General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-420

DSS Subsystem Implementation by Time-Shared Computer

Robert C. Tausworthe

FACILITY FORM 602
N70-12929
(ACCESSION NUMBER)
88
(PAGES)
CR-107200
(NASA CR OR TMX OR AD NUMBER)
(THRU)
(CODE)
08
(CATEGORY)



JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1969

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-420

DSS Subsystem Implementation by Time-Shared Computer

Robert C. Tausworthe

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1969

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

Preface

The work in this report was performed by the Telecommunications Division of the Jet Propulsion Laboratory.

Contents

I. Introduction and History 1

II. Organization of a Multiprogrammed Subsystem 3

III. Subsystem Development Configuration 5

IV. The Supervisor Program 5

V. Levels of Priority and Organization of Resources 6

VI. Programming Language for Assembly Development 6

VII. A Demonstration System 8

VIII. Conclusion 11

Appendix A. Demonstration Queue Monitor Assembly for SDS-900 Series Computer 15

Appendix B. Modified Universal Binary Loader for Use with QMONITOR 43

Appendix C. QMONITOR, Loader, and User Program Dump Module 73

Appendix D. Sample User Programs 77

Figures

1. Implementation of a complete subsystem by time-shared computer interfaced with hardware 4
2. Organization for development of computer-based subsystem 5
3. Hierarchy and priority of user programs and organization of subsystem monitor 7
4. Time-shared SDS-910 demonstration system configuration diagram 9
5. Time-shared SDS-910 demonstration system module connection and sequencing diagram 10
6. Timing diagram of QSWITC, the time-slot switch of the SDS-910 demonstration queue monitor 11
7. Time-shared queue-switching interrupt subroutine 12
8. Sample of program standard format 13

PRECEDING PAGE BLANK NOT FILMED.

Abstract

A rationale and method are presented for effecting the synthesis of multiple DSIF real-time functions in a single computer. Included are the management-supervisory, as well as engineering disciplinary roles and interfaces required to allow independent, noninterfering subfunction development and subsystem operation.

DSS Subsystem Implementation by Time-Shared Computer

I. Introduction and History

This report was written in consideration of the potential that has been afforded to the DSIF by small general-purpose digital computers and to indicate that this potential can be further increased without a corresponding increase in the number, speed, or memory size of the on-site machines. The principal factor that sets the use of digital stored-program machines at a tracking station apart from the use of similar equipment at a computing facility or a commercial data-management center—whose services and capability are time-varying and available to any and all subscribers—is primarily that the tracking station is configured and committed to provide a given, prescribed function for an extended period of time. Besides this fully dedicated application of an on-site computer, there are other definitive characteristics. First, the programs in these computers tend to make greater demands on the speed and time of the central processor than they do for the input-output (I/O) channels. Computational portions of the program are station-event-actuated and require computation synchronous with the event, whereas I/O can be somewhat asynchronous, as long as it is rapid enough so as not to accumulate too large a backlog. Further, the structure of these programs during their developmental phase is different in some detail from the version operating during a spacecraft mission; each of these is very different from nonreal-time programs of the problem-oriented or data-management types. The set of computer peripherals is extremely different from the usual I/O set found in a job-shop in that, in the station, the computer is a monitoring, processing, and controlling device integrally developed into various of the station subsystems. As such, the set of peripherals not only consists of the usual analog-digital voltage converters, but may also include special subsystem-oriented devices such as number-controlled and programmable oscillators, ranging coders, information-symbol-stream detectors, auto- and cross-correlators, switches, counters, etc. The set of peripherals required during subsystem development is usually different from (and larger than) the set actually needed during operations. Lastly, and very significantly, both the computer software and its associated subsystem peripherals are engineered together as a unit because software-engineering decisions affect peripheral hardware designs, and vice versa. What this implies is that software needs to be engineered, and that this engineering thus needs to fall under the same cognizant management as the hardware part of a subsystem.

From the user point of view, there are several similarities, however. For example, the addition of such capability makes possible certain previously impossible tasks.

It makes some previously difficult jobs relatively easy. Other functions—and this is a major consideration—are reduced in user cost, with increased reliability and confidence inherent in the end product.

The first use of a computer as an integral part of a JPL tracking operation was an experiment in 1962 during which real-time detections of radar echoes from Venus were made. The computer used was a very slow, small, and unreliable device—even by the standards of that day. Speed, memory, and longer-term reliability were available at extra cost, of course, but were not justifiable because of the experimental nature of the application and because, as naive as it was later to become viewed, it did the intended job with such apparent ease that the potentiality of such equipment was intensely obvious. This computer, incidentally, was originally purchased to attempt failure-mode signal processing of a *Ranger* seismometer experiment, but the experiment was subsequently deleted from the *Ranger* missions.

The next application of a stored-program digital device at a station was the Mod III Stored-Program Controller, installed at the Venus site for the 1964 Venus radar experiments as a programmable ranging device. Its sister machine, the Mod II, similar in structure, was a laboratory device that formed the core of the R&D work leading to the (special-purpose) Mark I Ranging Equipment. Both were designed and built at JPL from commercial logic modules at great time and expense to provide methods and technology for increasing the functional capability of a network station by being *part* of the signal-handling, monitoring, and control functions. The Mod III was slow, limited in what it could do, and had only moderate reliability. Since even the R&D site (Venus site) was not, at that time, configured to accept the monitoring it could provide, the control it could effect, nor the signal-processing flexibility it could extend, its potential was never fully realized. However, its presence was certainly noticeable. For the first time, multiple functions were provided for within the program itself. For example, during the 1964 Venus Radar Experiment, the Mod III controlled the ranging codes, keyed the transmitter, timed the round trip, switched in the receiver, followed a range ephemeris of the planet, phase-tracked the echo modulation, put out ranging residuals indexed with time, etc.

The digital processing schemes which had been demonstrated at the R&D site began to find their way into the entire DSIF, and in 1967 had been implemented into the Antenna Pointing System (APS), the Digital Instrumentation System (DIS), and the Telemetry and Command Processor (TCP). Meanwhile, other R&D systems were finding computer implementations: Programmed Local Oscillator (PLO), Mariner 1967 Ranging, and Radar/Radio Astronomy, to name a few. Each application used a separate computer, because there was no convenient way to develop two or more *complete* subsystems into *one* program. However, it was as easy or easier to build a software subsystem than it was to build that same function in hardware, and more importantly, it was cheaper, even with one computer per subsystem. Nevertheless, the jobs given to the single computer did not exhaust its capability.

