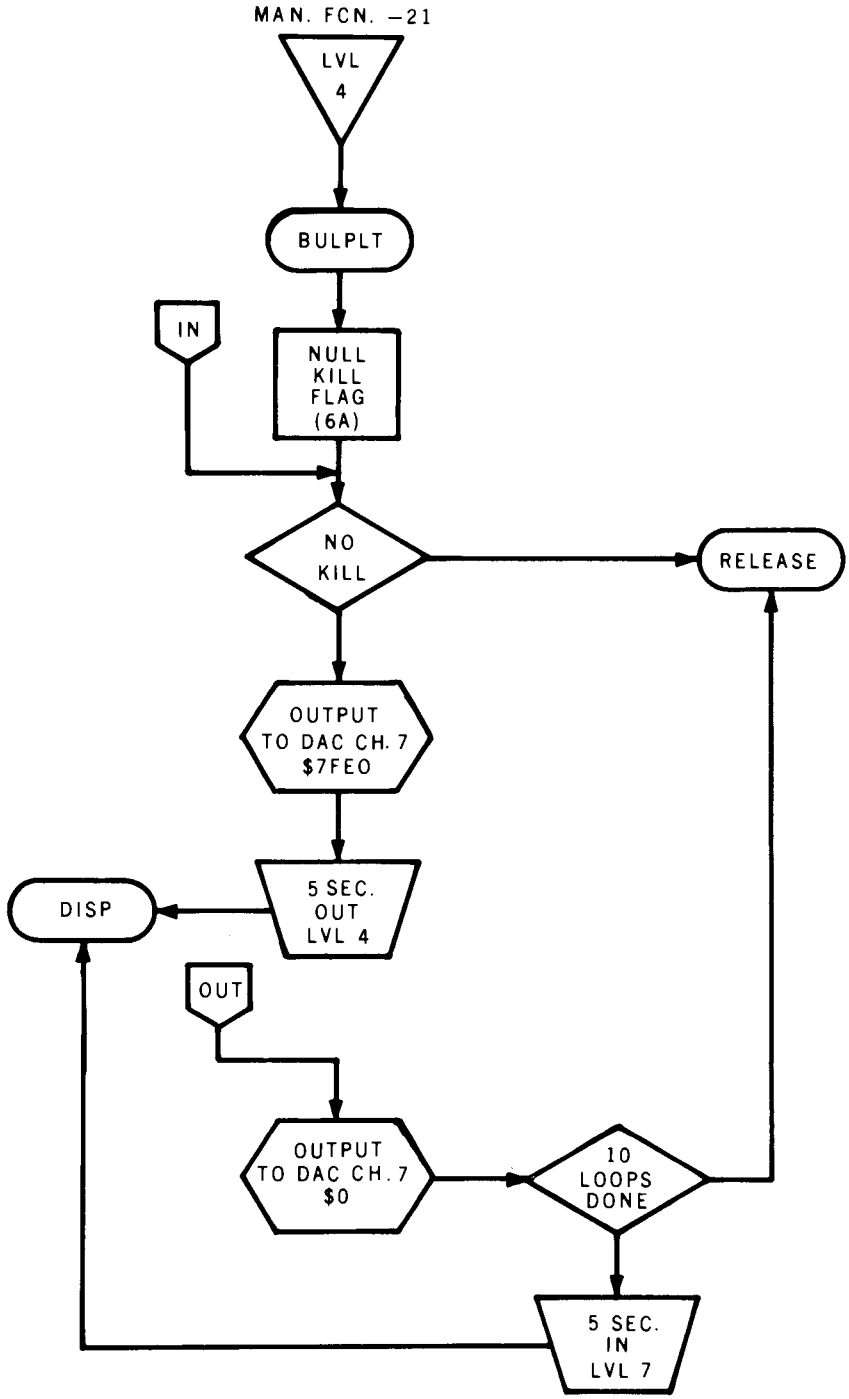


Fig. 41. Logic block diagram of program BULPLT.

ORNL DWG NO. 72-854

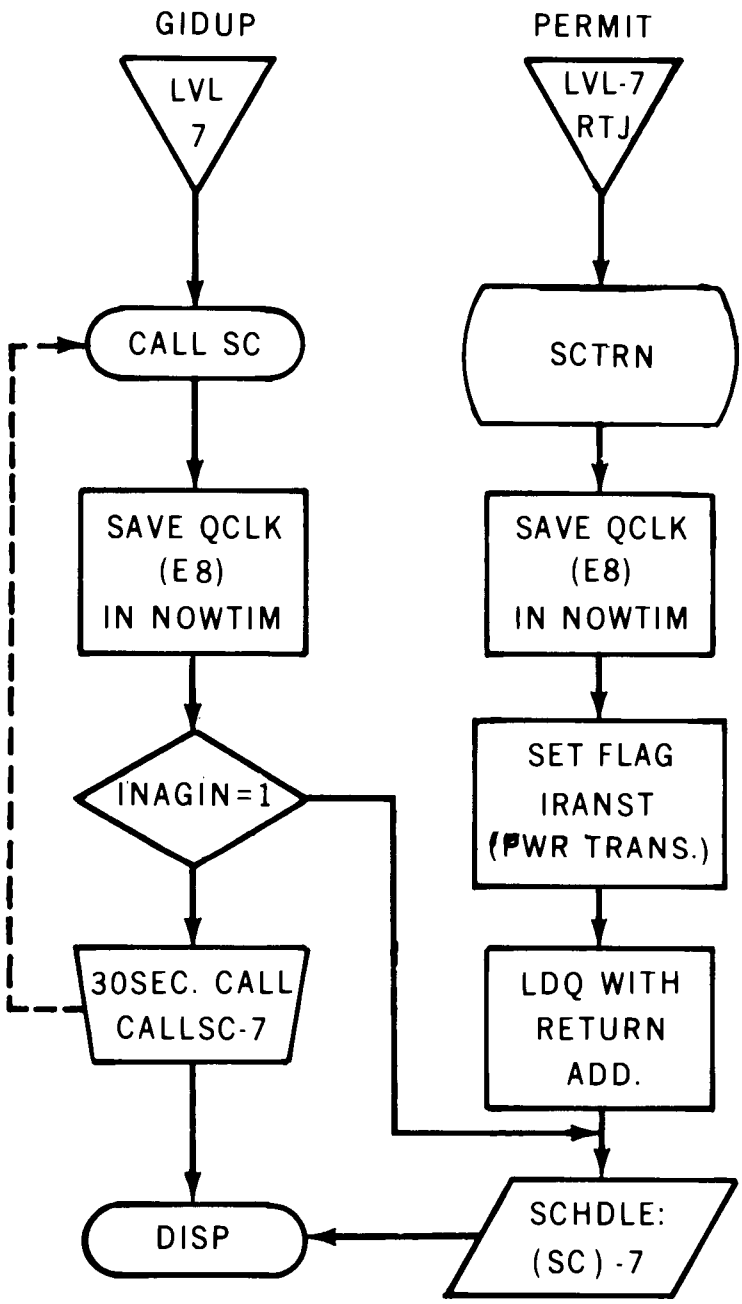


Fig. 42. Logic block diagram of program CALLSC.

ORNL DWG NO. 72-836

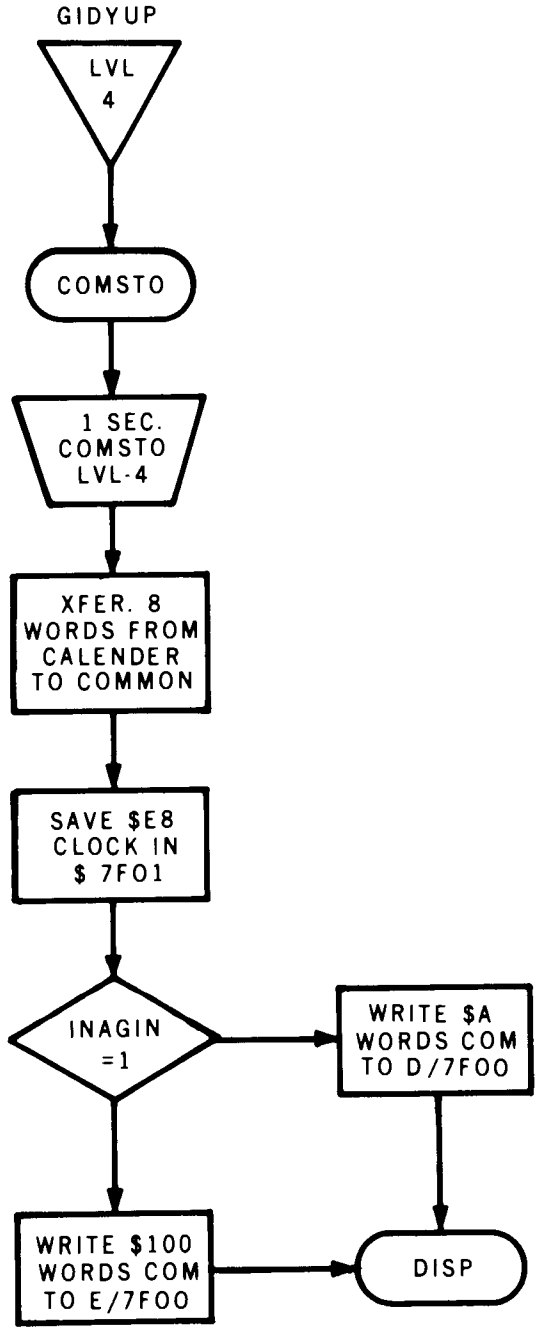


Fig. 43. Logic block diagram of program COMSTO.

ORNL DWG NO. 72-808

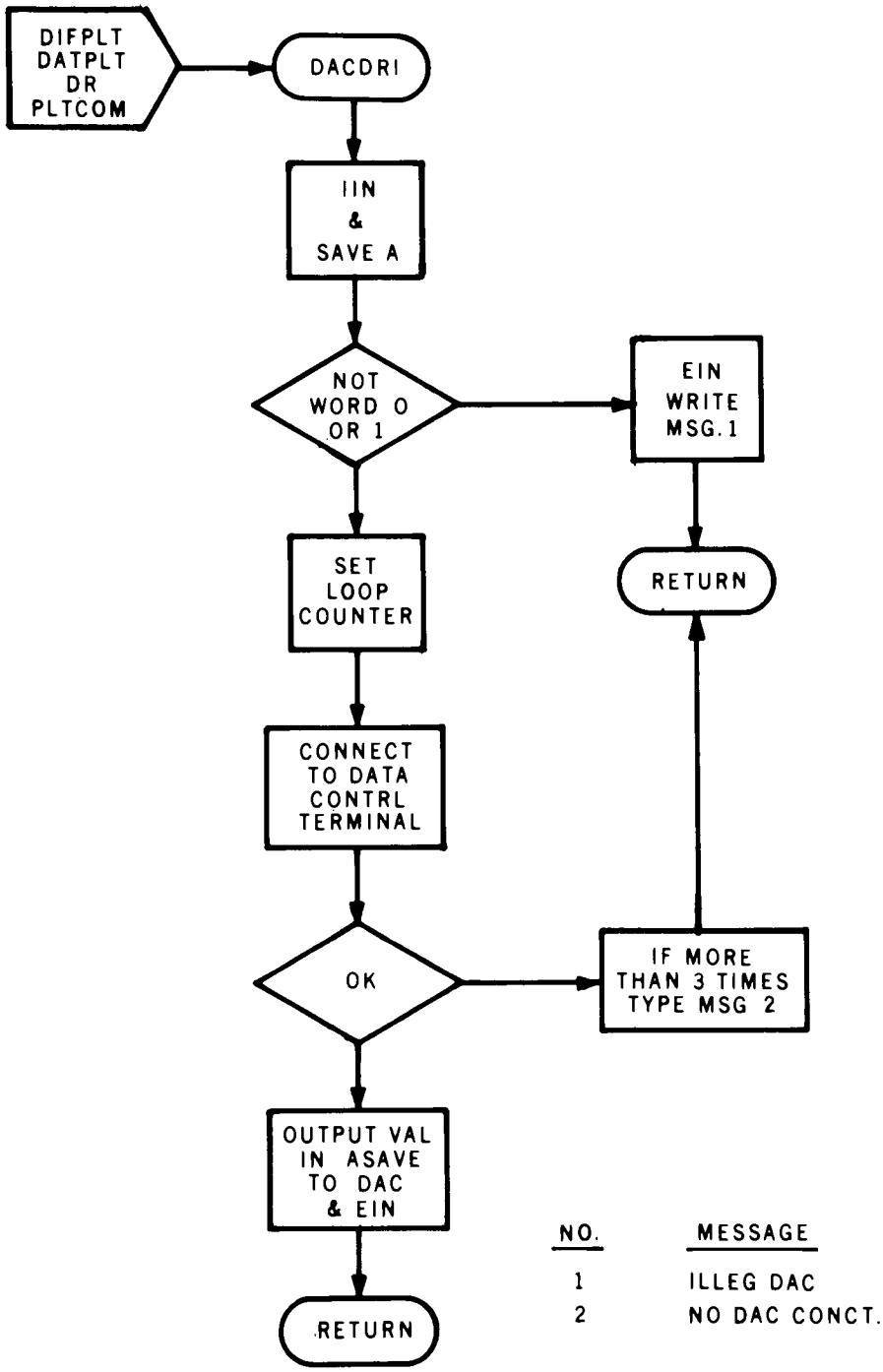


Fig. 44. Logic block diagram of program DACDRI.

ORNL DWG NO. 72-806

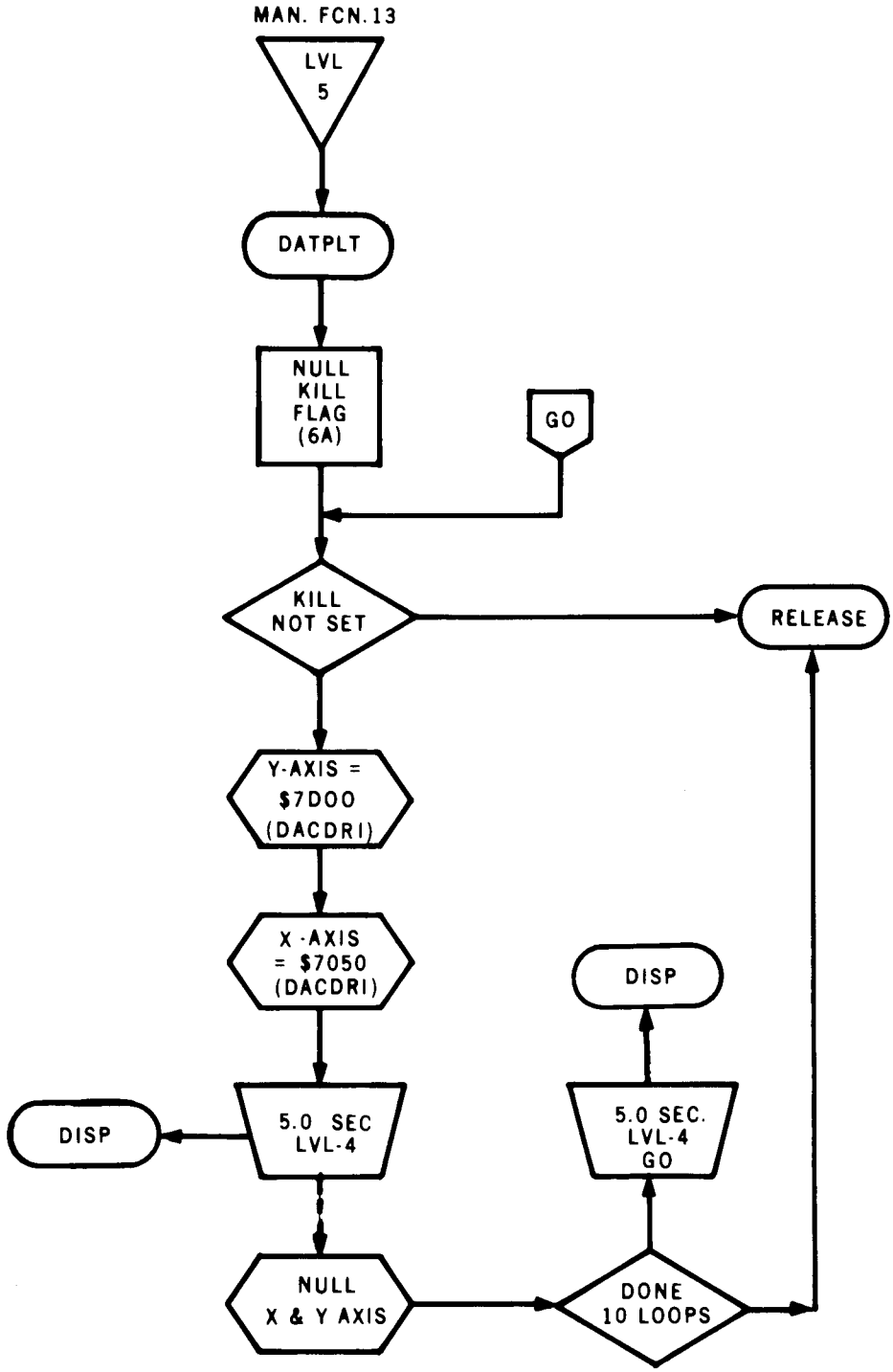


Fig. 45. Logic block diagram of program DATPLT.

ORNL DWG NO.72-832

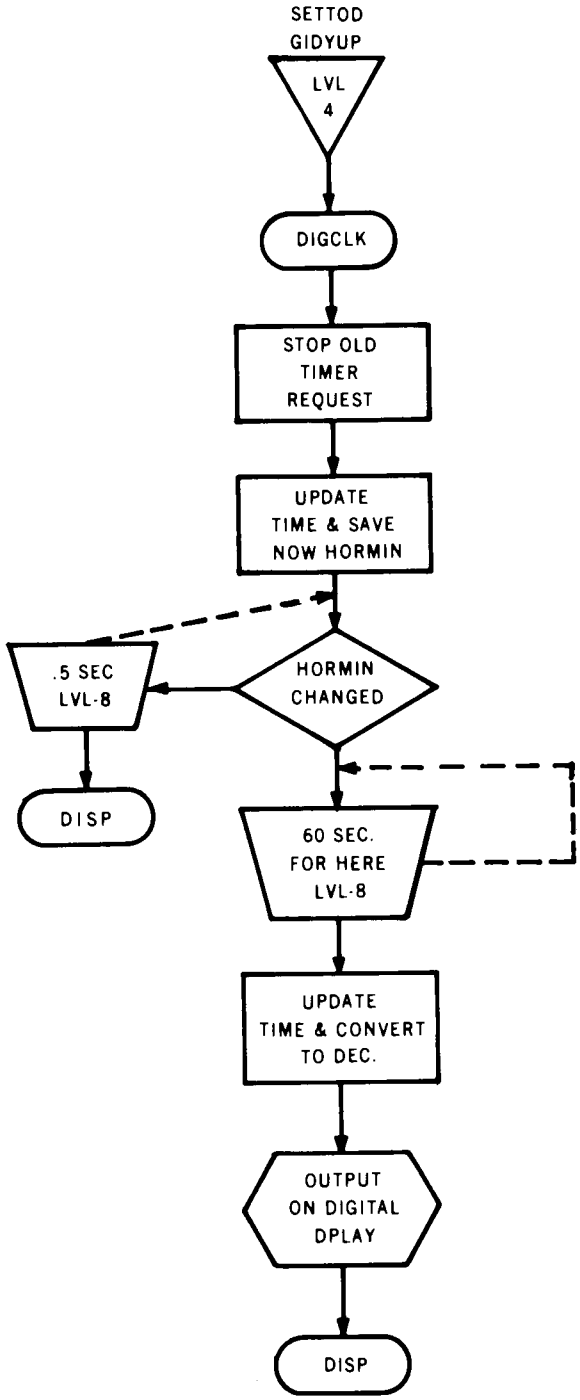


Fig. 46. Logic block diagram of program DIGCLK.

ORNL DWG NO. 72-812

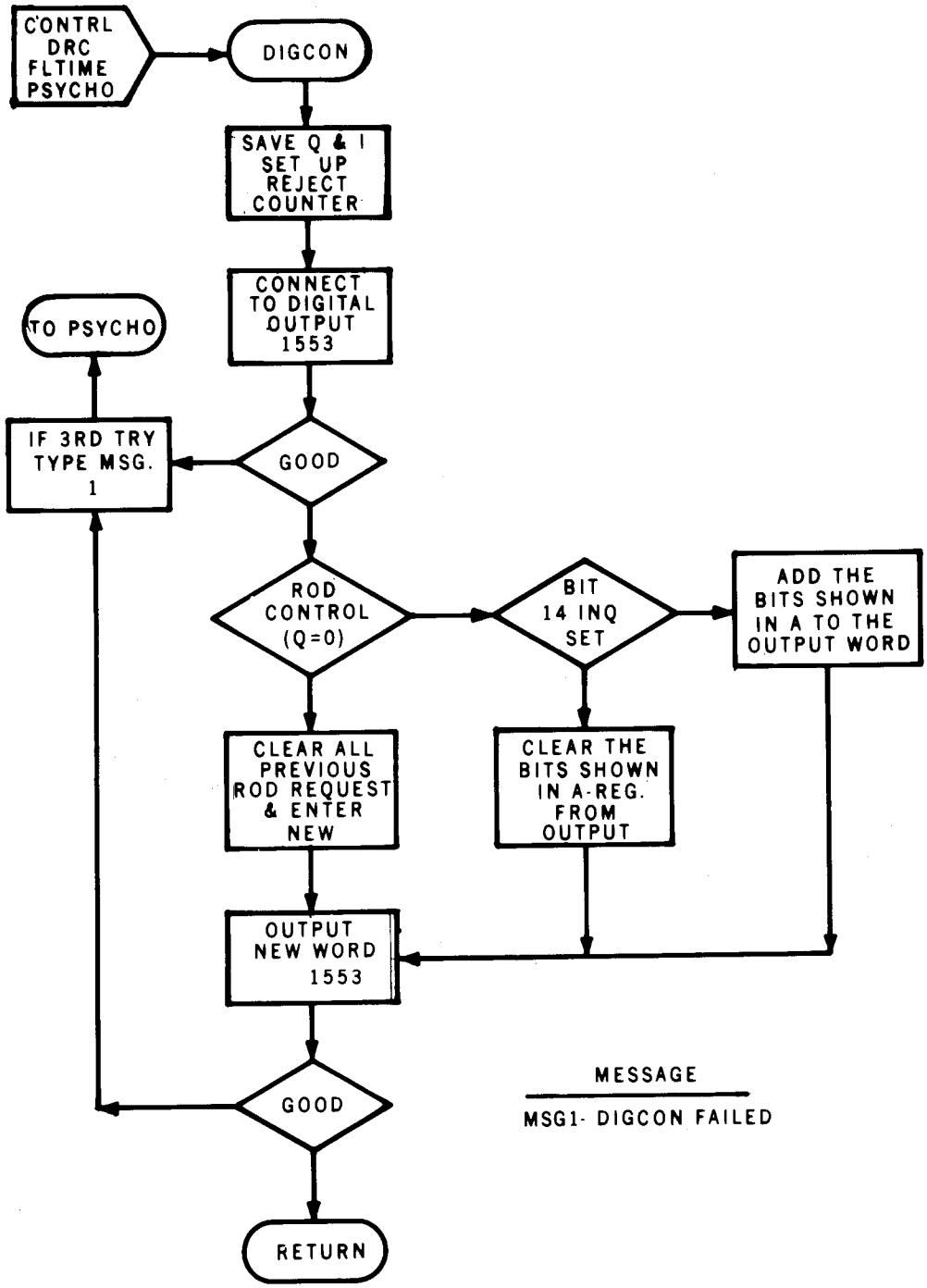


Fig. 47. Logic block diagram of program DIGCON.