The DSIF, in consideration of the advancing state of the art, has successfully applied the computer art to a network of tracking stations, and has thus developed a technology based on having one subsystem per computer, with the computer speed and memory size somewhat in excess of that required to do the job. These computers are now nearing the end of their useful cost-economic lives, and will eventually have to be replaced.

Generally speaking, several elements in a subsystem require infrequent and/or minor attention of the computer; others require such attention as would saturate the central processing unit (CPU) of any particular machine. Thus, whereas a computer of a given type could process several tasks of the first type with relative ease, it could not perform one of the latter in addition. Augmentation by certain special-purpose peripherals can often alleviate this CPU strain (usually at the expense of subsystem flexibility, however), so there is a tradeoff between CPU speed/time and peripheral complexity. Of course, each task could have its own computer whose size and speed are commensurate with the task to be performed, which, together with its outboard hardware, is jointly engineered for maximum task cost effectiveness. At the other extreme, all tasks could conceivably be incorporated into a large, very fast machine whose equivalent time-shared speed per program is, on the average, the same as that on each of the smaller machines, and whose memory is at least as large as the sum of all smaller machines. There must be, in this latter case, some means to relax or remove any simultaneous demand for CPU time, either by engineering the occurrences of the actuating event, or by providing peripheral buffers to hold or perform those parts of the tasks that are so time critical.

The possible engineering advantages of the one-machine concept are (1) that the computer, along with

all its peripherals, may be less expensive than the several-machine implementation, (2) that there is only one machine to be familiar with, and (3) that program-to-program interfaces are simple, not requiring hardware. However, it imposes some problems not readily quieted. One has to do with reliability: how does one provide for a failure? On the one-large-computer model one could envision an extra central processor and perhaps a spare bank of memory (but not total replication) to cure the fault, or at least provide fail-safe operation (i.e., a multiprocessor). The fail-safe feature holds automatically for the many-computers model but a cure requires a redundant backup or spare. The difficult problem for the one-computer model arises in specifying what speed, memory, and peripheral I/O devices are needed to do the intended multiple job and how the separate subsystems are to be software-integrated (multiprogrammed) into one computing unit.

It is interesting to note how the commercial job-shop computing facilities handle this latter problem. Either all jobs are entered sequentially in a queue to be processed each in turn, or else several job programs are entered into memory simultaneously and are processed on a time-sharing basis. Programs are continually shifted around as jobs are completed and new ones allocated to memory. Each facility has, as part of its implementation, a resident program called an *operating system*, or *executive program*, or *queue supervisor*, or *monitor*, to give a few of the names given to it, to acquire, allocate, and manage memory, and to load and supervise the execution of programs.

Because of the time dependence inherent in a tracking station activity, the job queue technique cannot be used. The time-share technique has the right elements, but its job-shop implementation is inefficient and not oriented toward the type of function the computer is to provide in a tracking station. On purely a cost-effective basis, it is not yet known whether it is more economical to group many or all present computer-based tasks into one large, fast machine, or to retain the one-job/one-computer relationship upon which the technology evolved, but with each computer smaller, inexpensive, and sized to the task. The one-computer-per-task engineering technique is now a matter of record. To be able to evaluate the one-computer-per-subsystem model, there must first be a model.

The ensuing sections consider the aspects of multiprogramming a computer for implementing specific functions into a tracking station. How a subsystem is managed, programmed, and operated is discussed; also considered is how new systems, as they evolve, are implemented at minimum interference and cost to the total station.

II. Organization of a Multiprogrammed Subsystem

The specific problem to which this report is addressed is that of providing a means and rationale for implementing any of the several computer-based DSS equipments by a single computer interfaced directly with the analog and special-purpose digital equipments in a tracking station. This is the logical next step from the one-computer-per-function technology already developed within the DSIF, and one which will lead to a next logical step of one computer per site, should the latter prove technically feasible and economically advantageous.

A word about nomenclature. In the DSN, the word *system* usually refers to the total configuration of subsystems that operate together to perform a function that cannot be accomplished by a single subsystem. A *subsystem* is an integrated set of hardware and/or software grouped together for development that performs a closely related group of functions and uses a common technology. For example, within the DSIF, there are RF Doppler and Signal Level, Ranging, Angle Tracking, Telemetry, Command, and Monitor/Control Systems. These systems are implemented by the conjunctive participation of several subsystems, such as the Antenna Mechanical, Microwave, Receiver/Exciter, Transmitter, Tracking, Telemetry/Command, Monitor/Control, Frequency/Timing, Recording, and Technical Facilities Subsystems. Each of the subsystems, too, provides a set of functions, and is thus divided naturally into a multiple set of sub-subsystems called *assemblies*. Functionally, as a typical example, the DSIF Tracking Subsystem measures Doppler and range, points the antenna, programs the local oscillator or exciter, converts data for communications lines, records tracking data, and compares data with predicts. The assemblies for providing these are the Tracking Processor, Doppler, Ranging, Programmed Oscillator, Communications Buffer, and Digital Recorder Assemblies.

The approach taken here is that each assembly is developed on an individual basis, separate insofar as installation, interface, test, modification, and operation are concerned, but unified in that they time-share a single central processor. This approach stresses and preserves a proven technology: The functional individuality of each assembly. To both the development engineer as well as

the operator, the assembly appears as if it were made with one computer per assembly.

Each assembly consists of an assigned fraction of the computer capability, along with certain interfaces with the computer, other subsystems, and other assemblies (Fig. 1). One such computer interface can be a special purpose Operations and Monitor terminals for that assembly. Such a terminal has display and controls as required; these may include I/O devices, indicator lights, test points, assembly hardware/software checkout facilities, test inputs, mode switches, parameter settings, etc.

The development of any particular subsystem is under the management and technical direction of a subsystem Cognizant Development Engineer (CDE). As indicated in Fig. 2, the several assemblies are developed by individual Assembly Cognizant Engineers (the initials ACE here seem singularly appropriate), who must not only build within the technical confines of the function and equipment involved, but must also adhere to various codes and standards imposed by system and subsystem management.

The organizational philosophy may be summarized as follows:

(1) The computer has sufficient speed, memory, and peripherals to perform all subsystem functions.