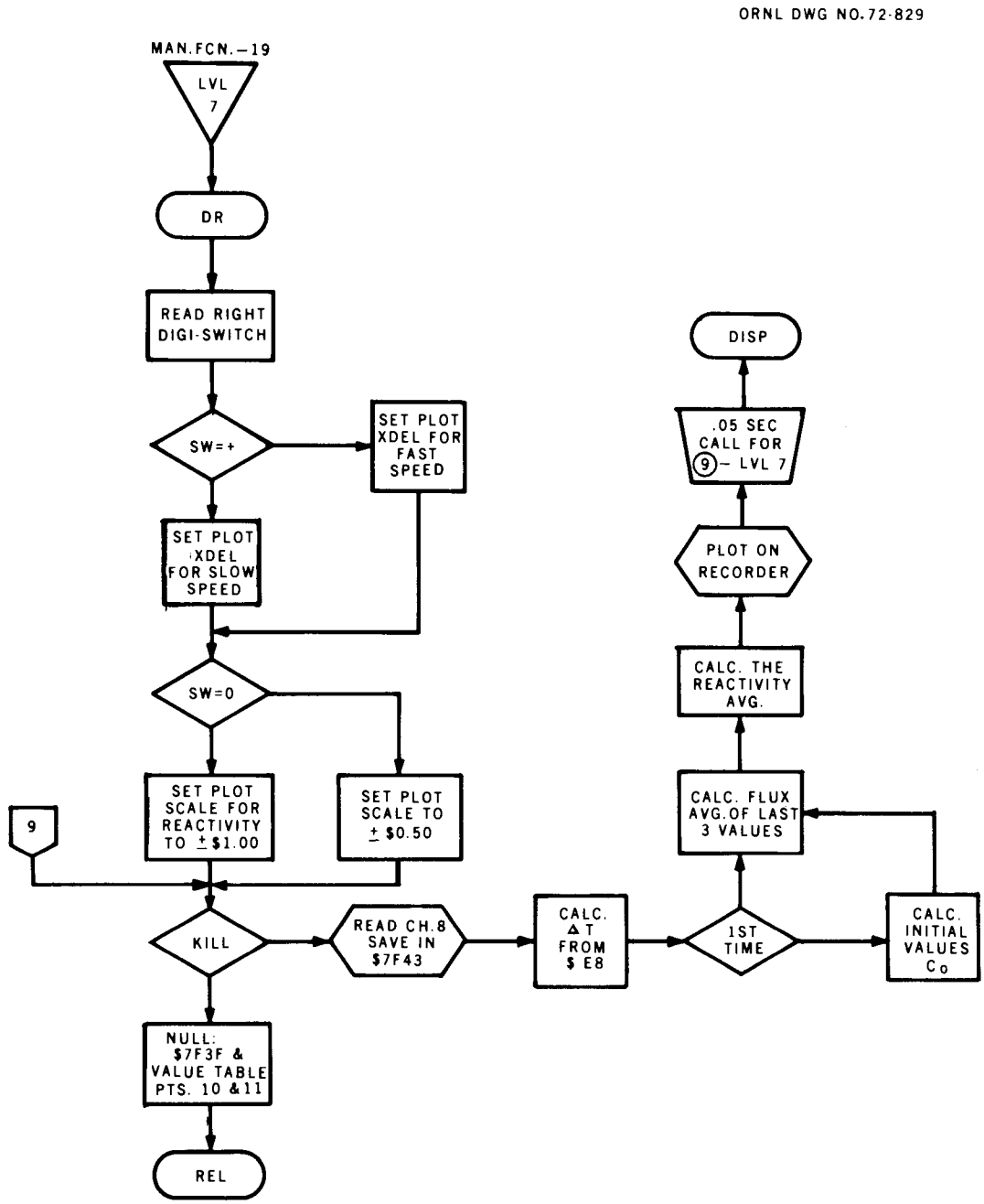
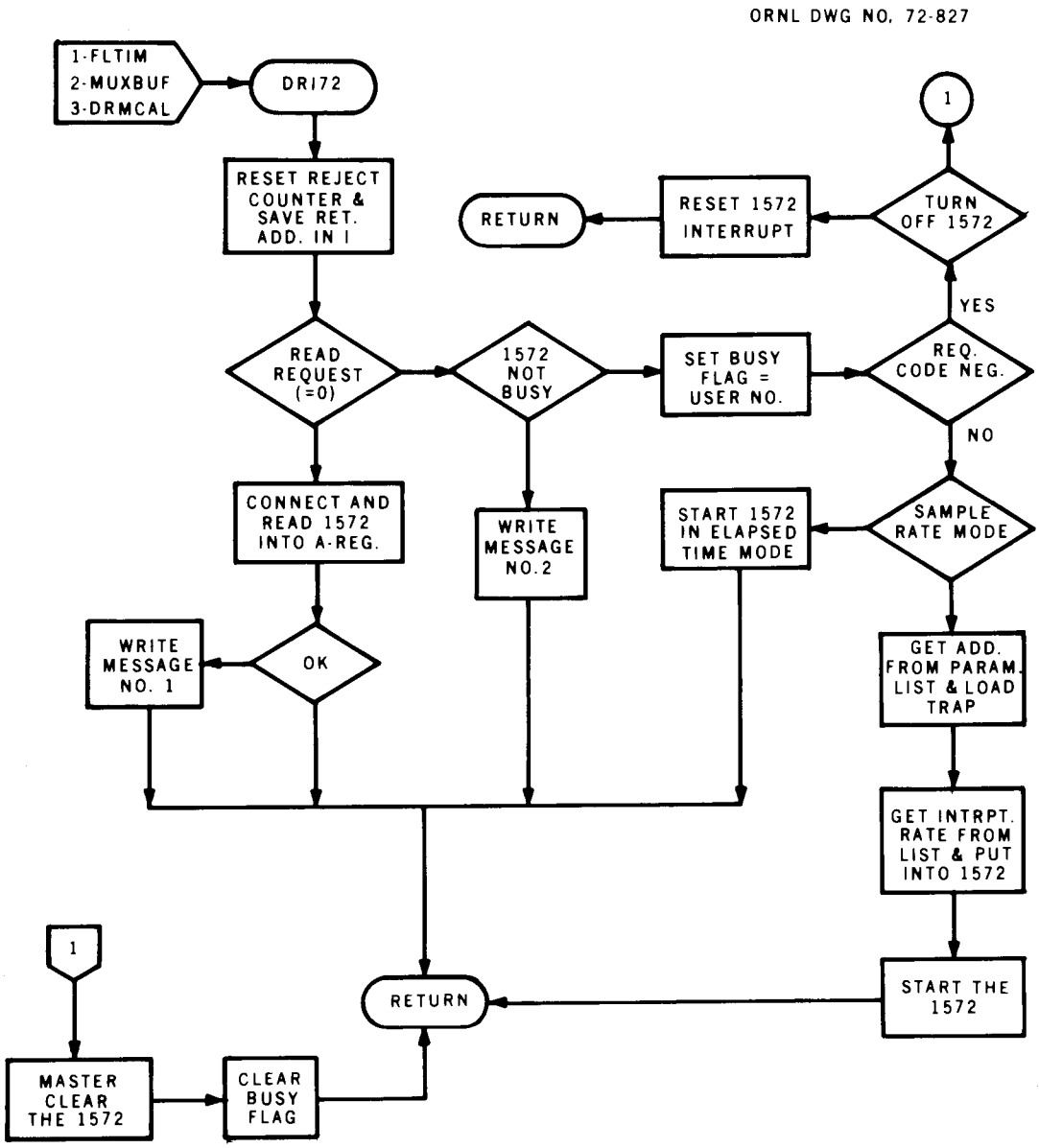


Fig. 48. Logic block diagram of program DR.



NO.	MESSAGES
1	1572 CONCT. BAD
2	1572 BUSY

Fig. 49. Logic block diagram of program DR172.

ORNL DWG NO. 72-839

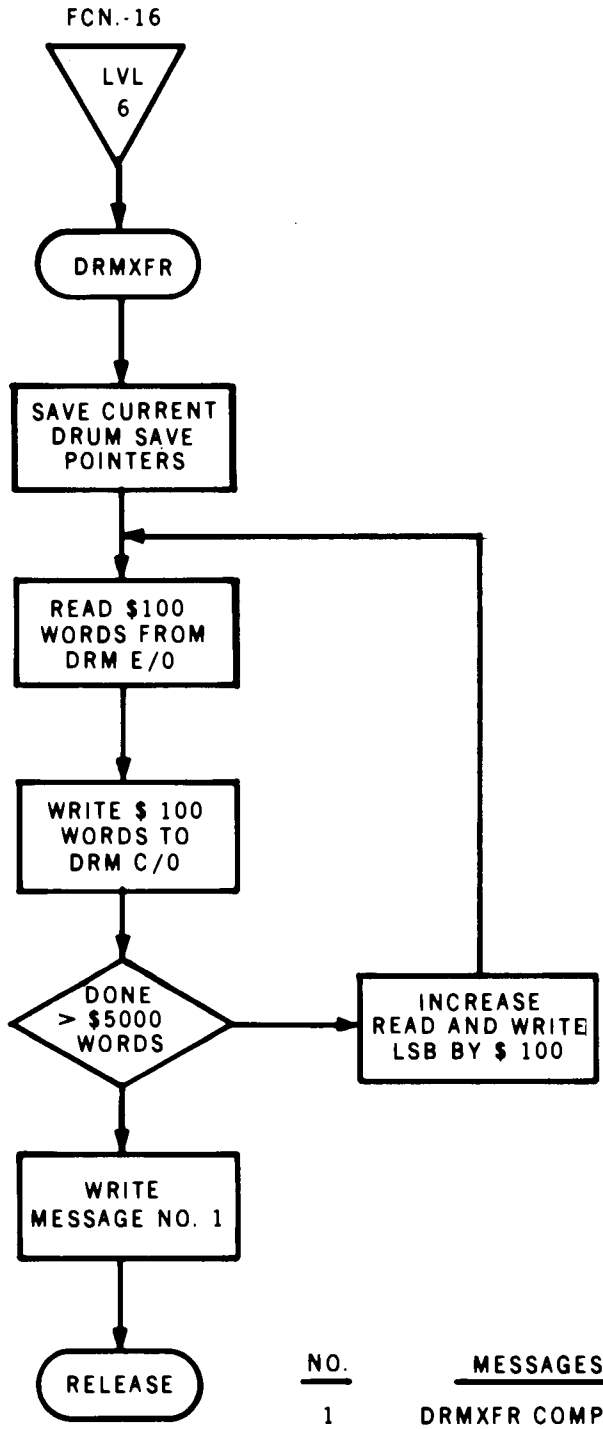


Fig. 50. Logic block diagram of program DRMXFR.

ORNL DWG NO. 72-805

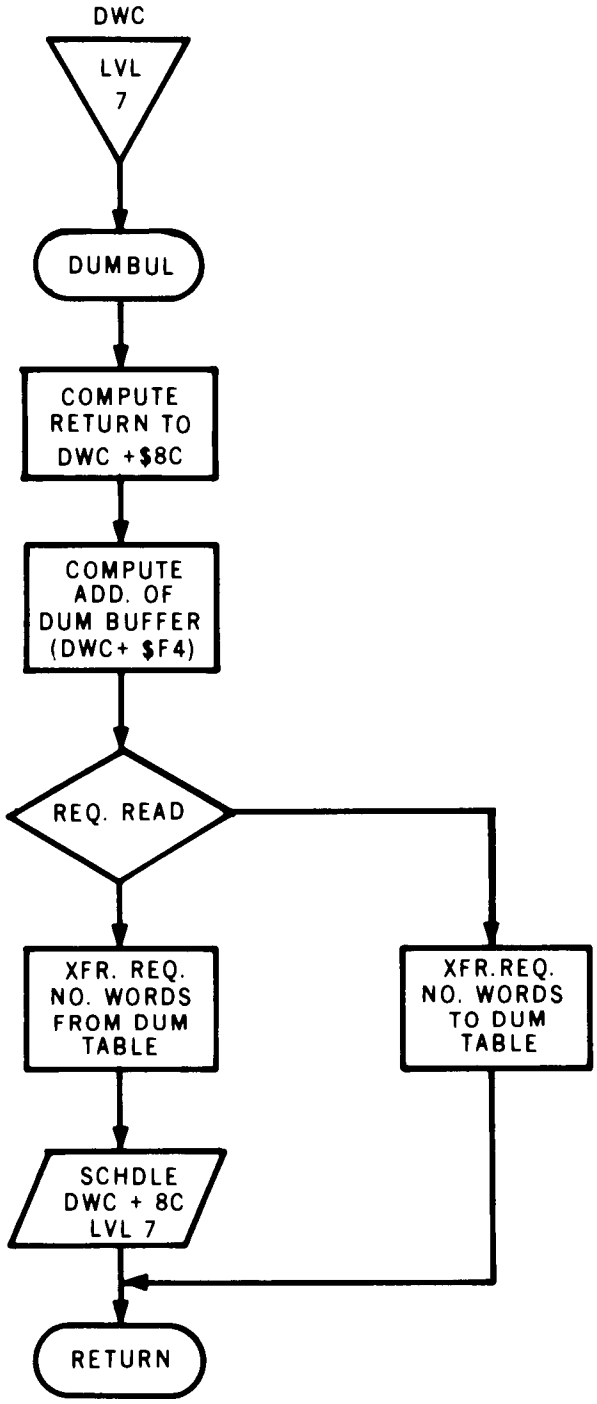
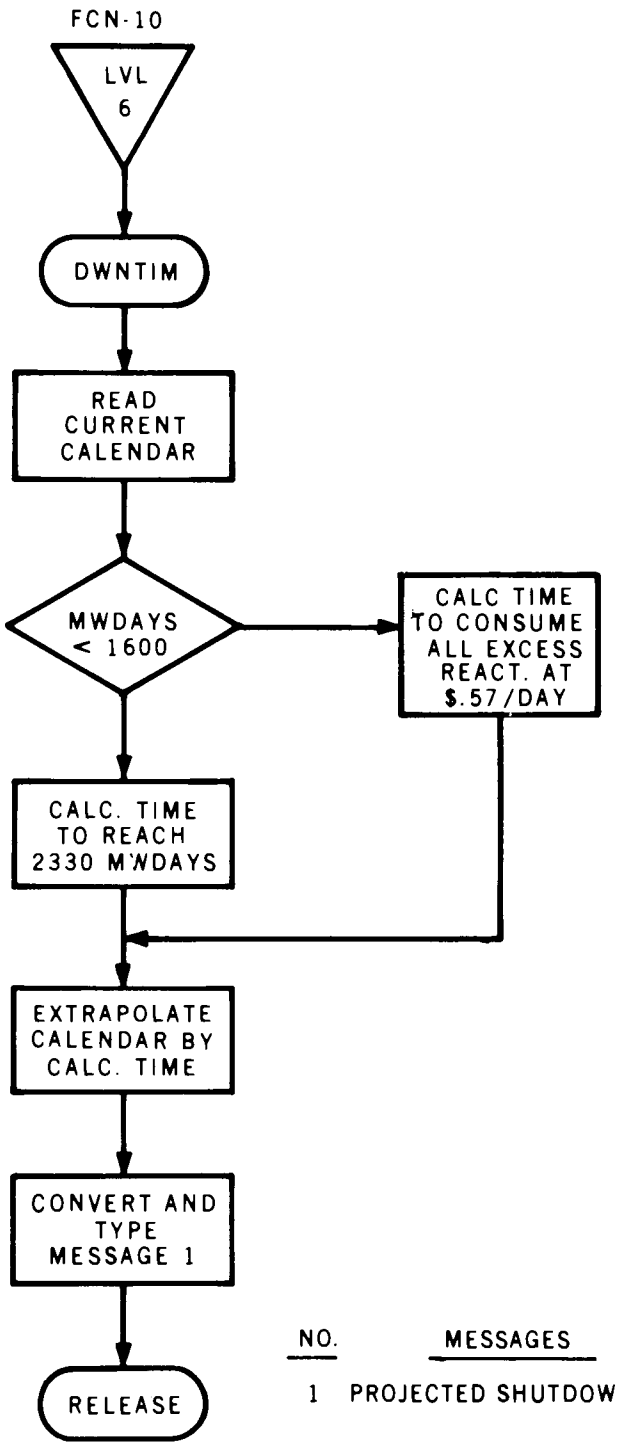


Fig. 51. Logic block diagram of program DUMBUL.

ORNL DWG NO. 72-817



NO.	MESSAGES
1	PROJECTED SHUTDOWN IS (DATE:TIME)

Fig. 52. Logic block diagram of program DWNTIM.

ORNL DWG NO. 72-807

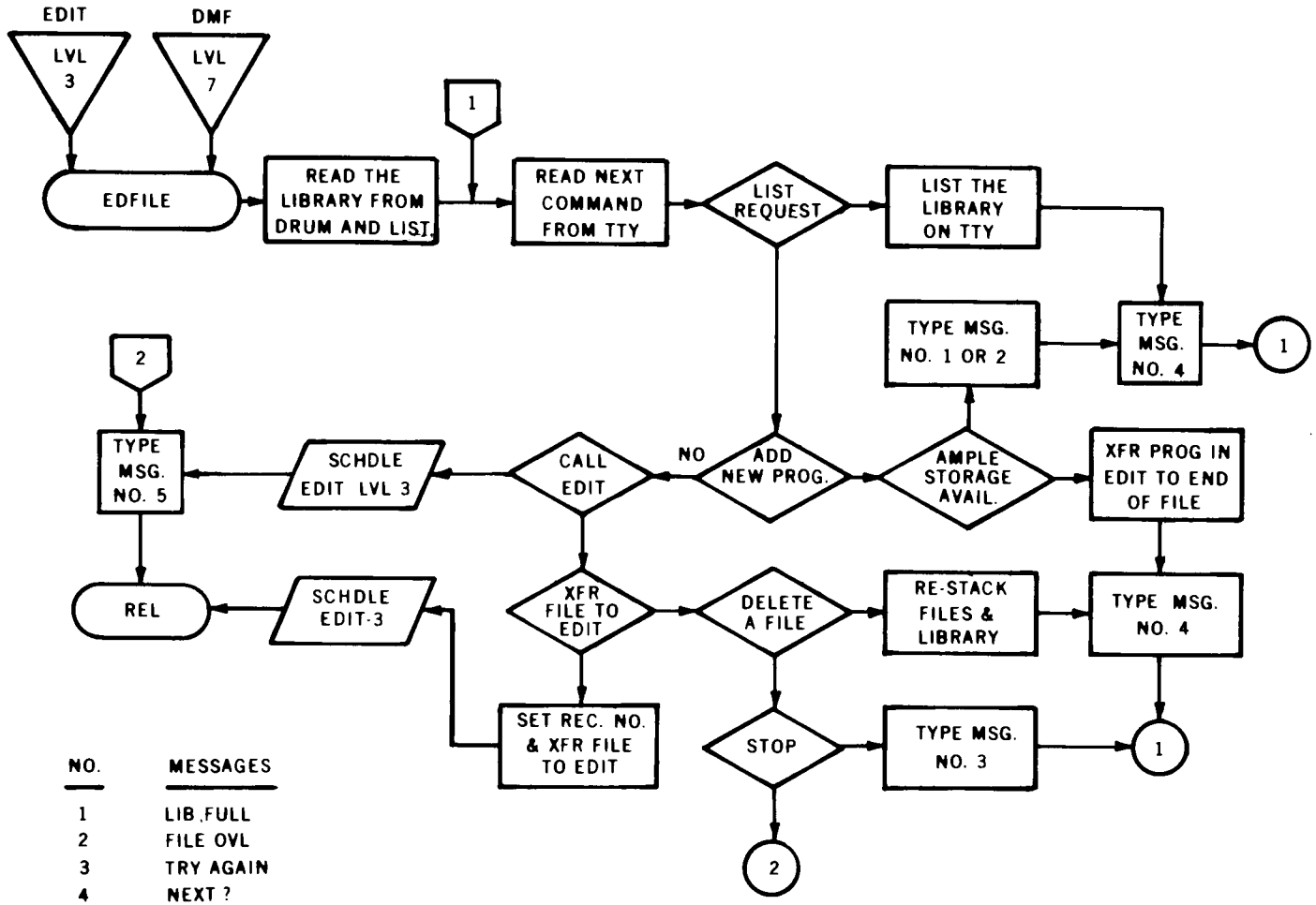


Fig. 53. Logic block diagram of program EDFILE.

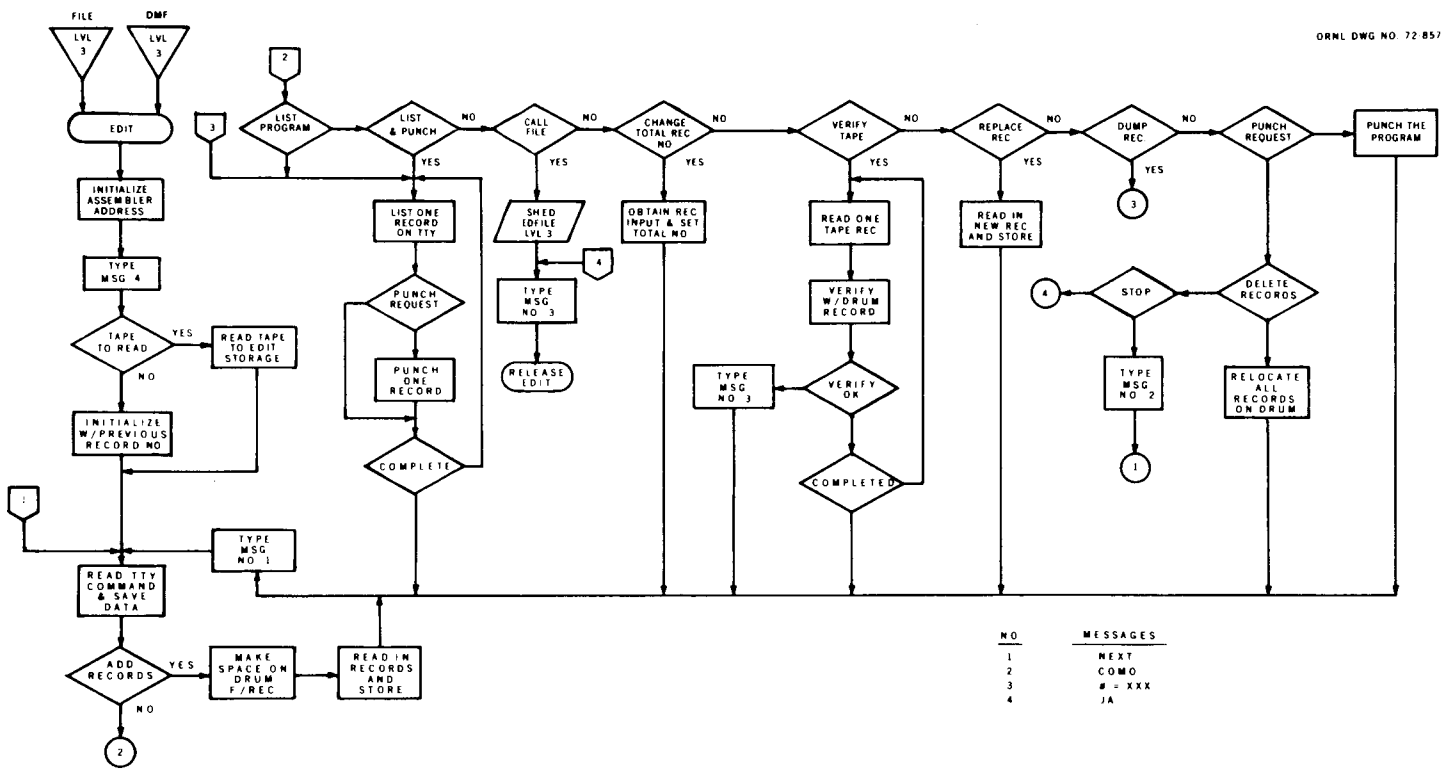


Fig. 54. Logic block diagram of program EDIT.

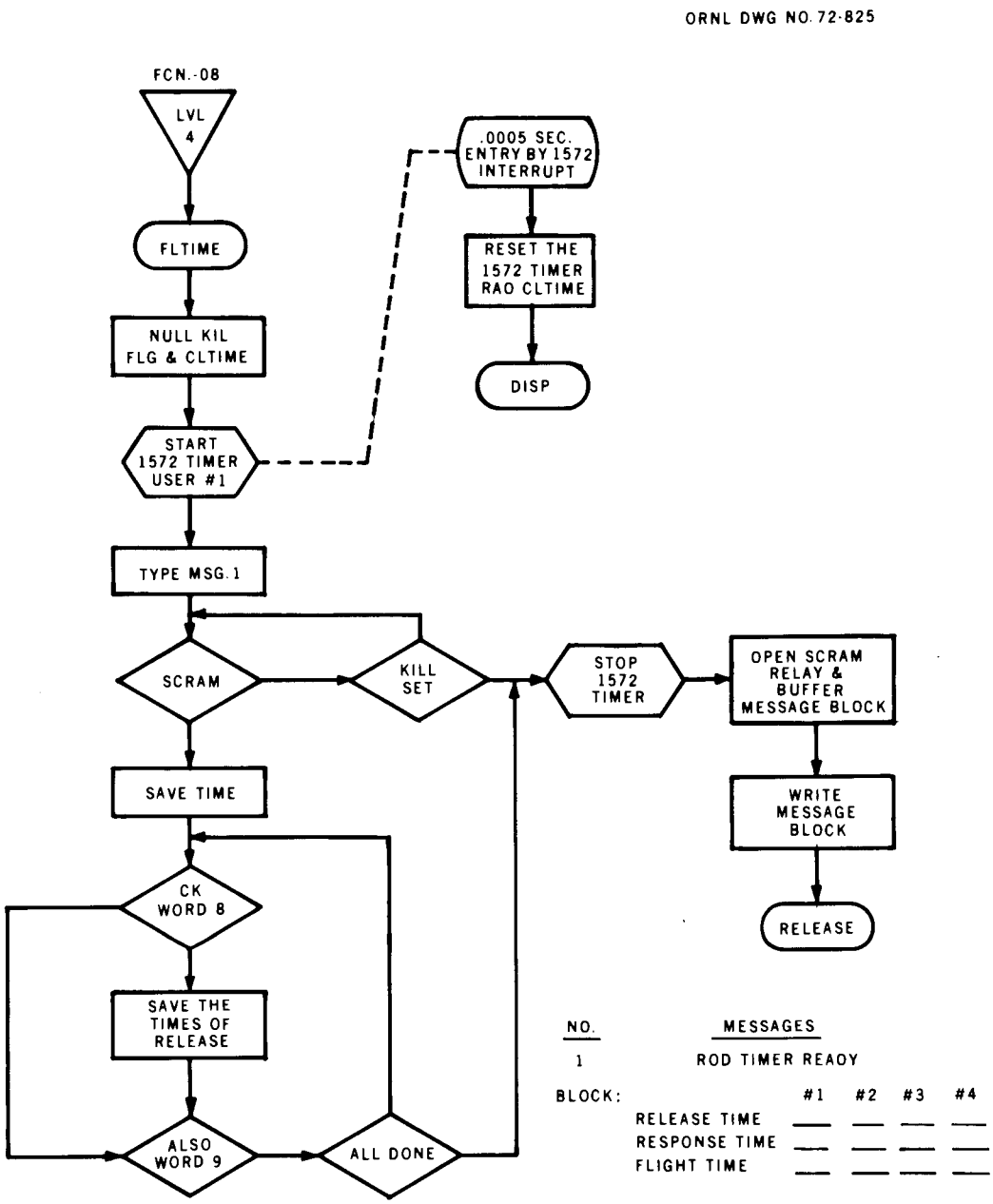


Fig. 55. Logic block diagram of program FLTIME.

ORNL DWG NO. 72-811

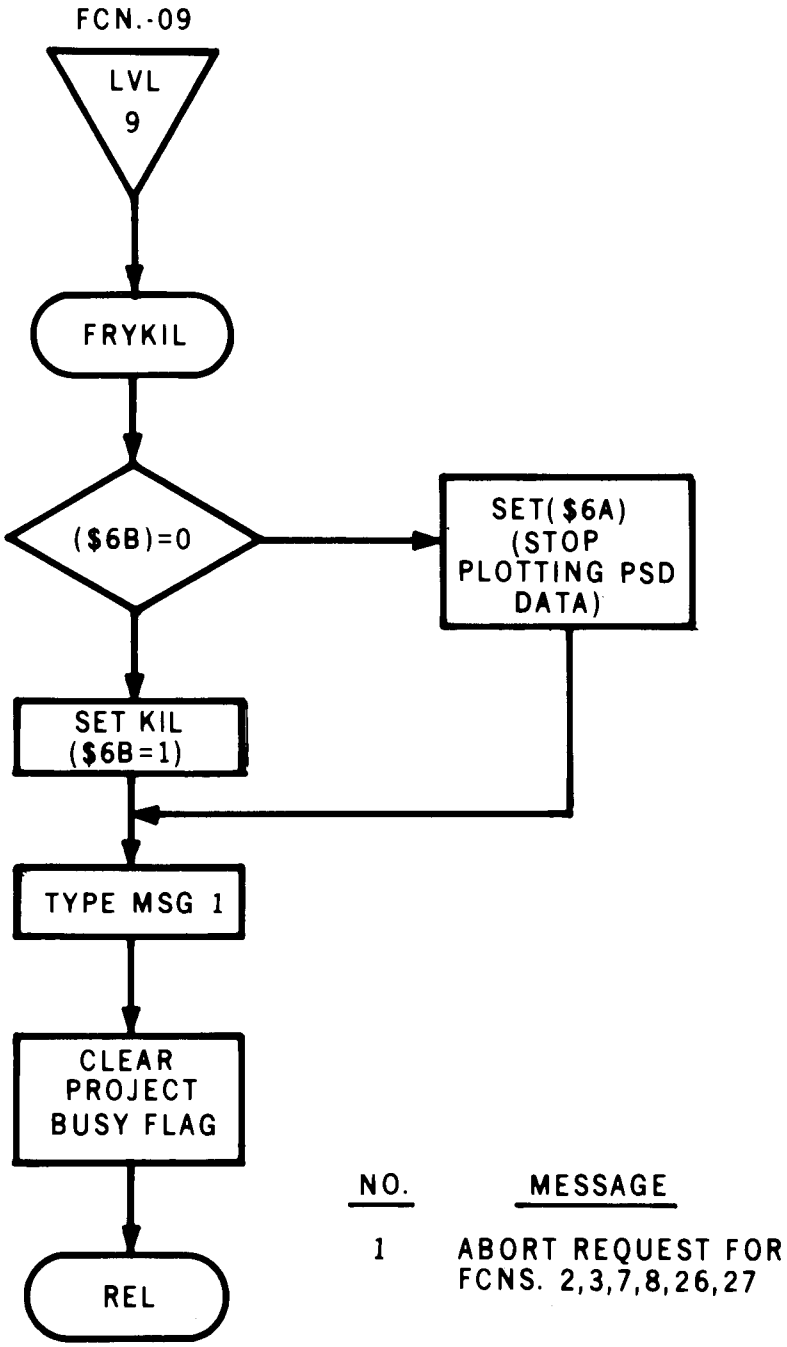
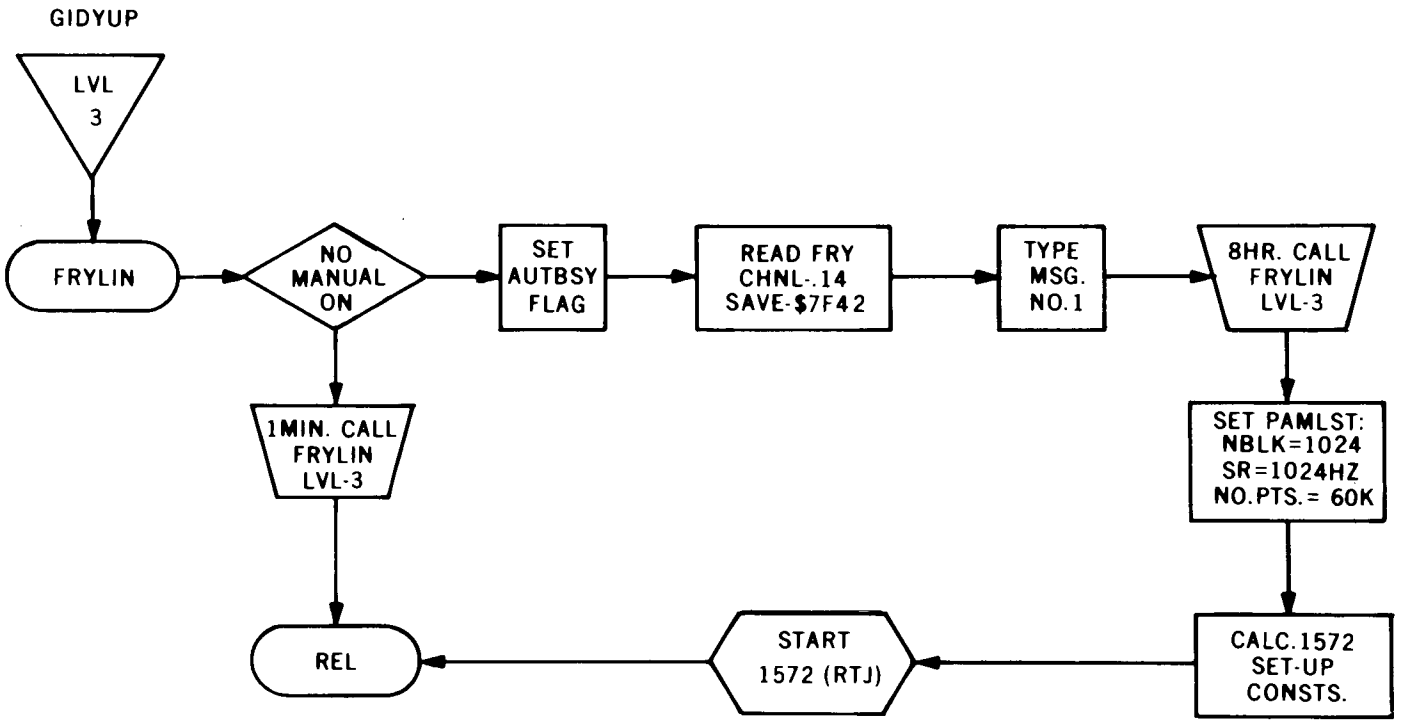


Fig. 56. Logic block diagram of program FRYKIL.

ORNL DWG NO. 72-815



NO.	MESSAGES
1	TAKING NOISE DATA. PLEASE DON'T MOVE RODS.

Fig. 57. Logic block diagram of program FRYLIN.

ORNL DWG NO. 72-863

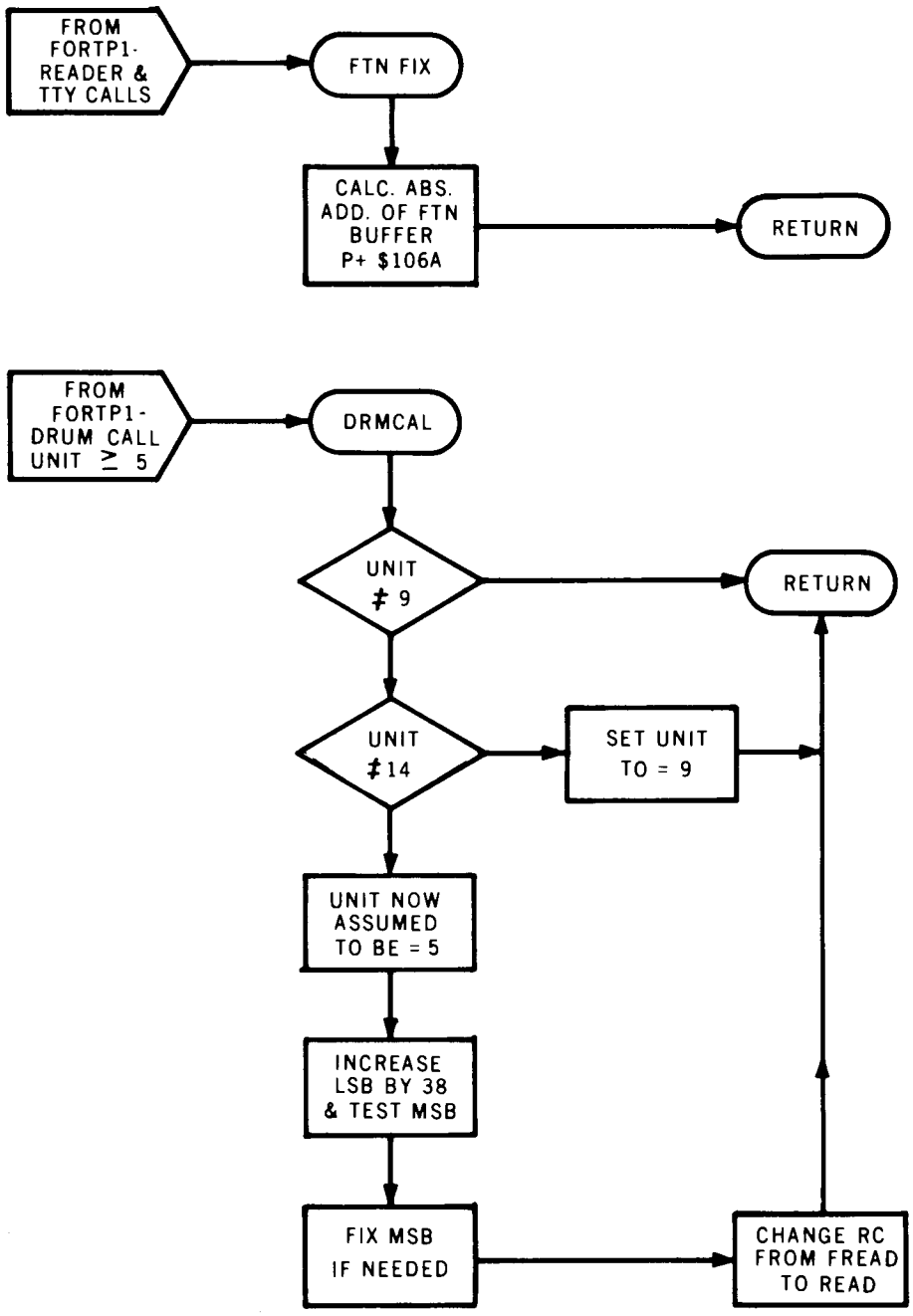
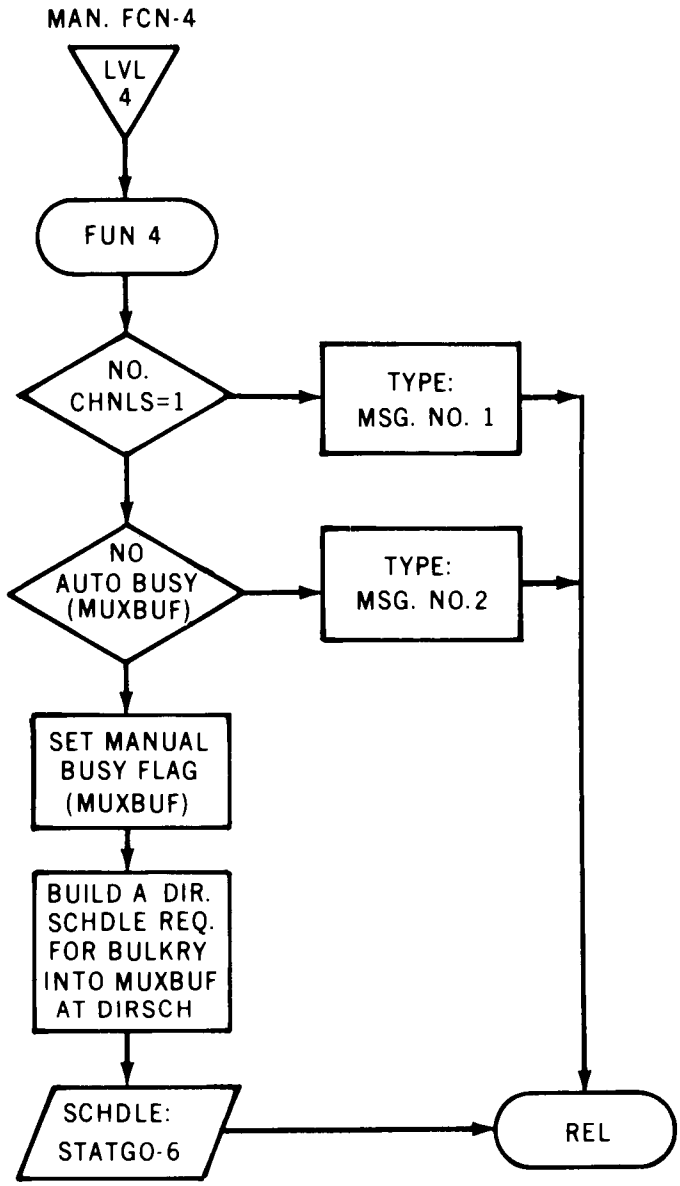


Fig. 58. Logic block diagram of program FTNFIX.

ORNL DWG NO. 72-847



NO.	MESSAGES
1	CHK.SIGN RH SWITCH
2	PDS BUSY TRY LATER

Fig. 59. Logic block diagram of program FUN4.

ORNL DWG NO. 72-810

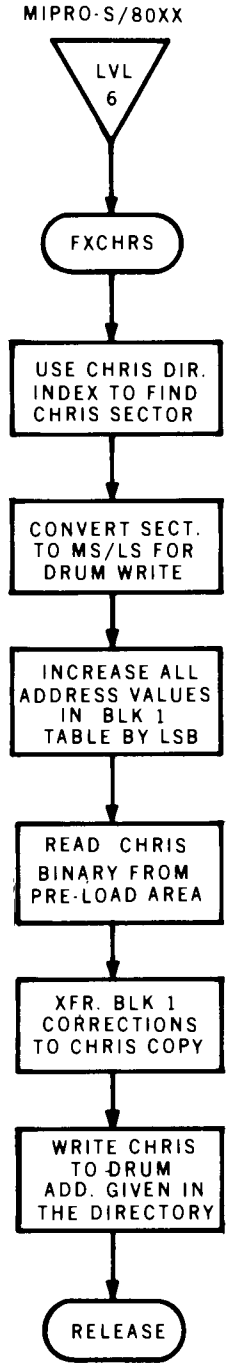


Fig. 60. Logic block diagram of program FXCHRS.