(2) Each assembly is allotted a certain fraction of the computer capability (speed, time, memory) by the subsystem CDE, commensurate with the function to be performed.

(3) Each separate assembly is developed as a separate entity by cognizant engineers with other assemblies and subsystems as interfaces.

(4) Each assembly is developed noninterferingly and not necessarily synchronously (except for program interfacing and schedule milestones) with other assemblies.

(5) Interface conflicts are resolved by subsystem CDE.

(6) Each assembly is viewed as a set of modules, connected together.

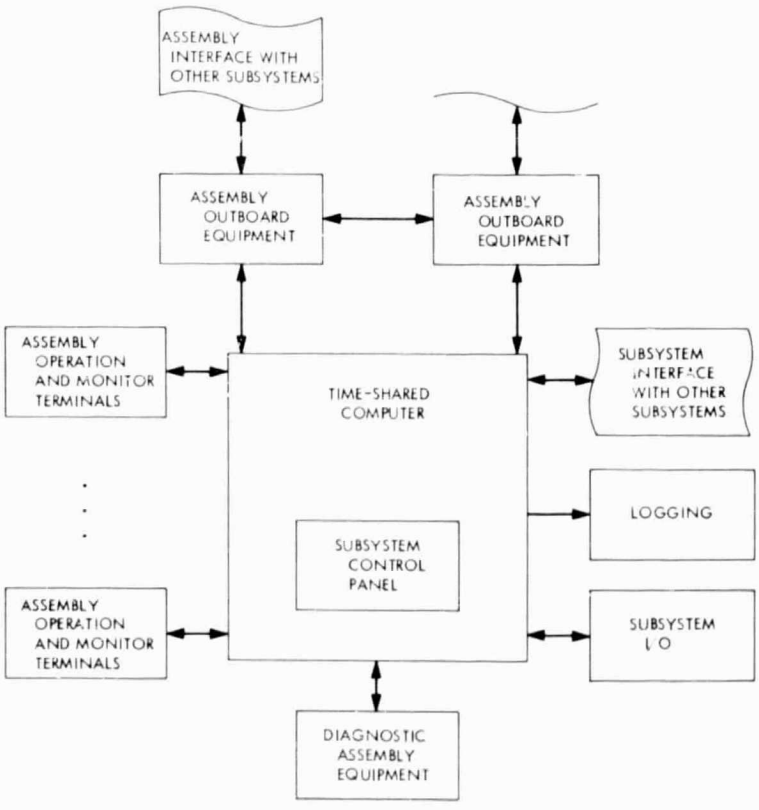


Fig. 1. Implementation of a complete subsystem by time-shared computer interfaced with hardware

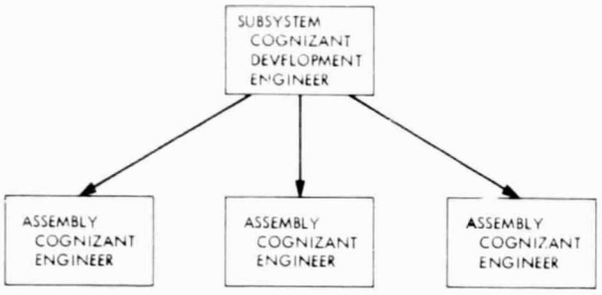


Fig. 2. Organization for development of computer-based subsystem

III. Subsystem Development Configuration

The special nature of the task is one that necessitates a fully dedicated installation of the computer; by this is meant that the total configuration and capability is engineered for the job at hand (allowing for reasonable growth potential of the subsystem), and that the functions performed are fixed for a given mission, extending essentially intact over an entire track. Some of the desirable characteristics are ones that extrapolate naturally from present technology; for example, The subsystem structure makes use of software as well as hardware management by the CDE. Development adheres to a set of module/assembly/subsystem/system standards, including programming standards for language, format, technique, and documentation, as well as interface standards for both software and hardware. Interface conflicts are resolved by the CDE.

Such characteristics are also constraints which imply a simpler, more efficient computer implementation in addition to the positive supervisory control of the way functions are performed.

The computer software configuration consists of a supervisory system (Queue Monitor) and the separate assembly program modules. The supervisory system software can be viewed as a separate assembly, and, as such, the prototype is developed within the DSIF. Because it is a prototype, and because it is written by systems engineers, rather than by programming experts, it may be somewhat inefficient in its use of the computer and insufficiently documented. However, it does have the required characteristic: it is an operating piece of software, expertly designed *in function*, specifically for a tracking station. The operating field version of the supervisor is a more efficient and fully documented version of the prototype possibly contracted outside the DSIF to "superprogrammers" whose expertise is in software rather than communications systems. Efficiency in the supervisor is necessary, as it will be used in each computer-based subsystem.

The software for each assembly, as well as its hardware, is developed by the ACE; it is not necessary to make it more efficient by superprogramming as long as it fits within the assigned computer fraction. It is thus the CDE's responsibility to ensure that the computing allotment is properly assigned, that there is consistency between the efficiency of the program and the allowed resource. Each ACE maintains and documents his own software, kept separate, and not released until the assembly is operative. Once all assemblies are operative, i.e., when the subsystem is fully developed, the entire computer software is dumped as a unit, approved, and certified by the CDE for use in a tracking operation.

IV. The Supervisor Program

Because of the dedicated nature of the installation, and because of some of the system management is allocated to the CDE, the Queue Monitor can be a relatively simple, low-overhead assembly, both in the amount of memory it occupies and in the time it takes to supervise the execution of assembly programs. Besides containing the time-share queue switch, it is also proper to include as part of the supervisory assembly those subroutines (such as I/O, programmed operators, etc.) that are common to many users, written in re-entrant code (or perhaps usable only when not already in use).

The Queue Monitor need not allocate CPU time to programs in a wait mode. This is not necessarily a bonus feature in the initial assignment of the fractional capability given to an assembly, but may be of advantage to a CDE trying to fit newly developed assemblies in with ones not completely utilizing their allotted fraction, by post-evaluation of assembly requirement.

During subsystem development the supervisor should be made to allow each ACE extreme flexibility in the development of his assembly. For example, it can allow for the loading, debugging, and testing of one assembly while others are either running or being similarly developed. This not only gives the subsystem the same structure it will have during operations, but makes it possible to develop each assembly separately, simultaneously, and noninterferingly with development, test, or operation of other assemblies.