ORNL DWG NO.72-844

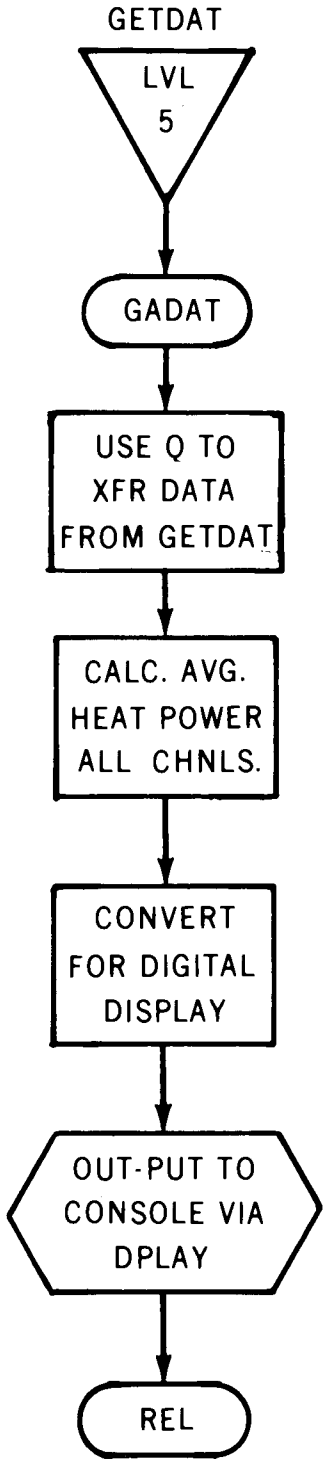


Fig. 61. Logic block diagram of program GADAT.

ORNL DWG NO. 72-819

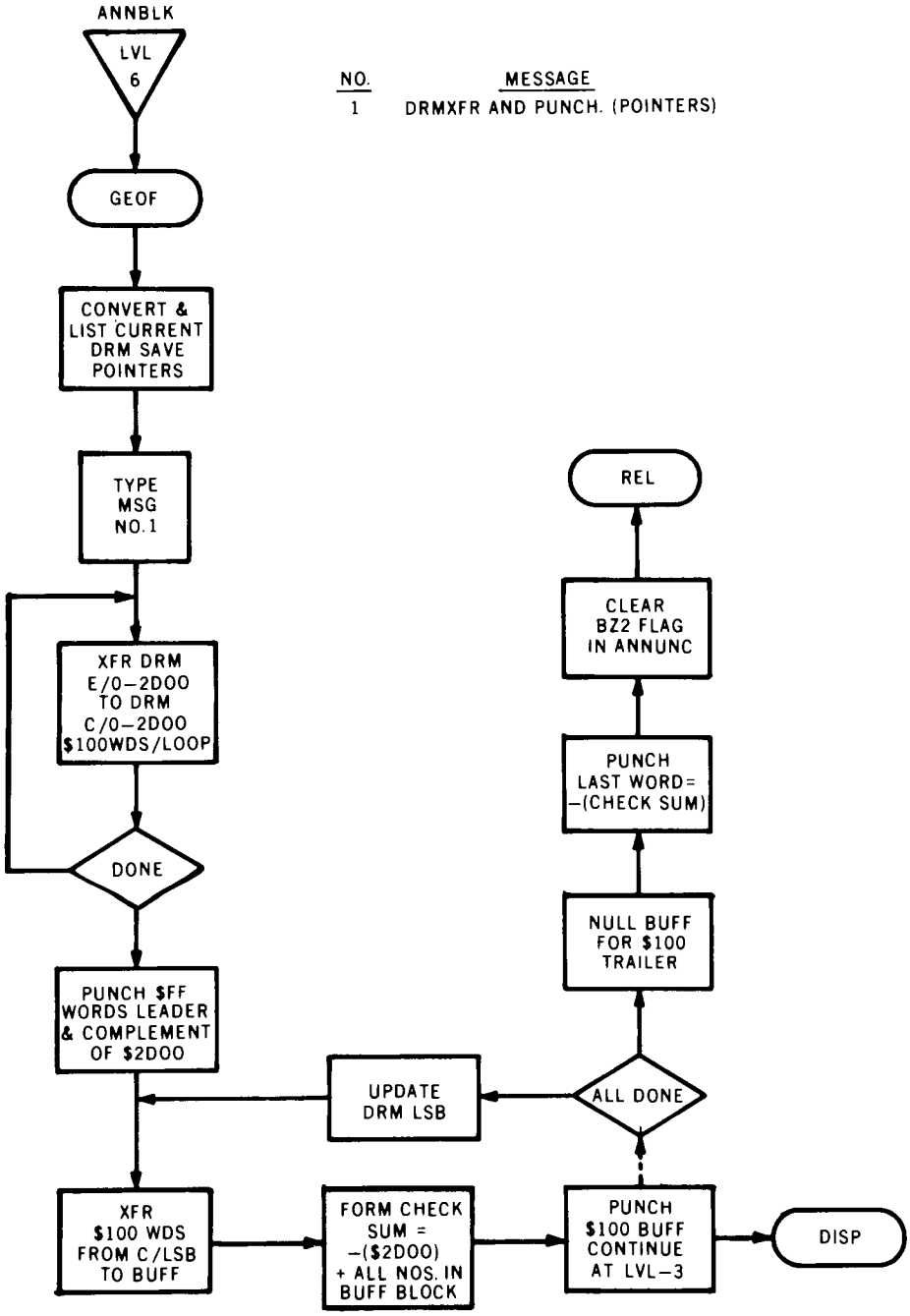


Fig. 62. Logic block diagram of program GEOF.

ORNL DWG NO. 72-851

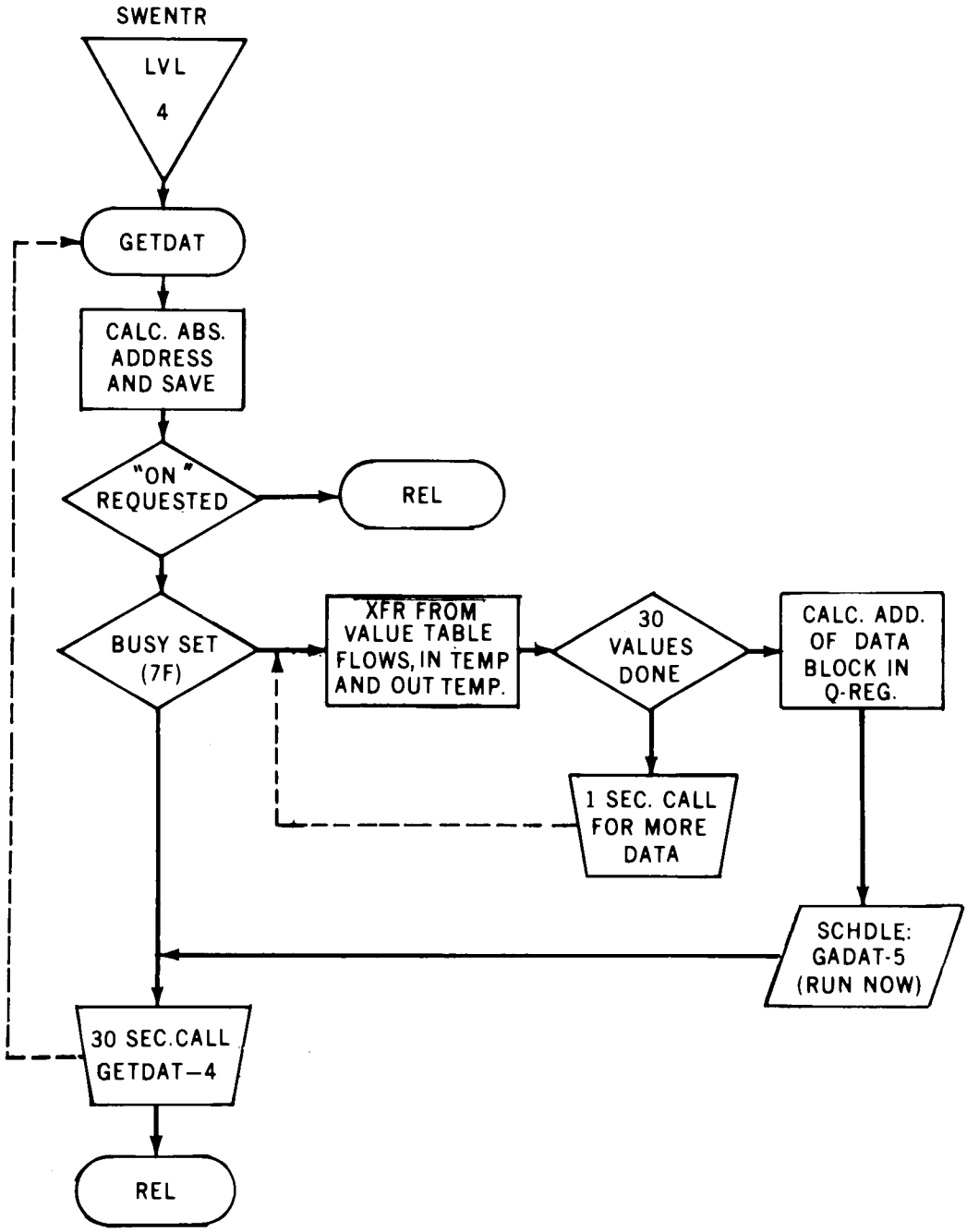
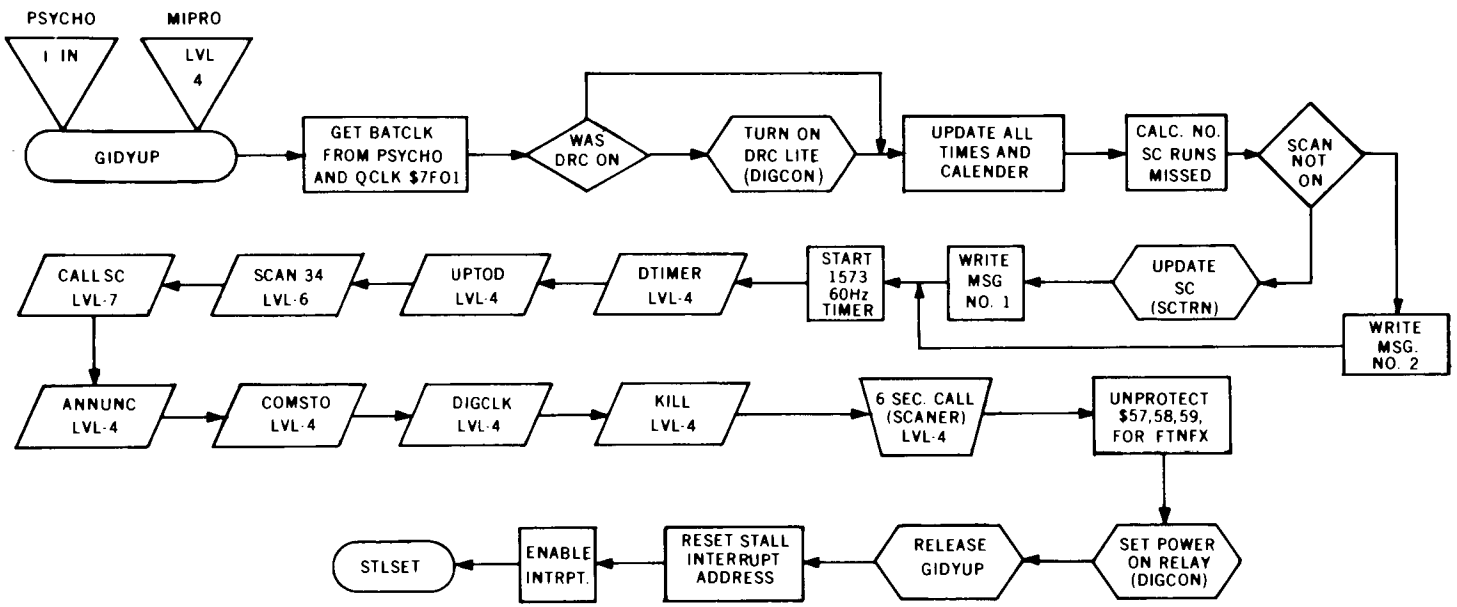


Fig. 63. Logic block diagram of program GETDAT.

ORNL DWG NO. 72-803



MESSAGE NO. 1: SCAN UP-(TIME)
NO. 2: RESET AT:(TIME)

Fig. 64. Logic block diagram of program GIDYUP.

ORNL DWG NO.72-848

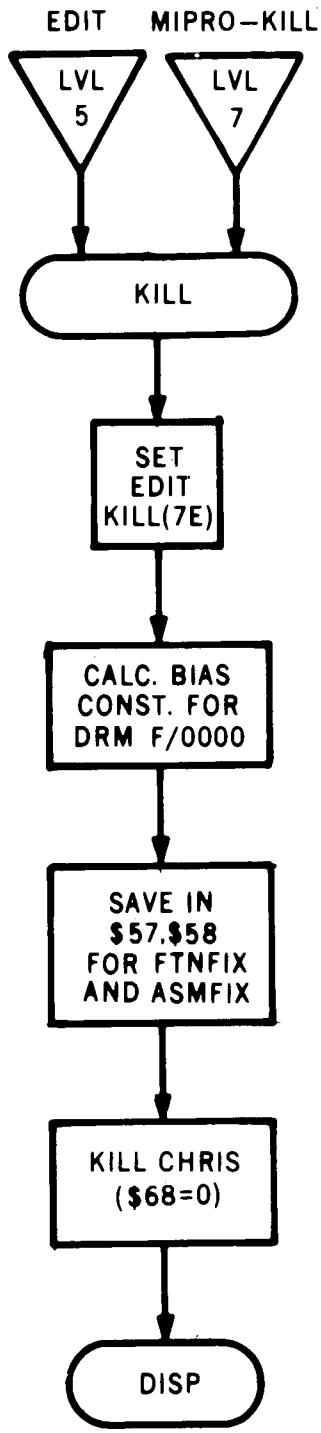


Fig. 65. Logic block diagram of program KILL.

ORNL DWG NO. 72-831

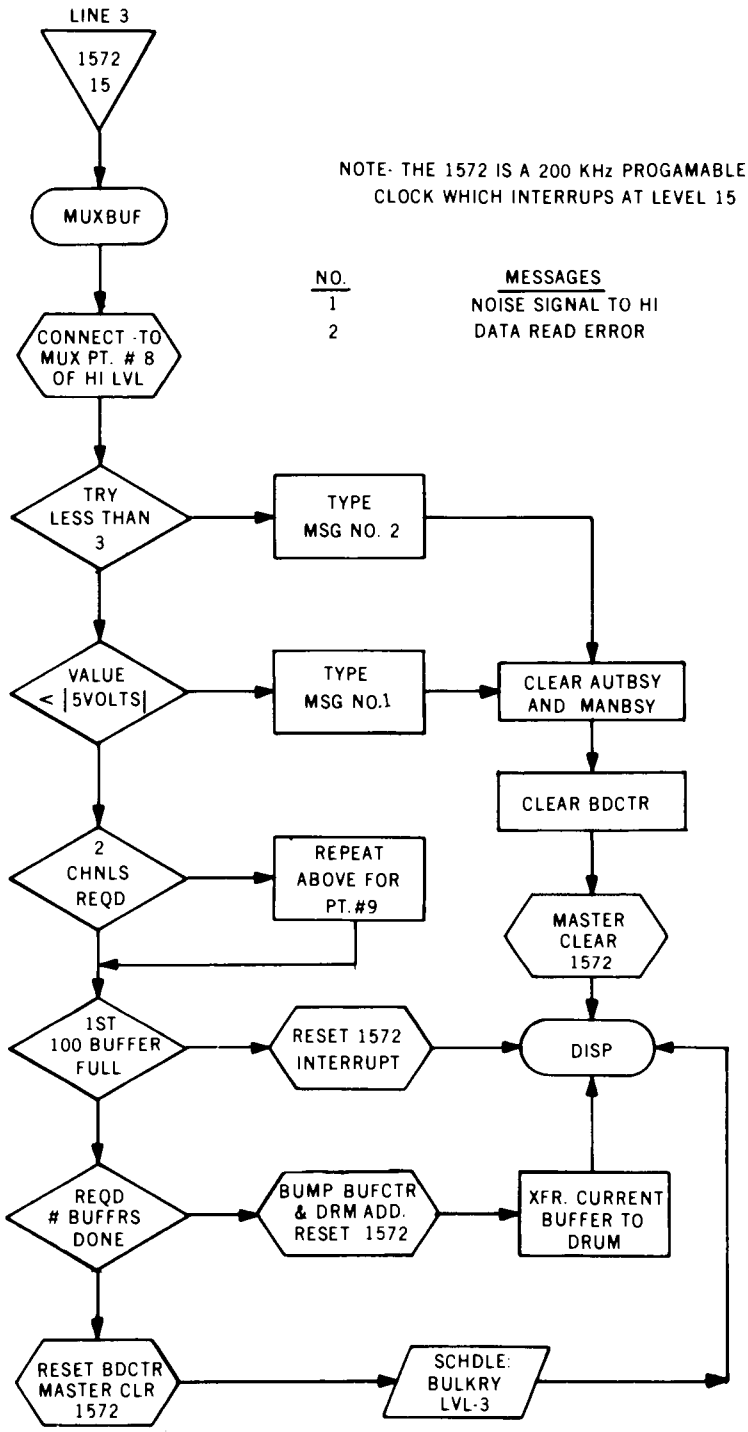


Fig. 66. Logic block diagram of program MUXBUF.

ORNL DWG NO. 72-835

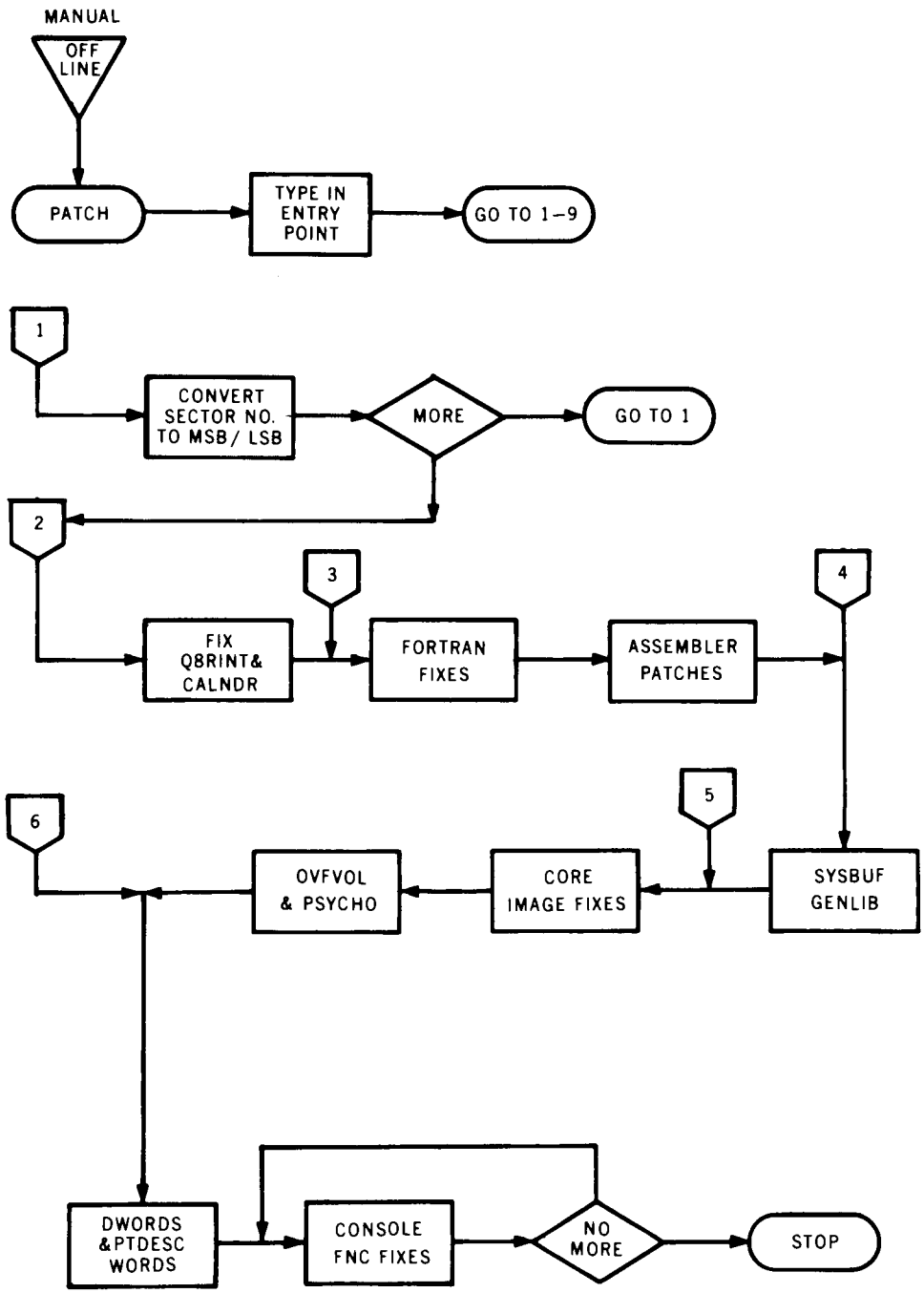


Fig. 67. Logic block diagram of program PATCH.

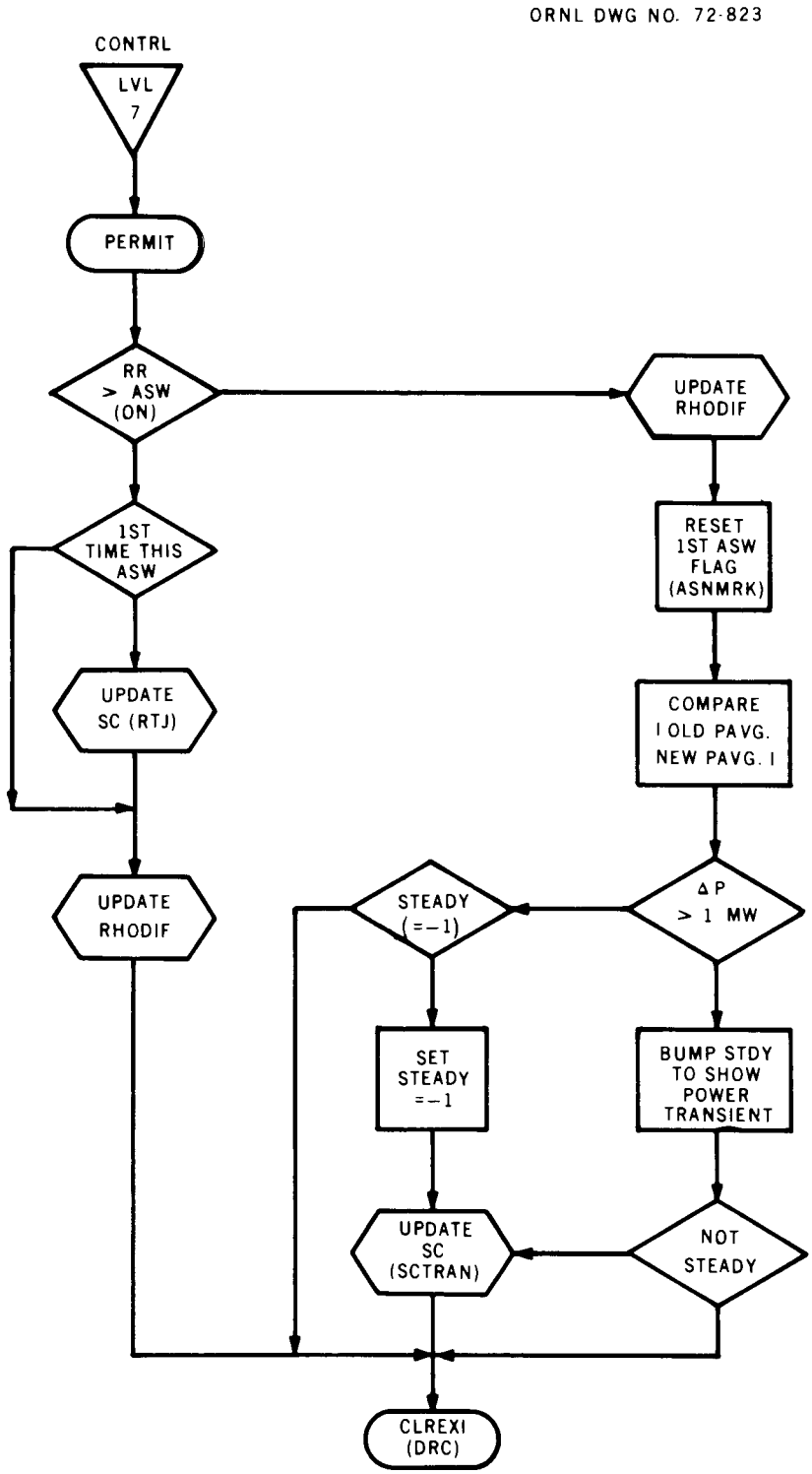


Fig. 68. Logic block diagram of program PERMIT.

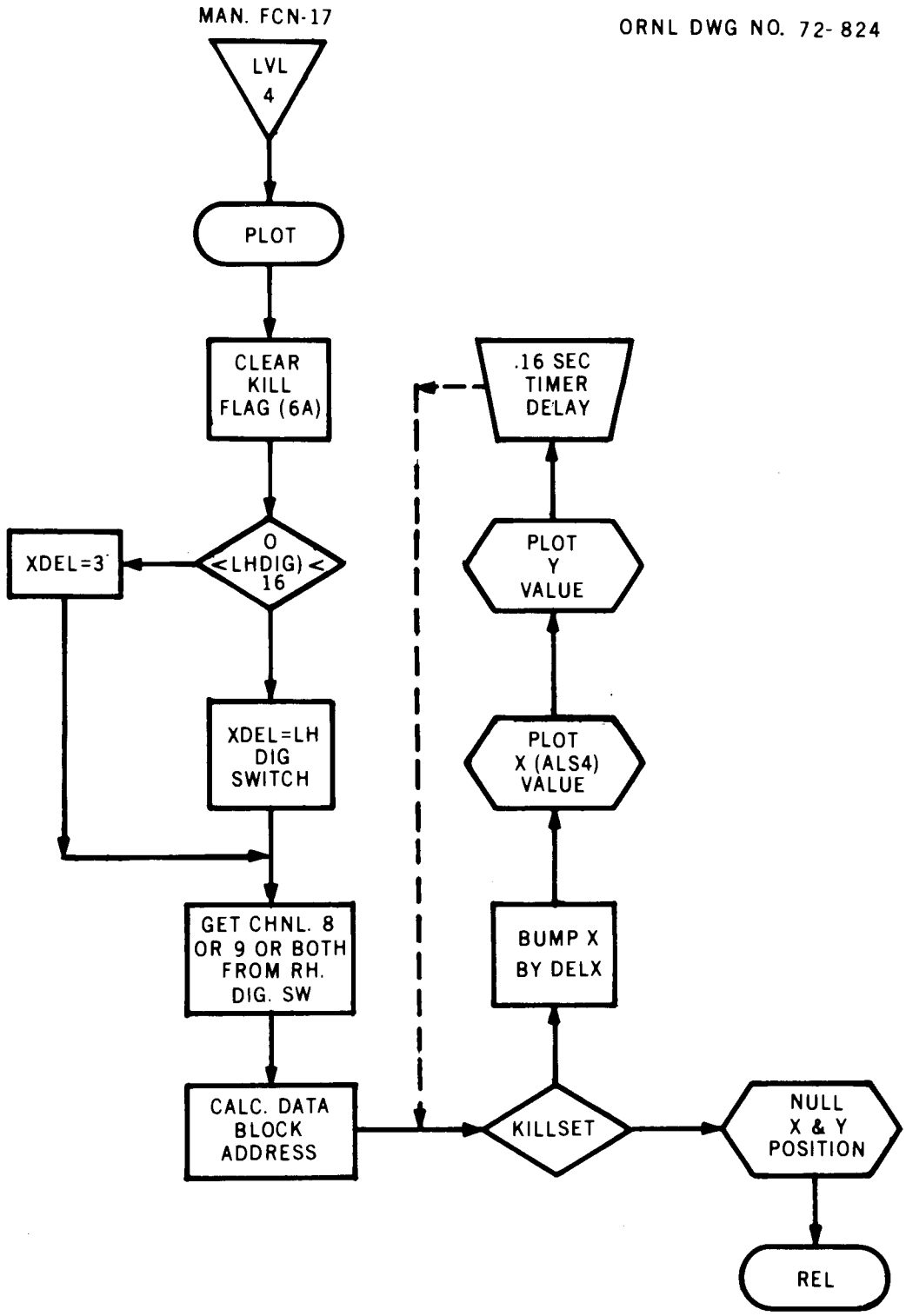


Fig. 69. Logic block diagram of program PLOT.

ORNL DWG NO. 72-814

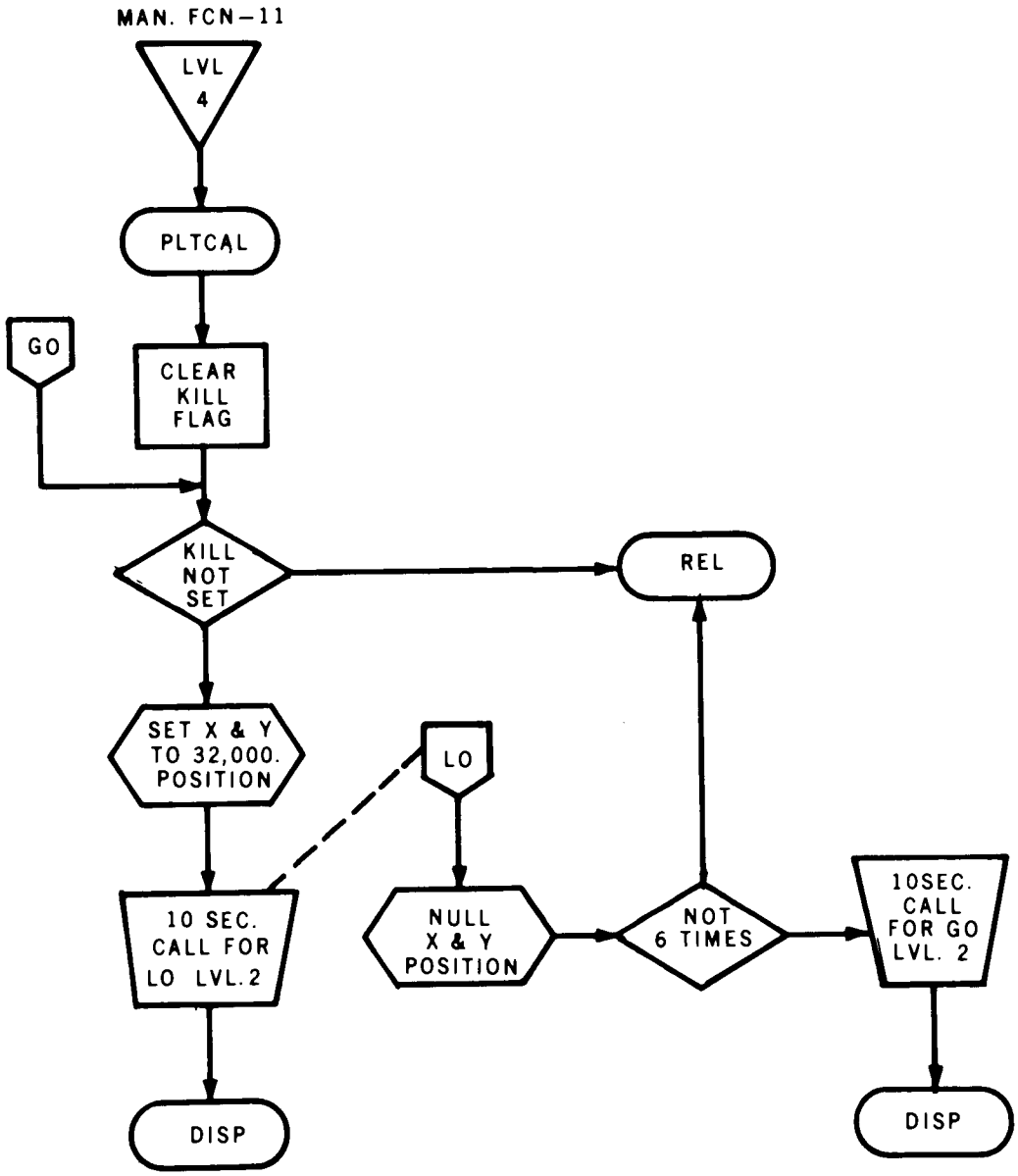


Fig. 70. Logic block diagram of program PLTCAL.

ORNL DWG NO. 72-801

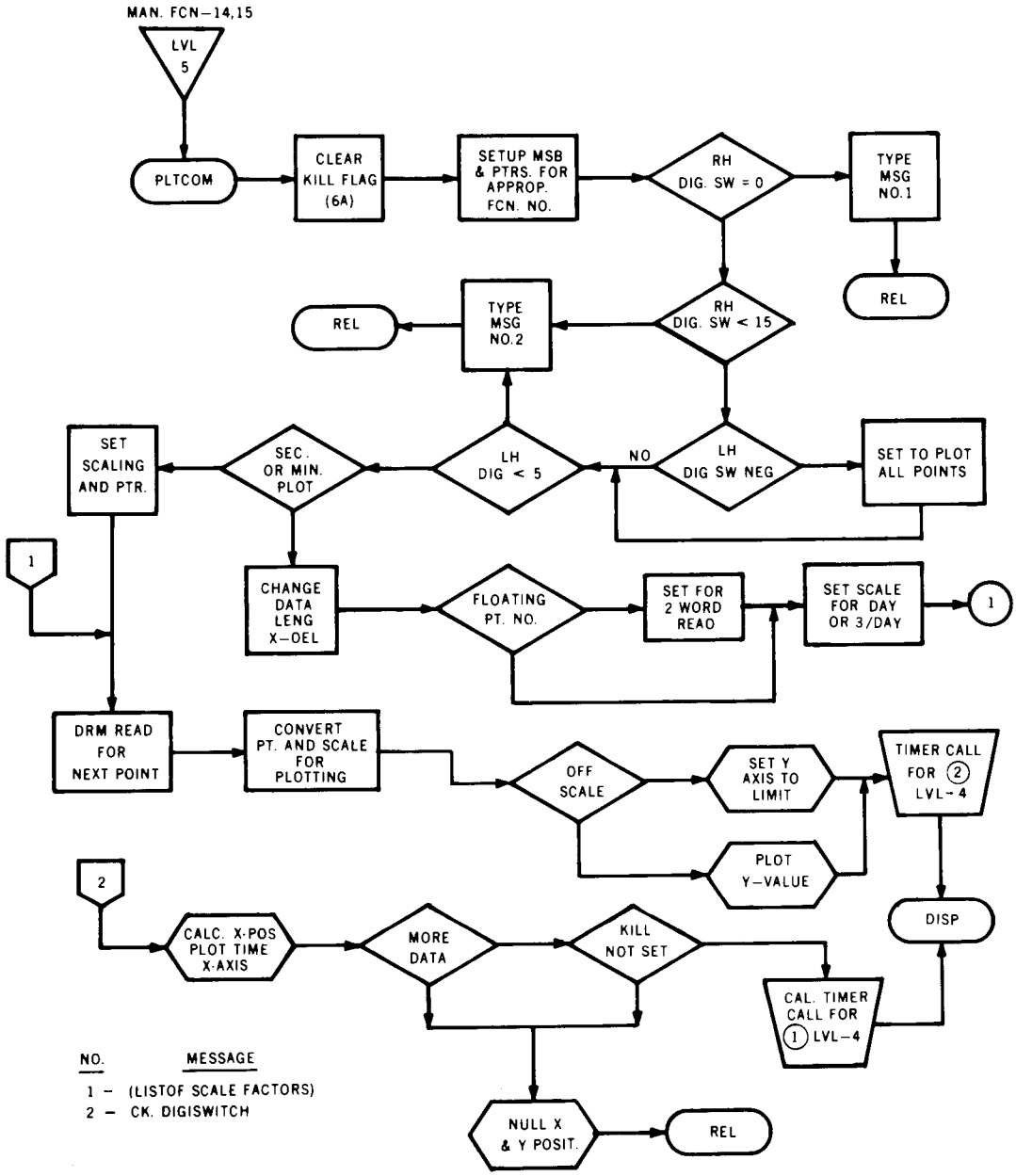


Fig. 71. Logic block diagram of program PLTCOM.

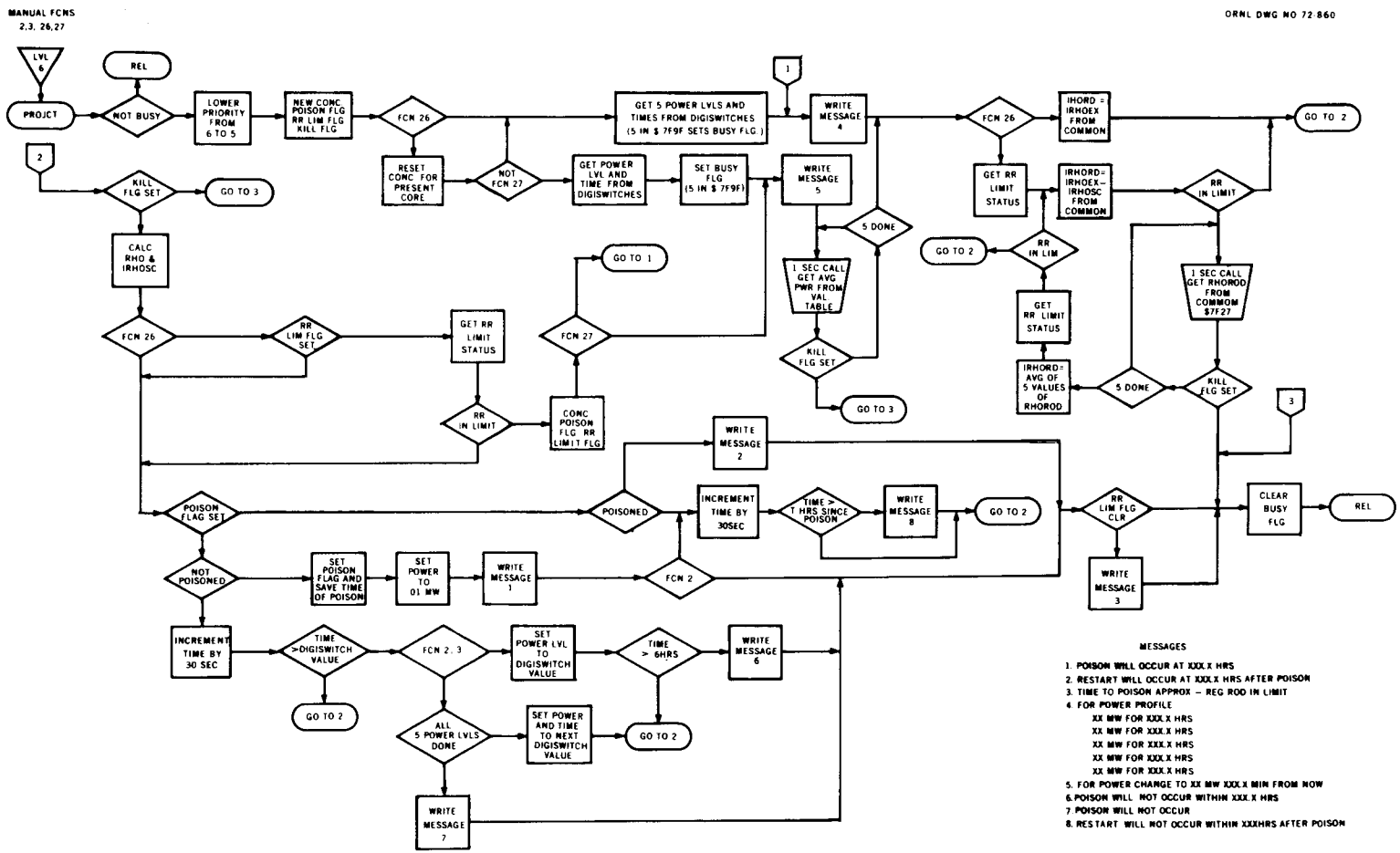


Fig. 72. Logic block diagram of program PROJCT.

ORNL DWG NO. 72-820

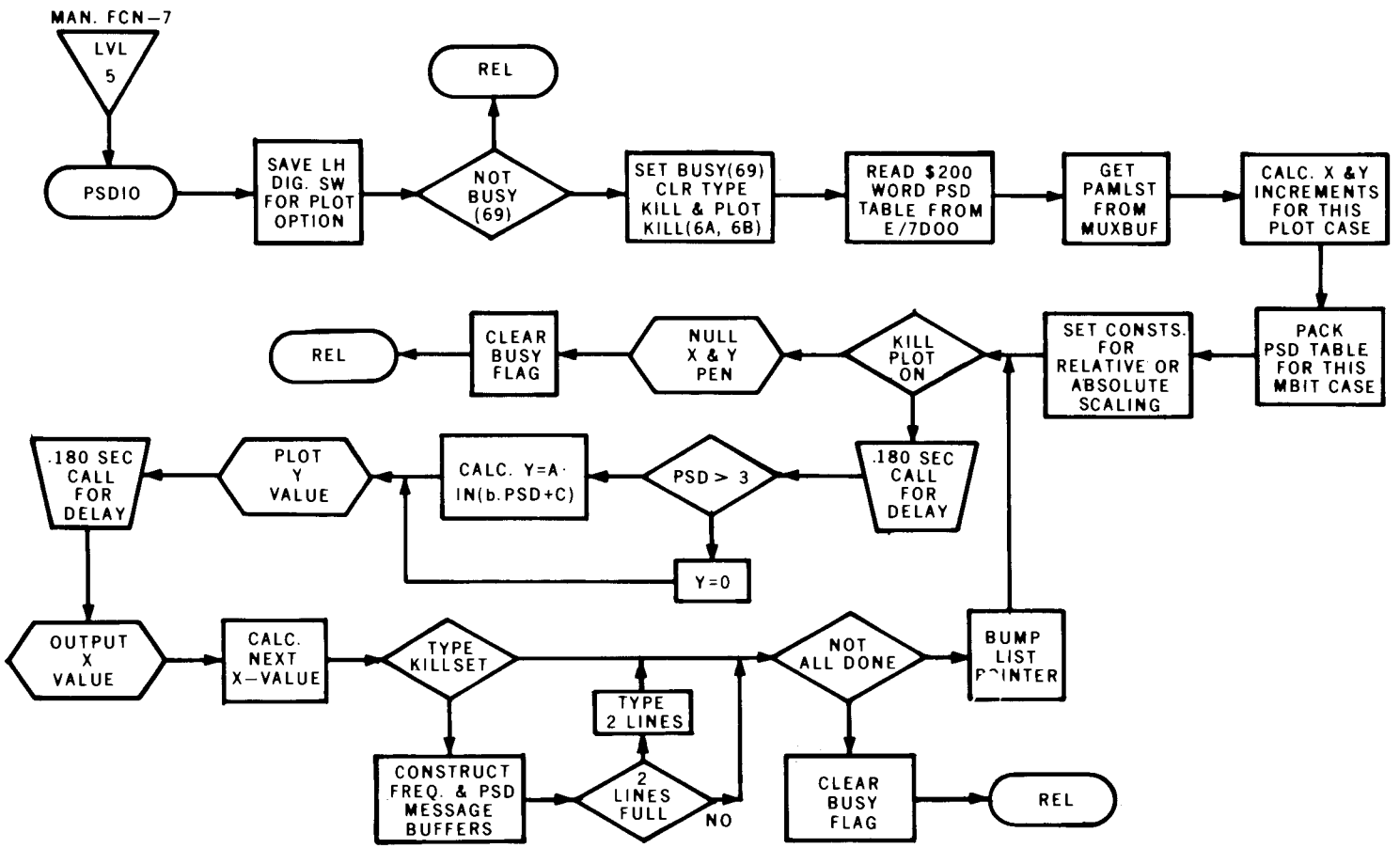


Fig. 73. Logic block diagram of program PSDIO.