However, it is not necessary for the program dump, i.e., the station version of the subsystem, to contain the full flexibility of the development subsystem supervisor. For example, it need contain only that part of the supervisor required to monitor the assemblies; that part dealing with loading and manipulating separate programs on an individual basis is not needed during operation.

V. Levels of Priority and Organization of Resources

One characteristic of real-time programs is that they are required to perform designated actions upon receipt of external influences. Some of these are demands for immediate response; others initiate actions which, subsequent to their initiation, are not as extremely time critical, but nevertheless, have a certain urgency; still others, such as those monitoring an equipment status, require no particular synchronization with that external equipment at all.

Many computers, and the ones we shall assume here in use, have an interface system of external inputs for interrupting the normal c .nputer CPU sequence and branching to a specified location for its next instruction. Such inputs are arranged in priorities, in which certain interrupts are allowed even when the computer is still processing a previous, but lower interrupt call; for others, the interrupt request is put into a priority queue to await its turn.

In accordance with such a priority interrupt scheme, the subsystem priorities can be divided into four categories:

Level IV. Fixed time-slot allocation of CPU time to assemblies in cyclic repetition. Number of slots per cycle for each assembly approved by CDE.

Level III. Interrupt-actuated queue (IRA): interrupt event initiates reassignment of assembly time-slot to one of the assembly subprograms. Upon subprogram exit, the assembly reverts to level IV.

Level II. System Interrupt: interrupt event common to many users initiates short-user programs. Length and execution order must be approved by CDE.

Level I. Private Interrupt: interrupt event initiates action for one assembly only. Interrupt linkage assigned by CDE and program length approved by CDE.

As will be noted, each user assembly has a certain time per subsystem cycle assigned to it, of which no other assembly can deprive it except when interrupted by level I and II priorities, and thus approved by the CDE. Should the ACE desire to make use of a time-event-actuated subprogram, he can do so at his own volition and cognizance as long as he stays within his time slot allocation. This he does merely by requesting the Queue Monitor to reassign his time slot from level IV to the level III subprogram or he can request a flag be set (by a level II event) which can be sensed in the assembly level IV software. For more critical applications, the ACE may request both extra time and priority through the CDE, to ensure that no infringement with other assemblies occur. The four priority levels are illustrated in Fig. 3.

VI. Programming Language for Assembly Development

The procedure for turning an ACE's requirements into an assembly program is the transformation of a sequence of instructions, written in the problem-oriented, or *source* language, into machine instructions coded and stored in the computer. Commonly this process is accomplished in two steps: first, the source language is scanned and translated into an intermediate language isomorphic to machine code; second, a loader interprets and inserts the program into memory and transfers control to a given location.

The compiler for a subsystem is a part of the development effort only; it is not necessary in the tracking station version. Because of this, the compiler may be a separate assembly in the development subsystem, or it may be a free-standing program in a computer other than the target machine. In the latter case, it is necessary to have the intermediate output discussed above, with only the loader as part of the development subsystem.

The terminal characteristics, in either case, regardless of the configuration or complexity of the compiler are the important factors. The source language should be simple, flexible, and properly oriented to the subsystem; the machine code should be efficient. In fact, it does not seem unreasonable, considering the costs involved, to sacrifice convenience in source language at the gain of machine code efficiency. For example, a subset of Fortran modified to exclude high-overhead routines, interspersed with symbolic machine instruction capability, would provide a convenient beginning source language.

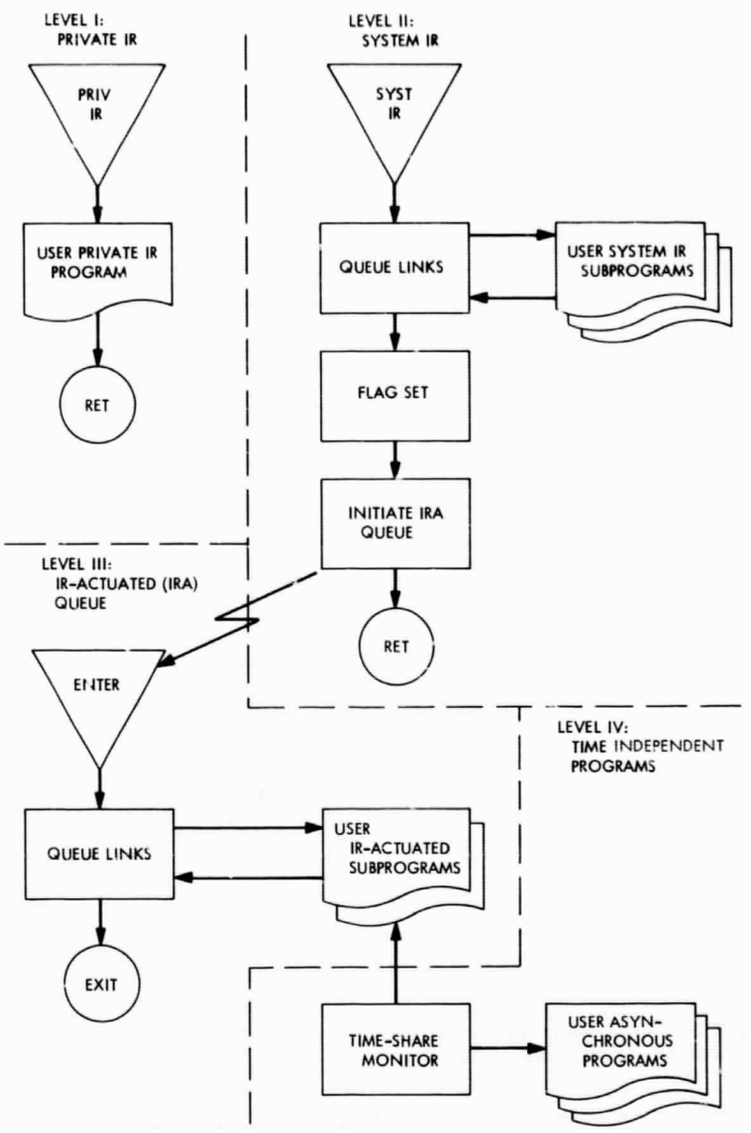


Fig. 3. Hierarchy and priority of user programs and organization of subsystem monitor

The compiler would naturally exclude the use of certain instructions which, if used improperly in an assembly, could result in the failure of another assembly. Such is only a benefit in the development subsystem since assemblies must be certified by the CDE prior to inclusion in the subsystem dump.