ORNL DWG NO. 72-822

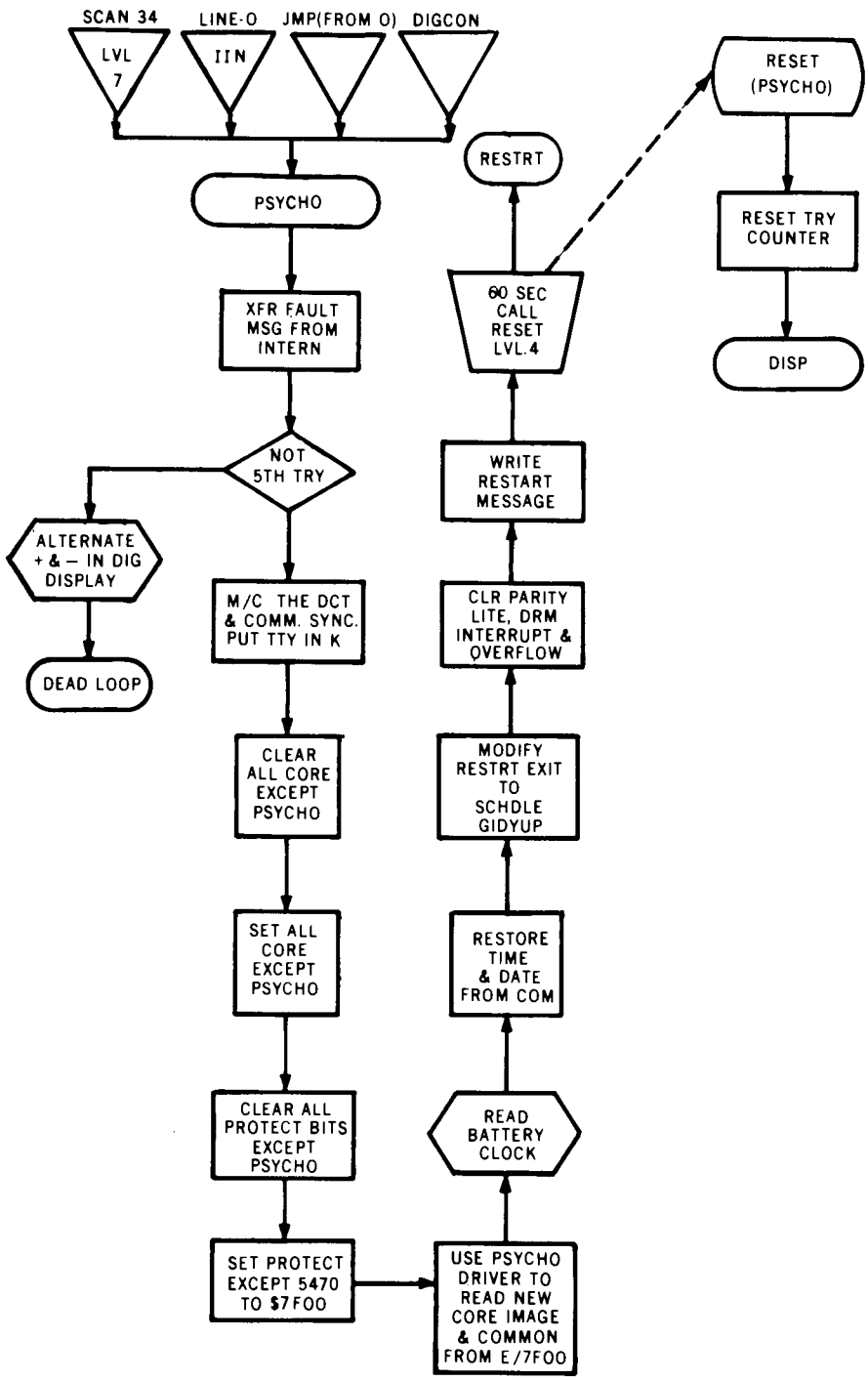
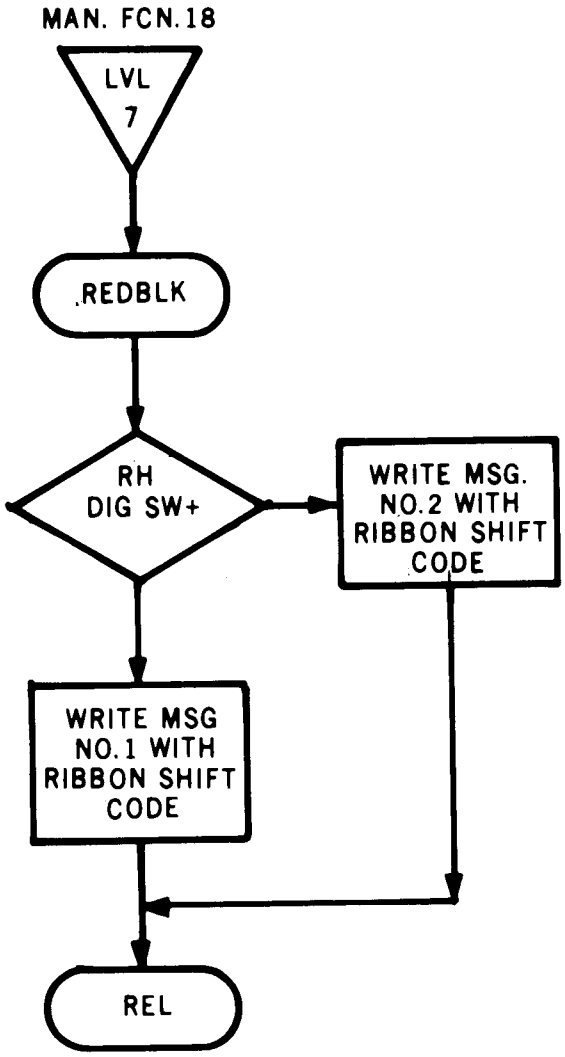


Fig. 74. Logic block diagram of program PSYCHO.

ORNL DWG NO. 72-849



NO.		MESSAGE
1	—	RED TO BLK
2	—	BLK TO RED

Fig. 75. Logic block diagram of program REDBLK.

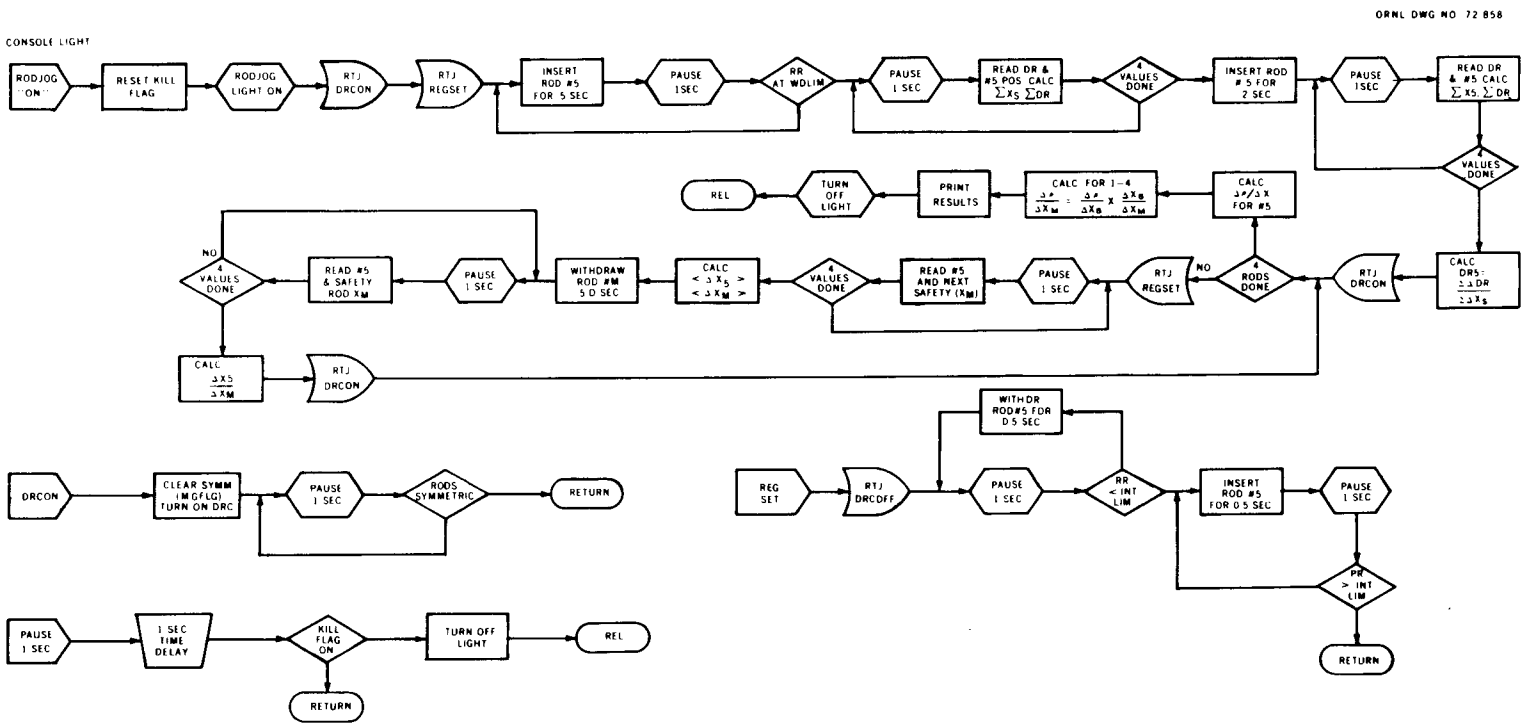


Fig. 76. Logic block diagram of program RODJOG.

ORNL DWG NO. 72-826

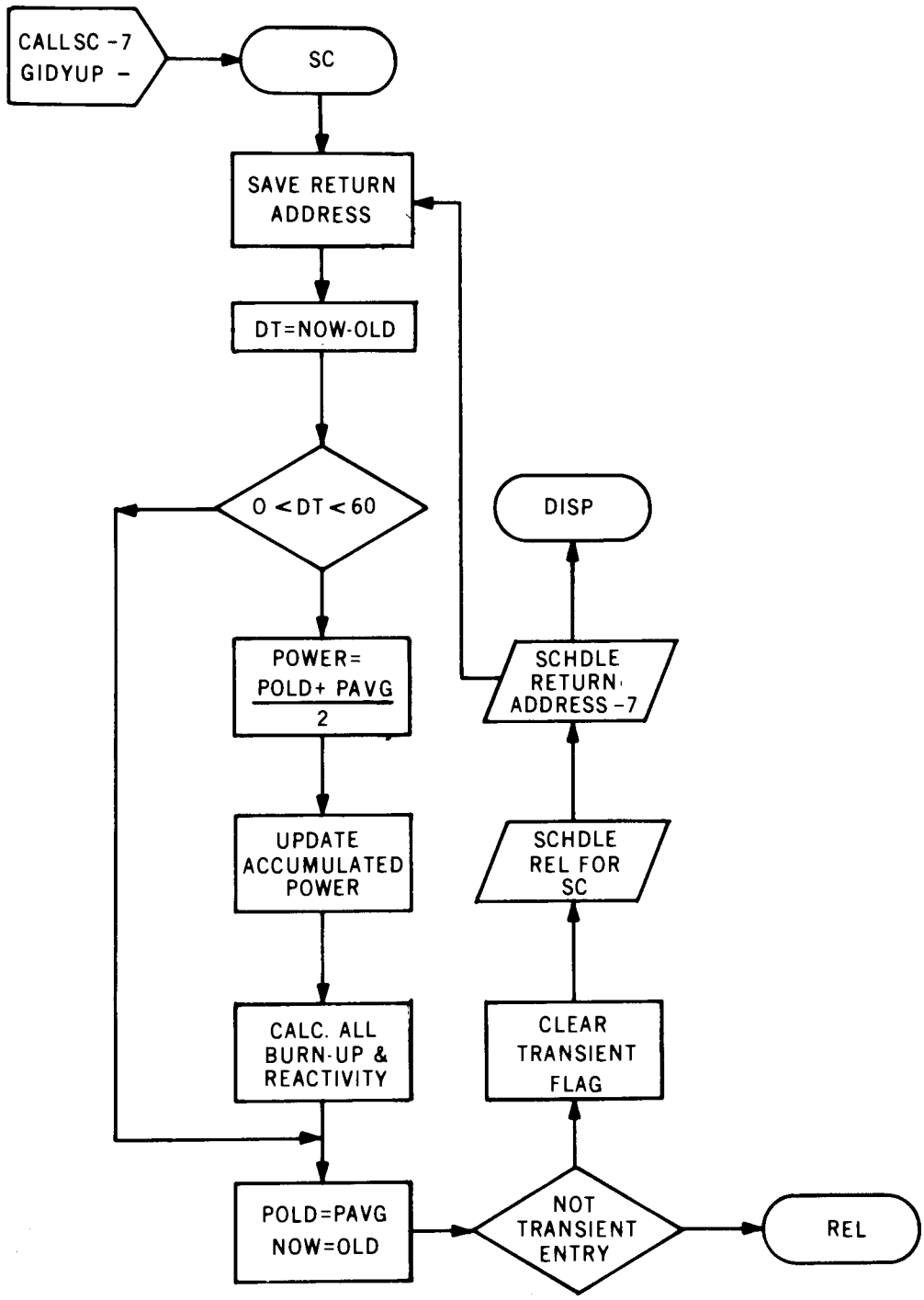


Fig. 77. Logic block diagram of program SC.

ORNL DWG NO. 72-804

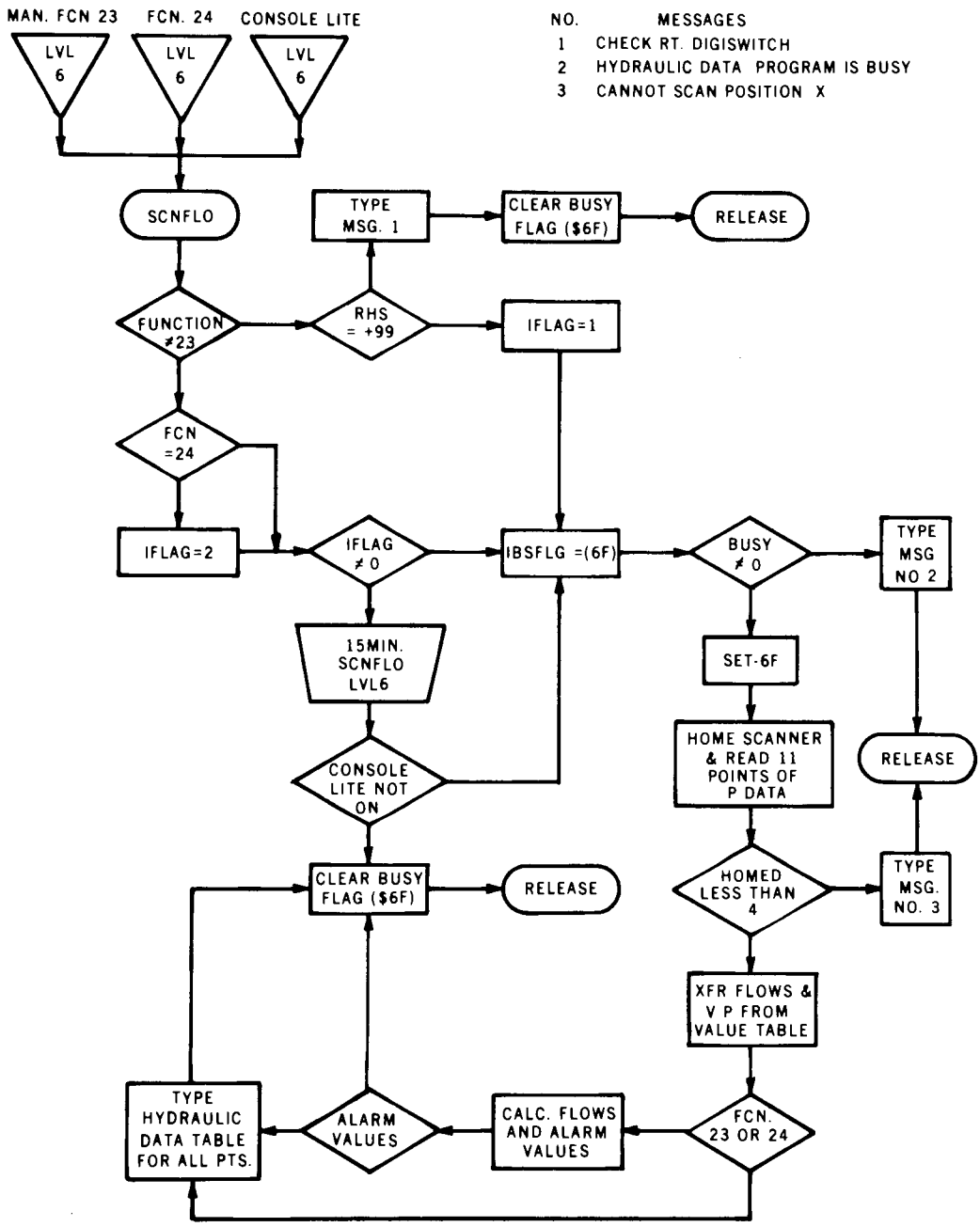
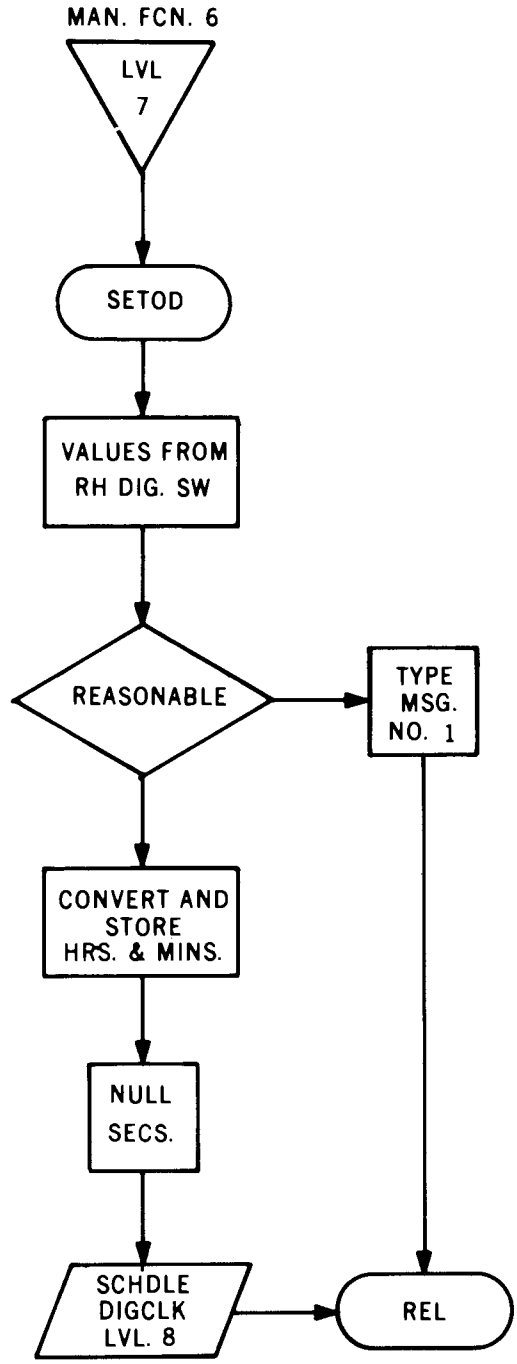


Fig. 78. Logic block diagram of program SCNFLO.

ORNL DWG NO. 72 -813



MESSAGE NO.1 ILLEGAL ENTRY !

Fig. 79. Logic block diagram of program SETOD.

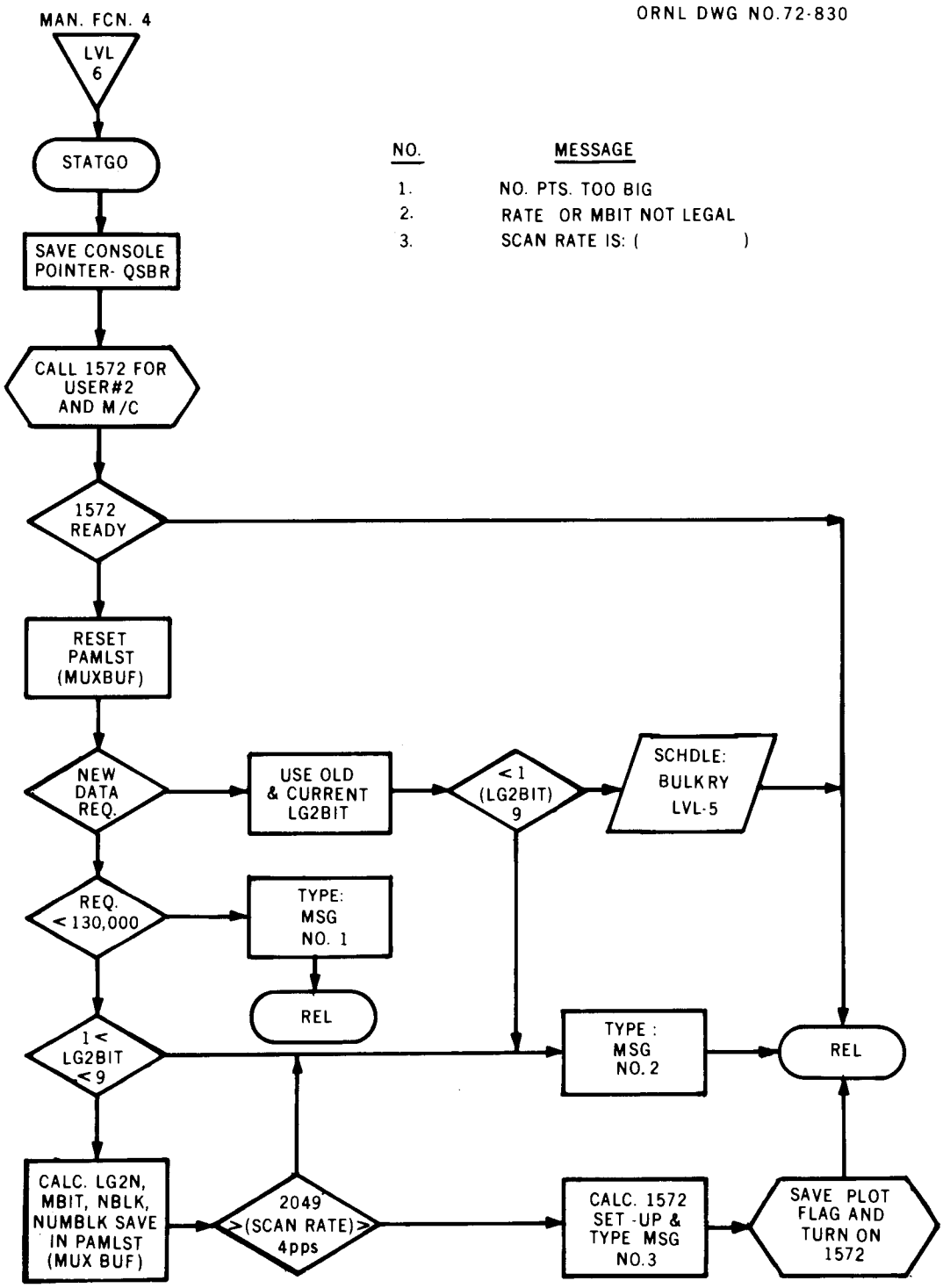


Fig. 80. Logic block diagram of program STATGO.