As the compiler program itself, once the source language is defined, is devoid of station-oriented constraints, it should properly be written by superprogrammers rather than by the DSIF. It does fall within the responsibility of the DSIF, however, to define the source language, perhaps in consultation with programmers, as regards to a proper balance between source language convenience and output code efficiency.

Assembly documentation is required on several levels. The first level is a block diagram showing the organization of the assembly purely on a functional basis. To turn the functional requirement into an operating unit, however, there has to be a technique analogous to that used to turn hardware functions into hardware. For software, this technique is a hierarchy of documentation, the first of which is the functional block diagram.

The second level of drawings consists of a sequencing chart for each mode. A sequencing chart has somewhat the same form as a conventional flow chart, but it is concerned with how the several functions required in a given mode are called up when required. The functions are shown only as blocks, and the chart shows the queueing scheme, the links to other modes, and the operation of interrupts. In cases where timing is critical, the chart is accompanied by a timing diagram.

The next level of drawings consists of a detailed flow chart for each of the functions to be performed in each mode. The machine program is derived from these flow charts and represents the final level of documentation by the ACE.

The DSIF has a standard procedure[1] for the preparation of DSIF computer programs that can be expanded to contain the same detail the DSIF requires on all hardware. As a minimum, subsystem documentation found in a station should include:

(1) Functional block diagram.

(2) Sequence block diagram of assembly programs.

(3) Timing chart of program actuations and CPU usage.

(4) Schematic of each hardware module.

(5) Flow chart of each program module.

(6) Source-language listing of each module showing resulting machine code, relative memory assignments, and characteristics of all interfaces between modules.

(7) Annotated memory map of each assembly program.

Each level of documentation adheres to a set of standards, some of which are set by the DSIF, others imposed by the CDE on the ACE, and by the ACE on his development personnel.

VII. A Demonstration System

To illustrate the desired characteristics, to demonstrate a specific documentation technique, and to aid in formulating some of the foregoing rationale, a simple supervisory assembly has been programmed for an SDS-910 computer. The reason for choosing the SDS-910 was that it was available for the purpose, and that any program which would work on it would also operate on an SDS-920 or SDS-930 as well, without alteration (albeit some alterations, making use of the more powerful instruction repertoire, could markedly increase its efficiency).

A simple functional block diagram is shown in Fig. 4, as the demonstration configuration now stands. It may be noted that the handling of level III IRA queues is not implemented in this version, but otherwise, the system performs all the functions outlined in the preceding sections of this report.

The indicated block diagram represents the first level of subsystem documentation. It shows the connectivity of the various computer interfaces and I/O devices with the internal functions of the computer. It shows, for example, that all buffer I/O, whether initiated by a user program or the loader, take place under the supervision of the time-share monitor. It shows that the loader, when initiated by the fill push-button, is executed in the time-shared queue mode. Parallel I/O, however, is programmable at user discretion.

From such a diagram, some of the things a CDE must supervise is readily apparent, such as the parallel I/O interface, the private IR links, the length of private IR programs, and the interchange between private IR programs and non-IR programs.

[1]DSIF Standard Procedure 10-18c.

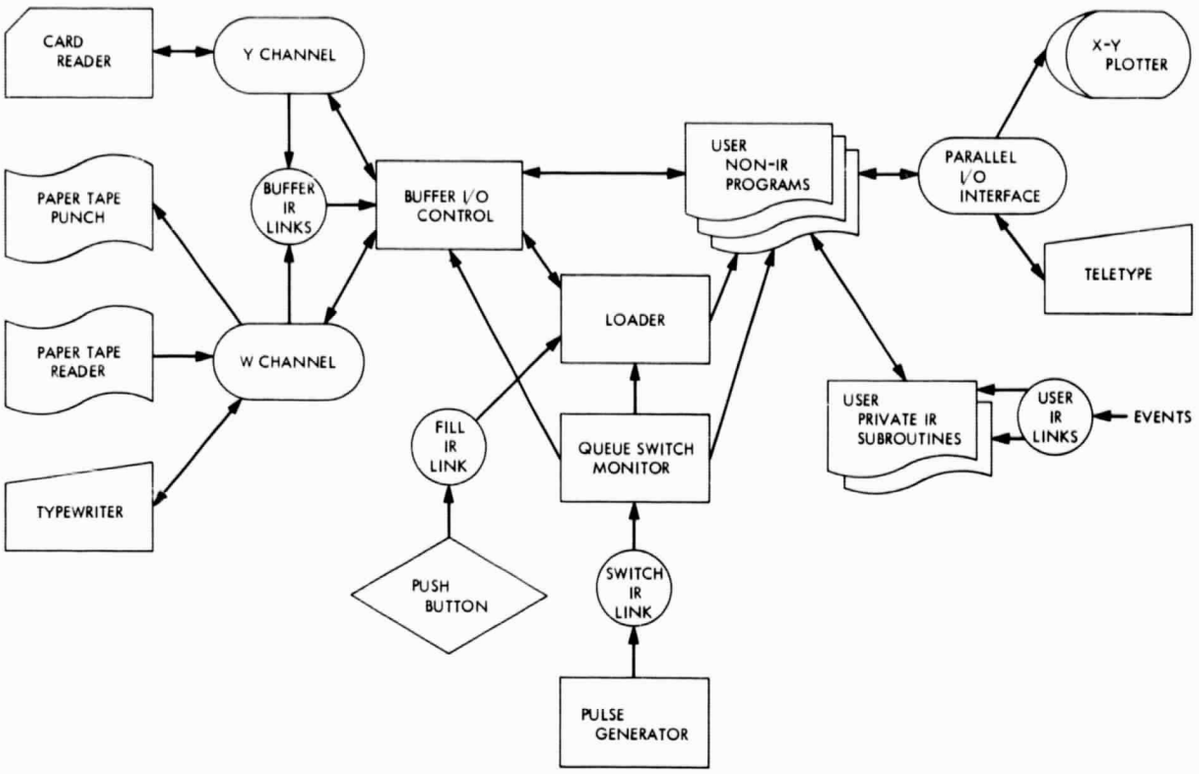


Fig. 4. Time-shared SDS-910 demonstration system configuration diagram

The next level of documentation, Fig. 5, is a sequencing chart showing the execution order of the assembly and the way modules are connected together. Each module is designated by its name (which is also its source language entry call). The diagram is annotated, showing how each subprogram is initiated, which subroutines it calls (designated by double arrows), and how exit is made. In the event there are multiple exits, as in QUBLDR for example, an assertion about the specific condition for exit is required.

In instances where timing of the various routines is critical or otherwise important, a timing diagram, such as Fig. 6, is appropriate. Figure 6 shows how the queue-switch interrupts are related to a time slot, how the QSWITC subroutine reacts to the interrupt, and how the user programs are sequenced.

The next level of diagram is the module flow chart, as shown for QSWITC in Fig. 7. This diagram uses standard flow-chart symbols for functions, decisions, etc. One item of note in the diagram is that each entry in a box has a reference number affixed to it. This is to identify that point in the final documentation level, the compiler printout, as illustrated in Fig. 8. Because of its availability, the source language was taken to be SDS SYMBOL.

The SYMBOL compiler not only lists the source statements, but also the machine code on the same printout. The compiler output paper tape format is accepted by the SDS Universal Binary Loader, which stores the program into memory. The SDS loader (QUBLDR), modified to operate its I/O under interrupt control by an evanescent subprogram PATCH forms the queue monitor loader. Because it was adapted from an existing program and because of the structure of the SDS-910 input media, no more than one user can use the loader at one time (i.e., it is nonreentrant), and only via the paper tape reader on the W buffer. In the event the W buffer is in use by another program, its use is time-shared between loading and the other I/O on a record-by-record basis.

Because SYMBOL (or METASYMBOL) only is available as a source language there are few natural restrictions imposed on user programs, other than those imposed by the programmer himself. Since the interrupt feature is enabled, and since the W and Y buffers have EOT and EOW interrupts, all buffer I/O must be under IR control.

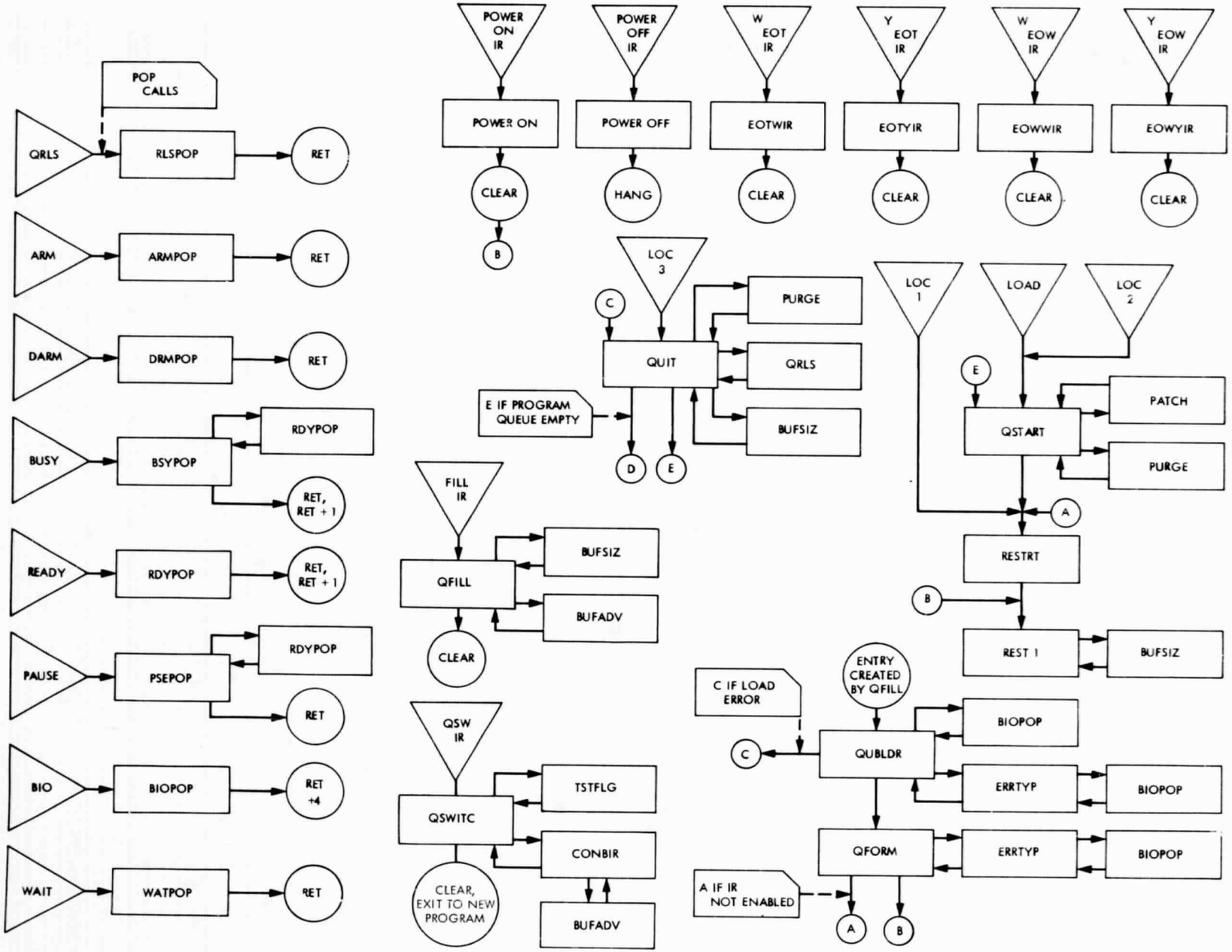


Fig. 5. Time-shared SDS-910 demonstration system module connection and sequencing diagram

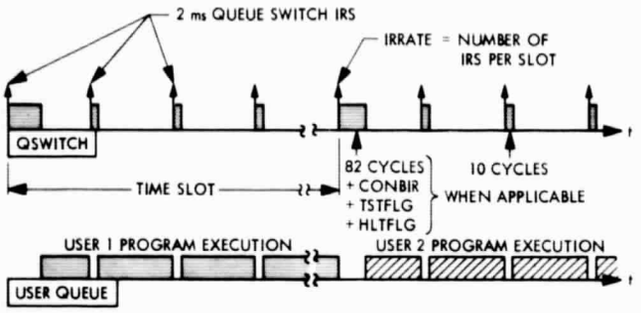


Fig. 6. Timing diagram of QSWITC, the time-slot switch of the SDS-910 demonstration queue monitor

Hence the user is provided a buffer I/O programmed operator, BIO; and to provide users with specific IR arming/disarming without negotiation with or interference to other user programs, ARM and DARM operators are provided.