ORNL DWG NO. 72-841

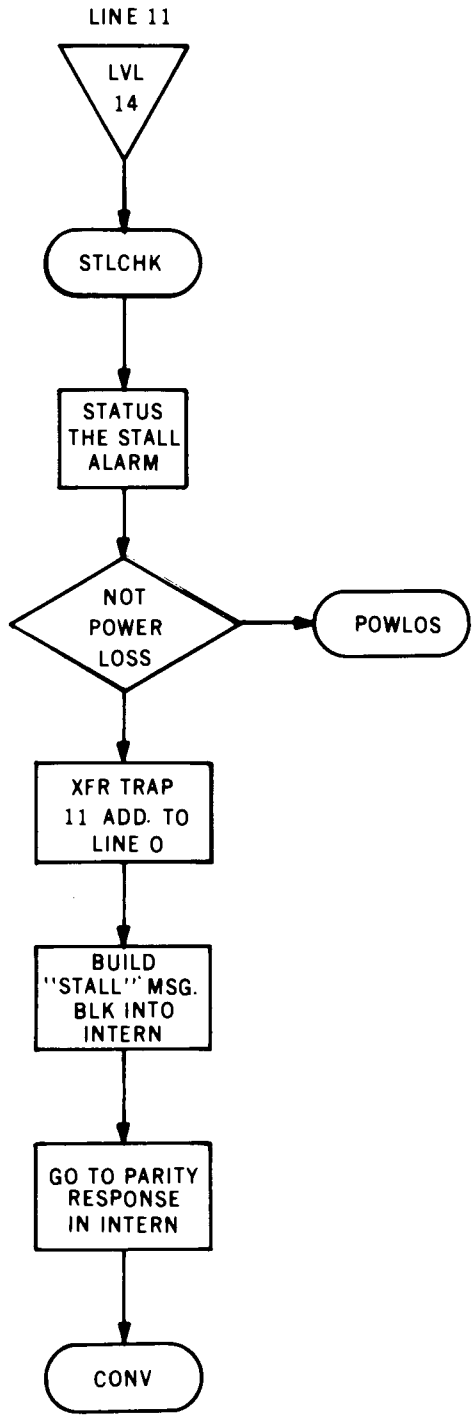
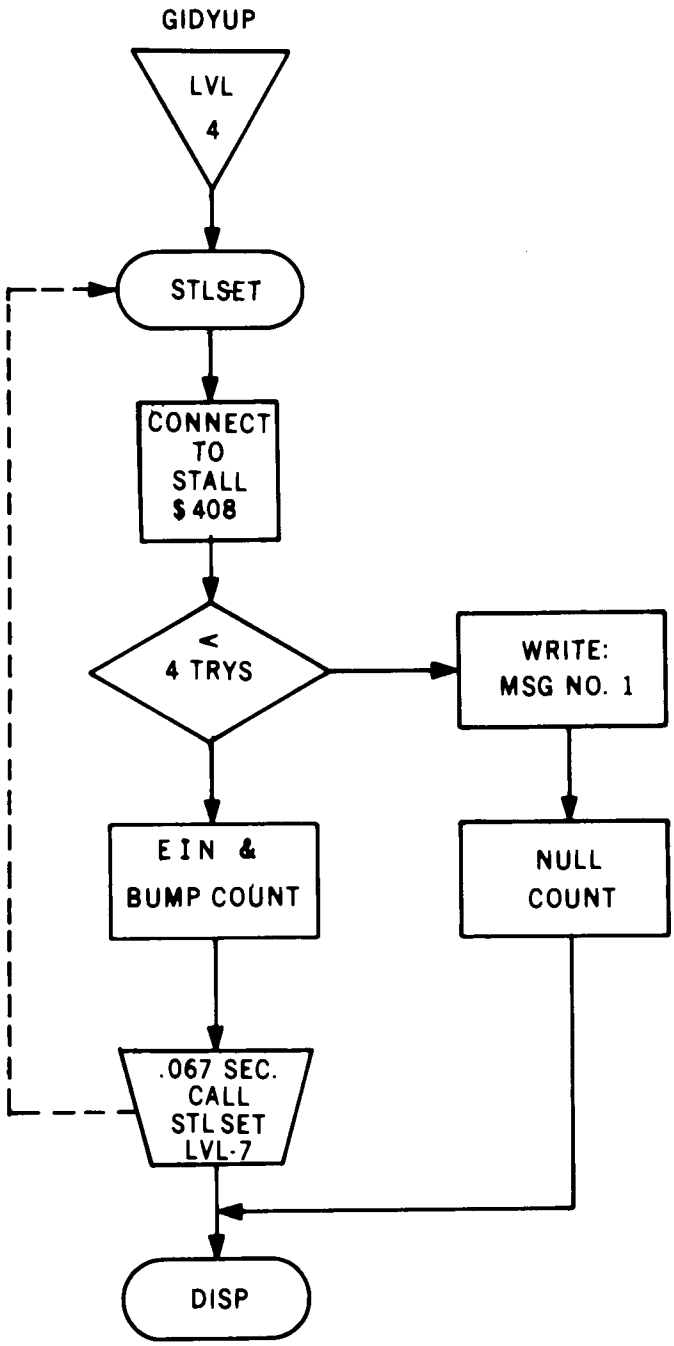


Fig. 81. Logic block diagram of program STLCHK.

ORNL DWG NO.72-852



MESSAGE NO. 1— STLSET FAILED

Fig. 82. Logic block diagram of program STLSET.

ORNL DWG NO. 72-842

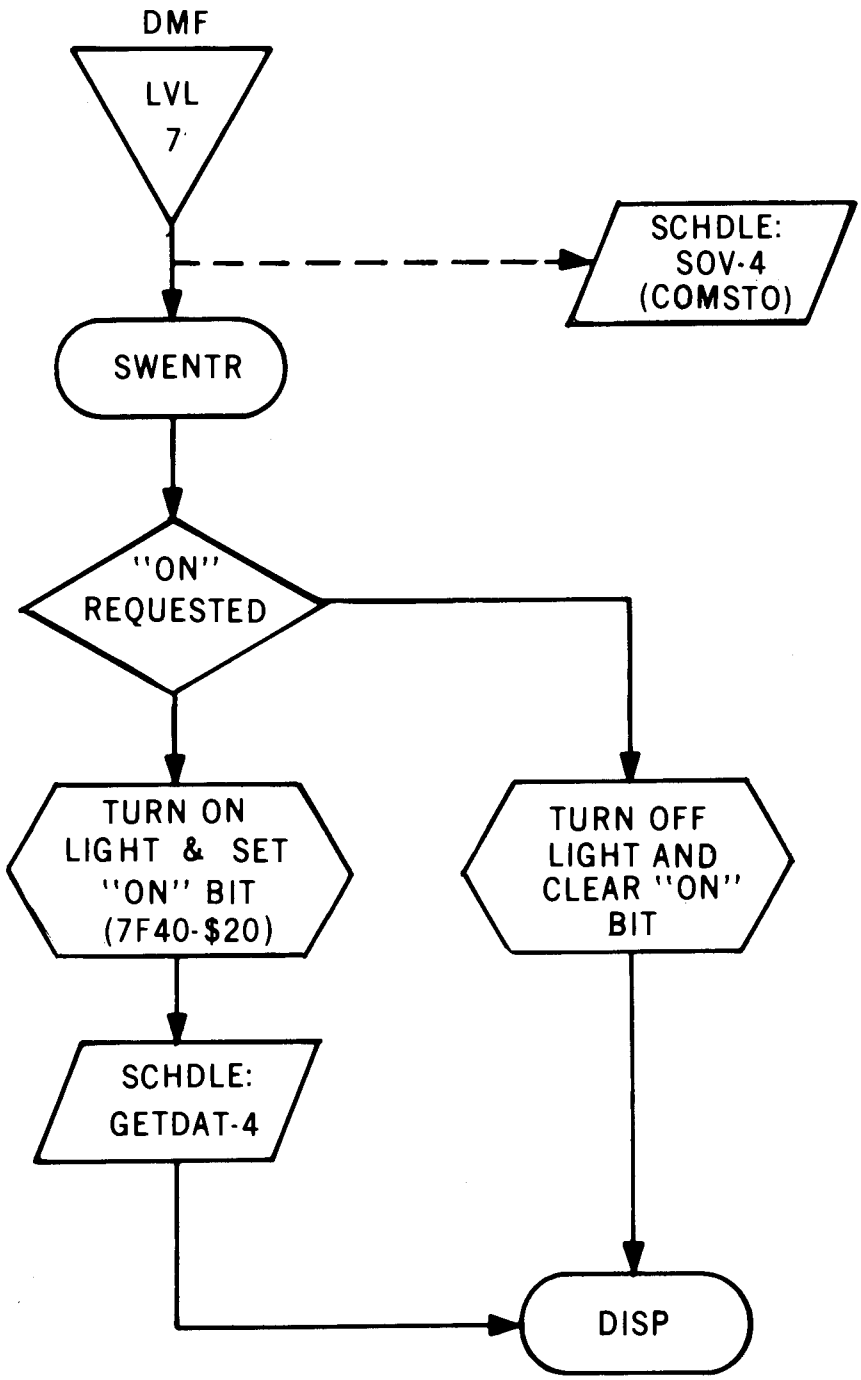


Fig. 83. Logic block diagram of program SWENTR.

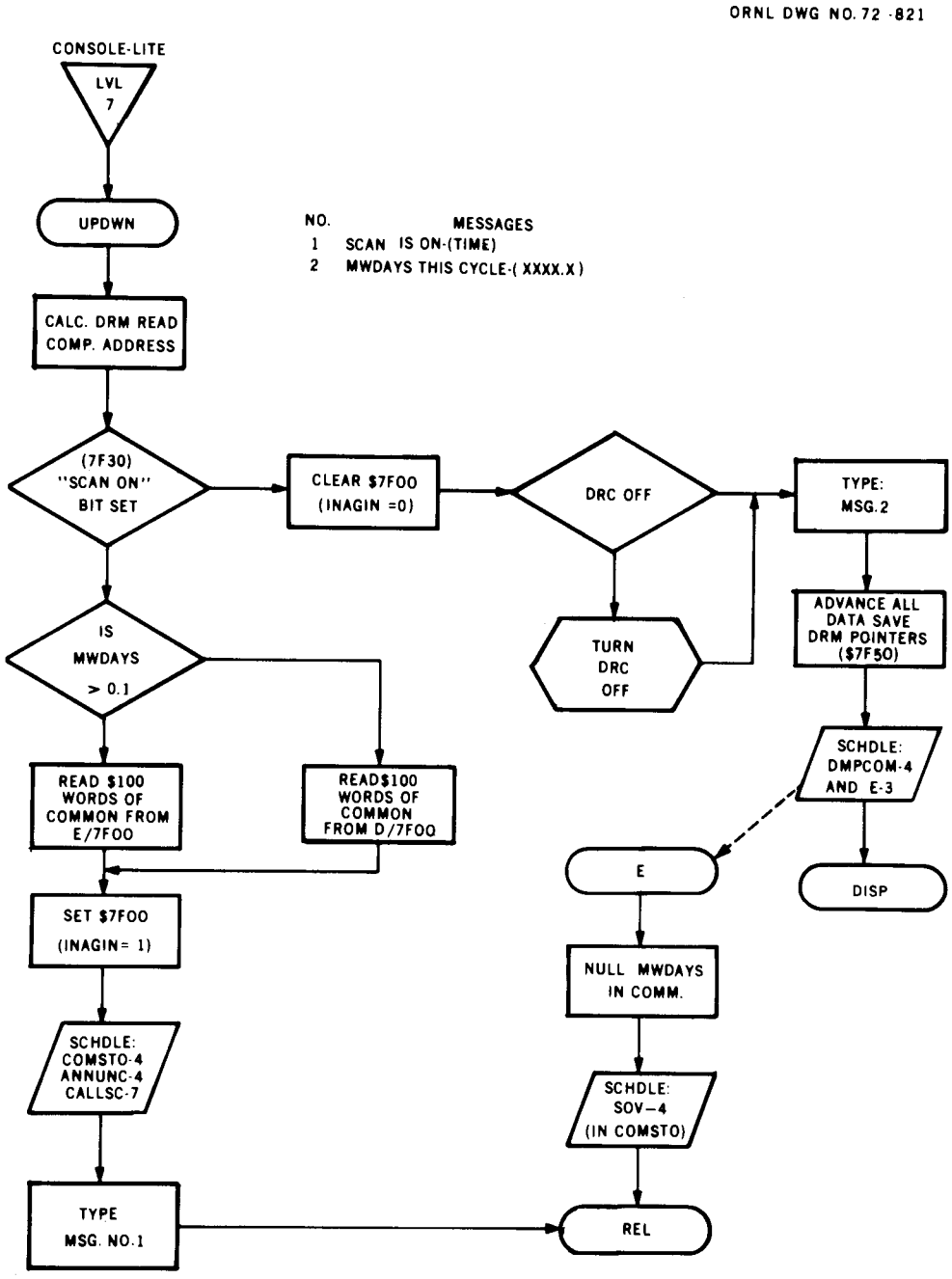


Fig. 84. Logic block diagram of program UPDWN.

4.4 Operating System Loading Procedure

The operating system is loaded initially from relocatable binary paper tape with the aid of a vendor supplied program called the System Initializer (SI). The primary purpose of the SI is to read the tape, link any external addresses, absolutize the relocatable binary and load it at the next available address, and list the program name and address at which it is loaded. In addition to the loading functions described above, SI will read a list of user prepared program names to construct a directory index table that allows mass memory programs to be scheduled directly by their directory name. The directory list is the first tape read during the loading procedure and is typed by the SI following the read command *Y. A typical list is shown in the sample loading document in Table 12.

Since all core resident programs can be called or addressed directly by name, the directory table is needed only for mass-memory programs. The 37 program names following the *YM statements in the example will be assigned to the first 37 programs loaded by separate mass memory load commands (*M) during the system build. In the sample load list shown, the program series starting with program LOAD and ending with ADPRO will be indexed as the first directory program (LOADSD). Since only a single *M command was used to load this series of programs, the SI did not assign each of the programs to a new sector address, nor otherwise treat them as separate programs except to list the name blocks as they are loaded. In a like manner, each subsequent program (or group of programs) loaded by a *M command will be assigned the next available name in the *YM directory list until the list is depleted.

The permanent core resident programs are loaded after the *YM library tape. The *L command is used to designate core resident programs. The SI reads the tape and then lists the program name and the address (in base 16) at which the program is loaded. After all the permanent core programs are loaded, the *M command is used to commence loading the programs that will reside on mass memory. The name of the program is listed, and it is followed by the sector (blocks of 96_{10}) address. Unless a separate *M command is used for each program, a new sector is not used for the first location of each program, and only the sector number for the first program is listed after each program.

After all the *M programs have been loaded, the SPACE program is loaded with an *L command, and the permanent file programs are loaded with the *F command. The

*T command is used to terminate the SI, and the last action of the SI is to type the drum sector number of the beginning address of the permanent core image. In the example cited, the sector number is 0308, which means a binary copy of the operating system core image is located on drum beginning with $0000. Pressing the instruction button on the computer console will transfer the core image from drum to core (see Fig. 85). A field modification has been made to the "instruction" button on the computer console to duplicate this function.

After the SI program is terminated, the system is modified with CHRIS and PATCH to correct known deficiencies and special binary addresses. The PATCH program is an off-line Fortran program to facilitate in the rapid conversion of sector addresses to MSB/LSB addresses and in the modification of selected programs of the system. The operator is required to respond to information requested by the program and is thusly supplied with a list of the actual changes made during the execution of the program. A sample list is given in Table 13.

The program is loaded as a routine off-line program and requests the operator to select the desired entry point. Entry No. 1(not shown in the example) will convert and list the MSB/LSB corresponding to any sector number entered by the operator. Entry points 2 through 8 are used to correct fixed locations within a given program, and this is done by PATCH after the program address is given by the operator. The last step of entry point 8 is used to calculate the proper setting of the drum guard switches. The proper configuration is complicated by the requirement that the switch pattern represent the hexadecimal track address (blocks of 2048 drum words) below which no drum writing shall be allowed. This requires the calculation of the first integer track address above the last system program loaded on drum.

The assembler program is usually the last program loaded; PATCH, therefore, asks for the address of ASSEM and proceeds to calculate and list the first track address above the end of ASSEM. The program then lists the required guard switch settings. In Fig. 7, the guard switches are set to read 000001100101 ($065).

The last important function performed by PATCH is the construction of the table that relates the console function number to a specific program. The table must contain

the MSB/LSB, program length, and function number. These parameters are calculated by PATCH and listed on the load document for future reference.

The detailed step-by-step procedure for loading the operating system is shown in Table 14.

Fig. 85. CDC-1700 computer console.

Table 12. System build loading document

```
SI
*Ø,5

Q
*Y.
*YM,LØADSD,1,JØBENT,2,JØBPRØ,3,JPLØAD,4,JPST,5,JPCHGE,6,JBKI
*YM,JPT13,8,RCØVER,9,LIBEDT,10,MØD1,11,MØD2,12,MØD3,13,MØD4,
*YM,BRKPT,15,RESTØR,16,DFIS,17,CHRIS,18,EDIT,19,GIDYUP,20
*YM,BULKRY,21,PSDIØ,22,GEØF,23,STATGØ,24,CALEN,25
*YM,MIPRØ,26,UPDWN,27,SCANER,28,FRYLIN,29,EDFILE,30,DRMCØM,3
*YM,GDAT,32,ADAT,33,RØDJØG,34,ANNBLK,35,DSC,36,SPARE,37
L,01 FAILED
ACTIØN
QU

Q
*L

Q
*L
LØCØRE   0000
L,01 FAILED
ACTIØN
RP
SYSBUF   0352
CS7FOO
DECLCM   0A14
FLØAT    0A14
DMF      0A30
KILL     0ACA
CØMSTØ   0ADB
MUXBUF   0AF9
HØØKUP   0CAO
CALLSC   0D1A
L,01 FAILED
ACTIØN
RP
DUMBUL   0D41
SCAN34   0E65
LYNE1    0EE9
TRANV    0F19
MAKEQ    0F2A
RW       0F4C
INTERN   0FBF
ALTERN   1013
MØNITØ   1173
FNRBUF   11B5
CØRE     1328
```

Table 12. (continued)

SPACDR	13D1
DISPR	14B9
PARAME	15FF
VØLATI	165D
CMRBUF	167E
CØMMØN	16D1
ØVFVØL	16E3
MANINT	16F4
JBCNCL	1794
BUFFDR	179F
DIAGNØ	18E2
TIMER	18FF
DR1544	197D
DIGØUT	199C
ADCDVR	1A22
PTRDAD	1B75
PUNCDR	1C4B
DRMDRZ	1D1C
SELECT	1E89
TELTYP	2081
TØD	21C5
ØCDEC	2215
DEØCT	2238
FØRTRA	2253
IØCØDE	2375
INITAL	23AB
RSTØRE	23BA
IGETCH	23CA
IPACK	23E4
UPDATE	241B
DECPL	2428
INTGR	244E
SPACEX	247B
HØLRTH	2493
DCHX	24DB
HXASC	2545
AFRMØT	2580
RFRMØT	25AA
AFRMIN	25CC
RFRMIN	25EE
ASCHX	2605
HXDC	2633
FLØTIN	26C8
FØUT	270A
EØUT	2774
EWRITE	2856
AFRØM	2862
RFØRM	287E
HEXASC	289A
HEXDEC	28B2

Table 12. (continued)

ASCII	28CF
DECHEX	28E4
INITL1	2904
FØRMTR	2923
Q8QFI	2AC0
Q8QFL	2ADA
Q8QFX	2B03
Q8QIØ	2B3A
REENTE	2C4B
Q8EXPN	2C5C
Q8PRMS	2CF4
IFALTR	2D08
SIGN	2D20
FXFLR	2D44
EXPPRG	2DB2
SQRTF	2E4D
LNUPRG	2EBD
TANH	2F36
SINCØS	2FA3
ARCTPG	3061
FLØATR	30F4
SQRØØT	3303
Q8AB	3331
DSPLY	3342
CØNV	3353
DCØRE	3394
SPECS	33BB
DWF	33DD
DWH	3425
DWC	34B8
CPSC	35B4
HX2ASC	3621
RDPT	3650
ØRALC	368B
ØRMI	37FD
ANNUNC	38FE
PICMID	39C1
PØWCAL	3A28
RHØRØD	3A72
DIGCØN	3B5E
STLSET	3BBD
RHØCAL	3BE8
DACDRI	3C01
DRI72	3C41
STLCHK	3CD1
SWENTR	3CEF
DIFPLT	3D07
LØGCHK	3D39
DIGCLK	3D3C
DMPCØM	3D69