There are other programmed operators provided but these are with regard to not wasting time in various wait modes, such as halts (WAIT), breakpoint waits (PAUSE), ready tests (READY), or busy tests (BUSY). Other programming restrictions are listed on the QUEUE MONITOR nameplate, Fig. 8, and the entire demonstration system compilation appears in Appendixes A, B, C, and D. The incomplete documentation is a natural consequence of the development status of the program. The entire QUEUE MONITOR requires 735 memory locations exclusive of the loader which occupies 650 locations.

Several user programs have been written to exercise the QUEUE MONITOR; all but one are I/O bound; that is they are usually waiting for a character to be finished typing, etc. During this wait, the CPU is free for the other, a compute-bound program which simply increments a cell and displays the number asynchronously as a plotted line. With separate programs to read cards, type card images, type on the teletype, test a breakpoint then type, and accumulate and plot, it was found that each I/O-bound program consumed under 5% of the CPU time and that the QSWITC overhead was about 5%.

VIII. Conclusion

This report has given a method, rationale, and specific demonstration for the implementation of a time-shared capability within the present DSIF which can readily be extended to the DSN Mark III configuration. The technique is one capable of utilizing at least 95% of a machine's capability (SDS-910), and greater utilization of a machine with a more flexible machine code repertoire or increased speed.