Table 12. (continued)

```
CØNTRL   3DB4
PERMIT   3F15
DRC      3F3C
PSYCHØ   4117
L,01 FAILED
ACTIØN
QU

Q
*M
LØAD     0001
BRANCH   0001
LIDRIV   0001
LCDRIV   0001
LMDRIV   0001
LLDRIV   0001
SCAN     0001
CHPU     0001
CØNVRT   0001
ADJØVF   0001
TABSCH   0001
TABSTR   0001
LSTØUT   0001
LINK1    0001
LINK2    0001
CØREXT   0001
DPRADD   0001
LØADER   0001
NAMPRØ   0001
RBDBZS   0001
ENTEXT   0001
XFRPRØ   0001
HEXPRØ   0001
EØLPRØ   0001
ADPRØ    0001
*M
JØBENT   0020
T7       0020
T11      0020
T3       0020
*M
JØBPRØ   0025
T5       0025
PRØTEC   0025
L,01 FAILED
ACTIØN
RP
ØNE      0025
TWØ      0025
*M
```

Table 12. (continued)

```
        SUBPRØGRAMS WITH THD FØLLØWING ENTRY PØINT
  NAMES HAVE NØT BEEN LØADED DURING *M LØAD.

THREE

ERRØR    C

Q
*M
JPLØAD  0032
L,CI FAILED
ACTIØN
CU

Q
*M
JPST    0037
*M
JPCHGE  0039
ASCHEX  0039
*M
JBKILL  003C
*M
JPTI3   003E
TI3     003E
*M
RCØVER  0044
ØUTSEL  0044
DMPCØR  0044
MASDMP  0044
*M
LIBEDT  004D
*M
UTILIB  0056
*M
PLINSN  0060
*M
FILE    0069
*M
GENLIB  0074  84690303 (ØRNL
*M
BRKPTD  0079
SIFT    0079
BIASCI  0079
RETJMP  0079
JUMPTØ  0079
ENTER   0079
ENTCØR  0079
PRTREG  0079
```

Table 12. (continued)

```
SETBRP   0079
TERMIN   0079
DMPCØB   0079
MASDMB   0079
RDSUME   0079
*M
RESBUF   0088
*M
CØNINT   008A
*M
CHRIS    008F
L,01 FAILED
ACTIØN
CU

Q
*M
EDIT     00A1
*M
GIDUP    00AC
*M
BULKRY   00B1
*M
PSDIØ    00DD
*M
GEØF     00E3
*M
STATGØ   00E5
*M
CALDR    00E8
*M
MIPRØ    00EC
SETTIM   00EC
SETDAT   00EC
*M


        SUBPRØGRAMS WITH THD FØLLØWING ENTRY PØINT
  NAMES HAVE NØT BEEN LØADED DURING *M LØAD.

SPARE2
SPARE1

ERRØR    C

Q
*M
UPDWN    00F0
*M
```

Table 12. (continued)

```
SCNFLØ   00F2
*M
FRYLIN   0103
*M
EDFILE   0105
*M
DRMCØM   010A
*M
GETDAT   010B
*M
GADAT    010D
*M
RØDJØG   0110
*M
ANNBLK   0115
L,01 FAILED
ACTIØN
CU

Q
*M
SC       0119
*M
DMRCAL   0120
*M


        SUBPRØGRAMS WITH THE FØLLØWING ENTRY PØINT
  NAMES HAVE NØT BEEN LØADED DURING *M LØAD.

RØDCAL

ERRØR    C

Q
*M
PTDESC   0121
*M
DWØRDS   012A
*M
DIGITA   012C
*M
LØG      012E
*M
PRØJCT   0136
*M
FUN4     0147
*M
DISCEL   0148
*M
```

Table 12. (continued)

```
SETØD    0149
*M
FLTIME   014A
*M
FRYKIL   014E
*M
DWNTIM   014F
*M
PLTCAL   0151
*M
AVGDAT   0152
*M


        SUBPRØGRAMS WITH THD FØLLØWING ENTRY PØINT
  NAMES HAVE NØT BEEN LØADED DURING *M LØAD.

SEBFR

ERRØR    C

Q
*M
DATPLT   0159
*M
PLTCØM   015A
*M
DRMXFR   015F
*M
PLØT     0160
*M
REDBLK   0162
*M
DR       0163
*M
DRKILL   016A
*M
BULPLT   016B
*M
CKINT    016C
*M
TCØN     016E
L,01 FAILED
ACTIØN
CU

Q
*L,44C0
SPACE    44C0
```

Table 12. (continued)

```
L,01 FAILED
ACTIØN
QU

Q
*F


        SUBPRØGRAMS WITH THE FØLLØWING ENTRY PØINT
  NAMES HAVE NØT BEEN LØADED DURING *L LØAD.

MANFLG
AUTFLG
CVTF
JPRETN
GNEXT
SELMAN
HRDERR
HPUNER
HERRØR
CØMCØN
CØMINI
ARDA4
SNAPE
DWBFR
CARD29
PØWLØS
IPRØC1
DUMALT
T30
T29
T28
T27
T26
T25
T24
T23
T22
T21
T20
T19
T18
T17
T16
T14
T13
T11
T7
T5
```

Table 12. (continued)

```
T3
MSGINT
TRIM

ERRØR    D

Q
*F
FTN
Q8QINI
Q3QEND
Q8CMP
Q8RWBU
Q8ERRM
Q8DFIØ
Q8QX
Q8QUNI
Q8FGET
Q8MAGT
TAPCØN
IØCK
PSSTØP
Q3PAND
Q8EXP9
Q8EXP1
Q8IFRM
Q3FS
Q8TRAN
Q8AB
SIGN
EXPPRG
SQRTF
LNUPRG
TANH
SINCØS
ARCTPG
Q8EXPN
FLØAT
Q8PRMS
IFALT
FXFL
L,01 FAILED
ACTIØN
QU

Q
*T
0308
PP
*
MI
S/8077
ER
```

Table 13. PATCH program messages

```
    PATCH    5470
    DRUM     6076
L,02 FAILED 02
ACTIØN
CU
J
*X,N
VERIFY SETUP- (SEE PATCH CØMMENTS)
EACH DRM WRITE IS DØNE BY TYPING- 2 AFTER ØK-
START EXECUTIØN AT ENTRY NØ.- 2
***ENT.-2
TYPE Q8RINT AND AVGDAT SECTØR/01EF 0166
 285    E
 285 1009
ØK- 2
  CE 2020
  CE 0000
ØK- 2
***ENT.-3
TYPE FØRTP1 SECTØR(XXXX)/ 0474
CØMPILER AND ASSEMBLER PATCHES
10CA 1818
10CA 1810
ØK- 2
 A97 C000  800
 A97 5800  C08
ØK- 2
 AC2  174
 AC2  BE6
ØK- 2
 ABF  9F9
 ABF  9FA
ØK- 2
10B2 64D9
10B2 8057
ØK- 2
115 B 64D9
115B 8057
ØK- 2
 323  9F9
 323  9F5
ØK- 2
 325 54F4
 325 1809
ØK- 2
  EF  9FD
  EF  9F5
ØK- 2
XFR FINFIX FRØM F/2330 TØ FØRTP1 + $16A0
   F 2330 0000
   3 4220 0000
```

Table 13. (continued)

```
ØK- 2
PASS 1 PATCHES
  5 B  76C
  5 B  FF2
ØK- 2
 565  135
 565 1806
ØK- 2
 5A1  A00 6138
 5 A1 5800  224
ØK- 2
 5A9 7E93
 5 A9 8057
ØK- 2
XFR ASMFIX FRØM F/2360 TØ PASS1+7C6
   F 2360 0000
   5 74A6 0000
ØK- 2
***ENT.-4
FØR ALL 3 WØRD INPUTS USE XXXX XXXX XXXX
***ENT.-5
TYPE SYSBUF CØRE ADD., GENLIB AND CALNDR SECT./ 0339 0074 00EA
  A4   13
  A4  339
ØK- 2
  E0   13
  E0  339
ØK- 2
  29 18FC
  29 1009
ØK- 2
***ENT.-6
TYPE SECTØR ØF FLØW AND SCANDT/ 0125 00E3
  55    1  2E0
  55 0000 6DE0
ØK- 2
***ENT.-7
TYPE MUXBUF ADD. AND BULKRY SECTØR/ 0AE0 00B1
BULKRY FIXES AT 206,39,19F,212,217,ARE
 206 7FFF 7FFF
 206  AEE  AEF
ØK- 2
  39 7FFF
  39  AE4
ØK- 2
 19F 54A9
 19F  AE4
ØK- 2
 212 7FFF
 212  AE9
```

Table 13. (continued)

```
ØK- 2
 217 7F41
 217 7F42
ØK- 2
***ENT.-8
TYPE CØRE IMAGE SECTØR- 031D
TYPE CØRE ADD. ØF ØVFVØL,PSYCHØ AND SPACE-16C1 40F0 44C0
41 4B   22   40
41 4B   25  2ED
ØK- 2
4150   2A  380
4150   2D  7A0
ØK- 2
16CC 18FF
16CC 1000
ØK- 2
0000 18FF
0000 1465
ØK- 2
  57 0000
  57 7E93
ØK- 2
  F9    5
  F9    5
ØK- 2
TYPE ADD ØF DWC AND DUMBUL- 3491 0D1A
3540 54F4  500
   9 0000
3540 1802  500
5400  D1A
ØK- 2
355D 54F4  300
7FBE 0000
355D 1802  300
5400  D1A
ØK- 2
TYPE DIR INDEX ØF BULKRY (I2)-21
DIR LENGTHS CHANGED AS FØLLØWS
 2BF 1040
 2BF 2440
ØK- 2
 2C6  22D
 2C6  42D
ØK- 2
 2CD   66
 2CD  4C0
ØK- 2
 2D4   8B
 2D4  18B
Ø - 2
```

Table 13. (continued)

```
TYPE ASSEM SECT AND SPACDR CØRE ADD/0867 13AA
SET GUARDS AT   66  881 IS SWAP SECTØR
140A  3FF
140A  881
ØK- 2
  C1  86C
  C1  8F3
ØK- 2
***ENT.-9
TYPE DWØRD,PTDESC SECTØRS/0123 011A
  80 0000 0000
  80 0000 6D20
ØK- 2
  82 0000 0000
  82 0000 69C0
ØK- 2
TYPE SECT CØRRØSPØNDING TØ EA CØNSØLE FN. NØ.
 0 =  0138
          0000  7500
 1 =  013A
          0000  75C0
 2 =  0142
          0000  78C0
 3 =  0142
          0000  78C0
 4 =  015B
               1   220
 5 =


      015C
               1   280
 6 =  015D
               1   2E0
 7 =  00DD
          0000  52E0
 8 =  015E
               1   340
 9 =  0162
               1   4C0
10 =  0163
               1   520
11 =  0165
               1   5E0
12 =  0166
               1   640
13 =  016E
               1   940
```

Table 13. (continued)

```
14 =  016F
                1    9A0
15 =  016F
                1    9A0
16 =  0174
                1    B80
17 =  0175
                1    BE0
18 =  0177
                1    CA0
19 =  0178
                1    D00
20 =  017F
                1    FA0
21 =  0180
                1   1000
22 =  0186
                1   1240
23 =  00F4
           0000  5B80
24 =  00F4
           0000  5B80
25 =  0181
                1   1060
26 =  0142
           0000  78C0
27 =  0142
           0000  78C0
28 =  0000
0000 8000 0000
DWØRDS AS FIXED BY ALL THE ABØVE CHANGES ARE XFRED TØ DM BY TYPING- 2
2
TYPE FN. NØ. FØR SINGLE CØNSØLE CHANGES (I2)- 29
29 =  0138
           0000  7500
7500 0000 2044
DWØRDS AS FIXED BY ALL THE ABØVE CHANGES ARE XFRED TØ DM BY TYPING- 2
2
TYPE FN. NØ. FØR SINGLE CØNSØLE CHANGES (I2)- 99
HAPPY CØMPUTING
STØP
J
```

Table 14. Operating system loading procedure

```
*****************************************************************
*   REV. NØV.11,1971
*
                LØADING THE ØAK RIDGE ØPERATING SYSTEM
*****************************************************************
*
*
        1. USE CHRIS TØ LØAD THE PRE-LØAD TAPE ØN DRUM F/17A0-5000.
*             (THIS TAPE LØADS UTØPIA AT F/17A0,FTNFIX AT 2330,
                ASMFIX AT 2360, CHRIS AT 2400 AND THE SYSTEM INITLZR
                AT F/3C00.

         2. DISCØNNECT THE STALL ALARM CABLE. USE CHRIS TØ TMM CURRENT
          SYSTEM TØ DM'S. 7,8,9,A,B,&C. USE CHRIS TØ READ THE
         SYSTEM INITILIZER INTØ CØRE. RMM,6000,7F00,F,3C00/

        3. DRØP ALL DRUM GUARDS, THE PRØTECT SW, DØ A MASTER CLEAR.
           SET P=$6000 AND RAISE THE RUN SWITCH TØ START THE INITALIZER.

        4. "SI" SHØULD TYPE ØN TTY. RESPØND WITH:

           *Ø,5    (ALL RESPØNSES MUST END WITH A RETURN).

        5.  " Q" SHØULD TYPE ØUT.
            MØUNT THE *YM TAPE IN THE READER.(NØTE, THE *YM TAPE MUST
            LØAD BULKRY,PSDIØ,AND GEØFF CØNTIGIØUSLY IF THE PATCH
           PRØGRAM IS TØ ACCURATELY FIX THE DIRECTØRY LENGTHS.
              ALSØ NØTE THE *YM LINES DØ NØT END WITH A CØMMA)

        6. TYPE:  *Y.   (NØTE THE PERIØD)

        7. THE TAPE SHØULD BE READ AND LISTED AND SHØULD FINISH WITH:

              L,01 FAILED
              ACTIØN

        8. RESPØND WITH: CU

        9." Q" WILL TYPE ØUT , RESPØND WITH: *U

        10. "Q"  WILL TYPE ØUT, RESPØND BY LØADING 1ST *L TAPE AND

           TYPE: *L

        11.  WHEN THE TAPE IS LØADED THE TTY WILL TYPE:
             L,01 FAILED
             ACTIØN
```

Table 14. (continued)

12.LØAD REMAINING *L TAPES BY RESPØNDING TØ ACTIØN WITH: RP

13. AFTER THE LAST CØRE RESIDENT TAPE HAS LØADED RESPØND TØ " ACTIØN" BY TYPING :CU (NØTE THAT THE LAST PRØGRAM LØADED DØES NØT EXTEND TØ $44CO).

14. Q WILL TYPE ØUT. LØAD THE FIRST *M TAPE AND TYPE : *M

(NØTE: THE *M TAPES WILL LØAD UNDER THE DIRECTØRY NAME LISTED IN THE *YM LIST ABØVE. EACH NEW *M WILL START ØN A SECTØR LISTED NEXT TØ THE PRØGRAM NAME.)

(IN THE EVENT UNPATCHED EXTERNALS ARE DETECTED DURING THE LØADING ØPERATIØN, A MESSAGE WILL BE TYPED AND ACTIØN WILL BE REQUESTED.THIS WILL BE A "Q" TYPE ØUT.IF LØADING IS TØ BE CØNTINUED RESPØND WITH: *M

15. WHEN THE TAPE IS LØADED, ACTIØN IS REQUESTED.TYPE:CU A "Q" WILL BE TYPED.

16. CØNTINUE LØADING MØRE TAPES BY TYPING: *M

17. AFTER THE LAST *M TAPE HAS LØADED TYPE: CU

18. MØUNT PRØGRAM "SPACE" A-TAPE AND TYPE: *L,44CO (THIS FØRCES SPACE TØ LØAD AT $44CO, THERFØRE VERIFY FRØM THE ADD ØF THE LAST *L PRØGRAM THAT SPACE WILL NØT ØVER-WRITE SØME ØTHER PRØGRAM.)

19. IN RESPØNSE TØ THE ACTIØN REQUEST TYPE: CU Q SHØULD TYPE-ØUT.

20. MØUNT THE (FTN *F) TAPE

TYPE: *F
A LIST ØF ALL CØRE RESIDENT UNPATCHED EXTERNALS WILL BE TYPED. THIS SHØULD BE CØMPARED TØ PREVIØUS BUILD LIST TØ DETERMINE IF NEW EXTERNALS ARE UNPATCHED.

21. AFTER "Q" IS TYPED RESPØND WITH: *F TØ LØAD THE *FTN TAPE.

Table 14. (continued)

22. WHEN THE TAPE HAS LØADED "ACTIØN" WILL BE REQUESTED.
TYPE: CU - A "Q" SHØULD RETURN. RESPØND BY TYPING: *T
THIS WILL DIRECT THE SI TØ WRITE THE SYSTEM CØRE IMAGE ØNTØ DRUM AND LIST IT'S SECTØR ADDRESS. THE SI SHØULD THEN TYPE "PP", THUS REQUESTING THE PRØGRAM PRØTECT SWITCH BE RAISED. IF THE "PP" MESSAGE DØES NØT ØCCUR THE BUILD SHØULD BE REPEATED.

23. RAISE THE PRØTECT SWITCH AND TYPE: *

24. PRESS MANUAL INTERRUPT TØ GET: MI
EXECUTE THE FIXCHRIS PRØG. WHICH WAS LØADED IN PLACE ØF CHRIS DURING THE SYSTEM BUILD. THIS WILL READ A BINARY CØPY ØF CHRIS FRØM DRUM F AND PATCH IT TØ RUN IN THE LØCATIØN FØR THIS SYSTEM AND WRITE IT ØNTØ DM.
TYPE: S/8077 (IF CHRIS IS DIR. INDEX 18)

25. USE CHRIS TØ CHANGE JØBPRØ+$3ED FRØM $2000,$60 TØ $206C,$9A0

26. USE THE SYSTEM TØ LØAD THE "BULKRY" (LØNG) A-TAPE I.E.:
MI/*P/*K,I2/*1/*L /CU/*2/*Z (THE "/" INDICATES PAUSE).

27. USE CHRIS TØ WRITE THE BULKRY BINARY ØNTØ IT'S DRM LØC.
WMM,5470,64A0,X,Y/ (WHERE X,Y IS MS.LS FØR BULKRY).

28. MØUNT CØMPILER TAPE #1 AND ENTER LIBEDT BY MI/*LIBEDT.
TYPE: *V (NØTE-THE TAPE READER MUST BE ASSIGNED AS INPUT UNIT)
"IN" WILL TYPE AFTER EACH PASS IS LØADED.
WHEN TAPE ENDS "ACTIØN" WILL BE REQUESTED.
MØUNT THE NEXT TAPE AND TYPE: RP
AFTER THE THREE CØMPILER TAPES HAVE LØADED,MØUNT THE "S+ASSEMBLER" TAPE AND TYPE: *V

29. BEFØRE LEAVING LIBEDT ØBTAIN A *DM & A *DL LISTING.
EXIT LIBEDT BY TYPING: *K

30. LØAD & EXECUTE THE PATCH PRØGRAM (*P/CU/*X,N). THIS WILL FIX THE PATCHES ØN THE DRUM. NØTE THE CHANGES BEFØRE MAKING THEM . USING THE LAST SYSTEM PATCH PRINT ØUT SHØULD PRØVE A USEFUL GUIDE.AFTER AUTØ-LØADING THE SYSTEM IS READY.

INTERNAL DISTRIBUTION

1. N. J. Ackermann, Jr.
2. R. K. Adams
3. J. L. Anderson
4. F. T. Binford
5. C. J. Borkowski
6. A. R. Buhl
7-28. J. B. Bullock
29. B. L. Corbett
30. J. A. Cox
31. F. L. Culler
32. S. J. Ditto
33. D. N. Fry
34. W. O. Harms
35. S. I. Kaplan
36. R. C. Kryter
37. R. V. McCord
38. L. C. Oakes
39. J. Pidkowicz
40. D. P. Roux
41. G. S. Sadowski
42. W. H. Sides, Jr.
43. R. S. Stone
44. D. B. Trauger
45. A. M. Weinberg
46. K. W. West
47-48. Central Research Library
49. Document Reference Section
50-53. Laboratory Records Department
54. Laboratory Records, ORNL R. C.
55. ORNL Patent Office
56-63. Technical Information Center (ACRS distribution)
64-68. Division of Technical Information Extension
69. Laboratory and University Division, ORO

EXTERNAL DISTRIBUTION

Argonne National Laboratory, EBR-II Site, Idaho Falls, Idaho 83401
70. R. W. Hyndman

Atomic Energy of Canada, Ltd., Chalk River Nuclear Laboratory, Chalk River, Ontario, Canada
71. Albert Pearson

Atomic International, North American Rockwell, 8900 DeSoto Ave., Canoga Park, Calif. 91304
72. E. B. Ash
73. R. W. Keaten

Babcock and Wilcox, P. O. Box 1260, Lynchburg, Va. 24505
74. H. L. Reeves, Jr.

EXTERNAL DISTRIBUTION (continued)

Commonwealth Edison Company, 72 W. Adams, Chicago, Ill. 60690
75. Dennis R. Swann
76. Larry D. Thompson

General Electric Company-APED, Mail Code 633, 175 Curtner Avenue, San Jose, Calif. 95125
77. L. K. Holland
78. R. L. Johnson

General Electric Company-BRDO, 310 DeGuigne Drive, Sunnyvale, Calif. 94086
79. S. Bailey
80. R. Pflasterer

Gulf General Atomics, P. O. Box 608, San Diego, Calif. 92112
81. S. Nowicki

Hanford Engineering Development Laboratory, Richland, Wash. 99352
82. J. W. Upton, Jr.
83. N. P. Wilburn

Tennessee Valley Authority, New Sprankle Building, Knoxville, Tenn. 37902
84. G. R. Owens

Tennessee Valley Authority, 805 Edney Building, Chattanooga, Tenn. 37401
85. D. C. Keeton

U. S. Atomic Energy Commission, Washington, D. C. 20545
86-90. F. C. Legler
91. J. A. Patterson
92. E. A. Womack

Westinghouse Electric Corp., Atomic Power Division , Waltz Mill Site, P. O. Box 158, Madison, Pa. 15663
93. F. Baloh
94. D. Goeser
95. F. M. Heck
96. L. E. Strawbridge

Wyle Laboratories, 3200 Magruder Boulevard, Hampton, Va. 23366
97. H. P. Danforth