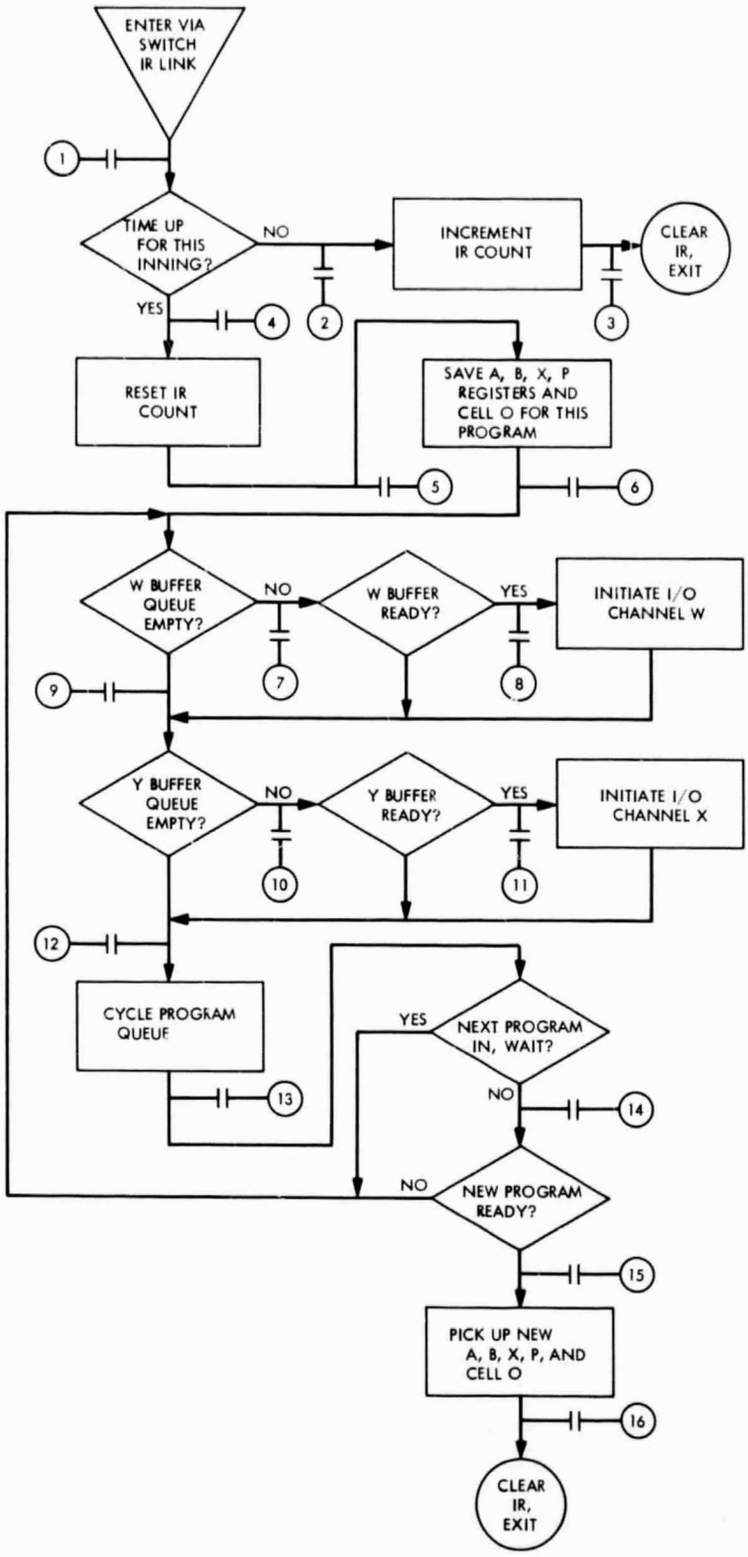


Fig. 7. Time-shared queue-switching interrupt subroutine

```
                        179          PAGE
                        180  ********************************************************************
                        181  *                                                                  *
                        182  *       QUEUE SWITCHING INTERRUPT MODULE                  (QSWITC)*
                        183  *                                                                  *
                        184  ********************************************************************
                        185  *                                                                  *
                        186  *       FUNCTION: TO GIVE TIME-SLOT TO NEXT PROGRAM IN QUEUE LIST. *
                        187  *                                                                  *
                        188  ********************************************************************
                        189  *                                                                  *
                        190  *       SYMBOL TABLE DEFINITIONS:                                  *
                        191  *                                                                  *
                        192  *              IRCNT           NUMBER OF IR'S LEFT THIS SLOT, NEG I*
                        193  *              XQSW            X REG TEMP SAVE                     *
                        194  *              IRMAX           NUMBER OF IR'S PER TIME-SLOT, NEG I *
                        195  *              QNOW            PROGRAM NOW IN CONTROL, NEG I       *
                        196  *              QA              \                                   *
                        197  *              QB               \                                  *
                        198  *              QX                 ARRAYS FOR SAVING A,B,X,P,CELL 0 *
                        199  *              QP               /                                  *
                        200  *              QC              /                                   *
                        201  *              QTOPAD          CELL WITH LOC OF TOP OF W BUFF QUEUE*
                        202  *              P1              =+1, I                              *
                        203  *              QITEM           POINTER TO QUEUED PROG LIST, NEG I  *
                        204  *              QNUM            NUMBER OF ITEMS IN PROG LIST, NEG I *
                        205  *              QO              =+0 , I                             *
                        206  *              QLIST           ARRAY WITH QUEUE OF PROG NUMBERS    *
                        207  *              HLTFLG          ARRAY WITH PROGRAM HALTS; GO IF NEG.*
                        208  *              TSTFLG          ARRAY WITH PROGRAM TEST INSTR:      *
                        209  *                               GO IF 0; ELSE EXU TSTFLG; GO IF SKP*
                        210  *              N1              =-1, I                              *
                        211  *                                                                  *
                        212  ********************************************************************
                        213  *       ENTRY:  IR TO CELL DESIGNATED DURING COMPILE, BRM QSWIIC    *
                        214  *       EXIT:   CONTINUES AT PREVIOUS INTERRUPTION OF PROGRAM IN QUEUE *
                        215  ********************************************************************
                        216  *                                                                  *
                        217  *       SUBROUTINES:                                               *
                        218  *              BRM CONBIR      TO CONNECT BUFFER IR'S FOR NEXT I/O;*
                        219  *                               RET $+1                            *
                        220  *                                                                  *
                        221  ********************************************************************
     00000252           222  QSWO    EQU     $               ENTRY
00252  0 00 00000       223  QSWITC  PZE                     CALL SAVE CELL
00253  0 53 00042       224  QSW1    SKN     IRCNT           END OF INNING
00254  0 01 00257       225          BRU     $+3              YES.
00255  0 61 00042       226  QSW2    MIN     IRCNT            NO. INCREMENT IR COUNT
00256  0 01 40252       227  QSW3    BRU     *QSWITC              RETURN TO PRESENT PROGRAM
00257  0 37 00027       228  QSW4    STX     XQSW            SAVE X REG
```

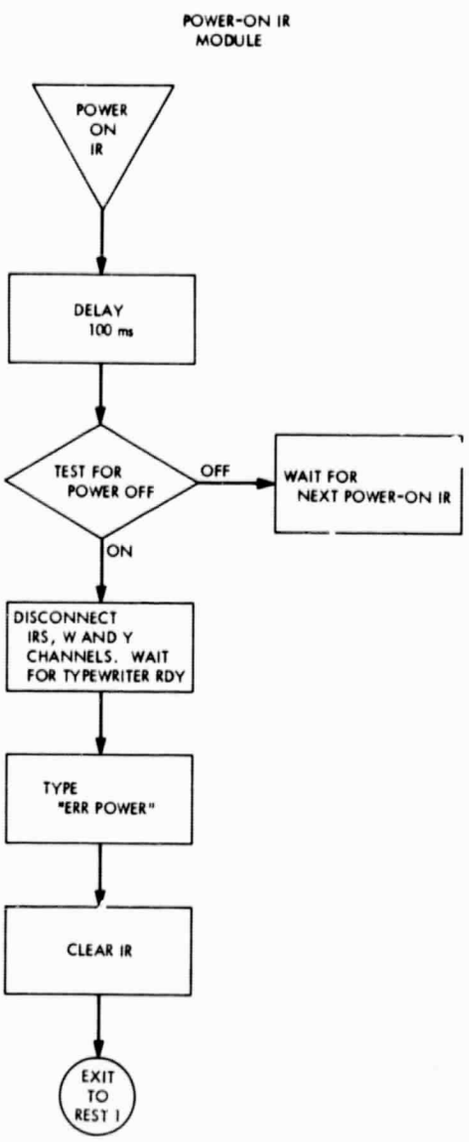
Fig. 8. Sample of program standard format

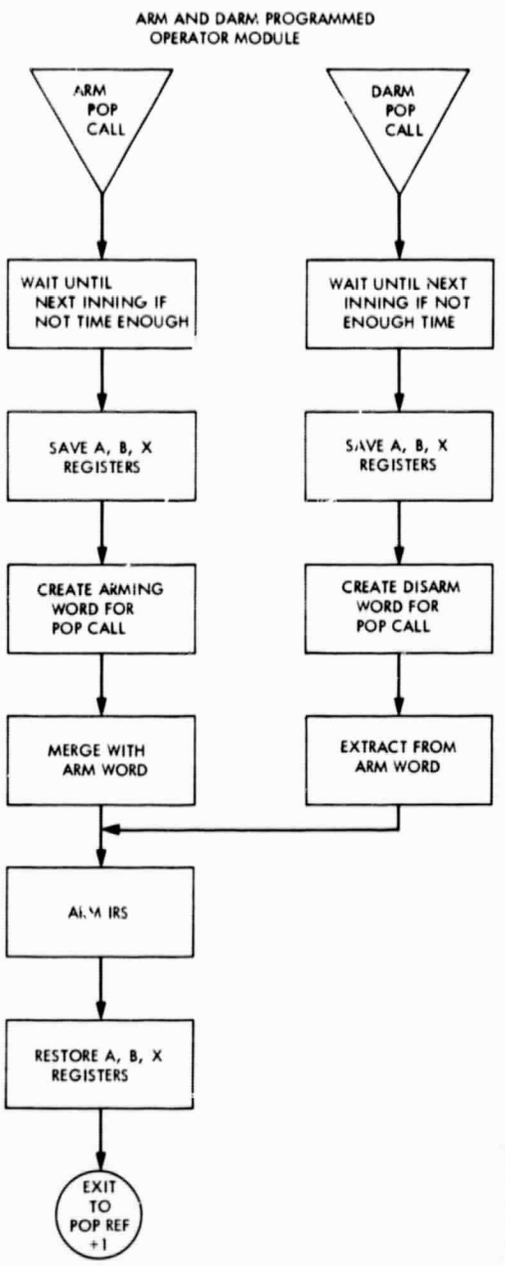
```
00260  0 71 01061      229          LDX     IRMAX           RESTORE IRCNT
00261  0 37 00042      230          STX     IRCNT            TO (NEG) MAX VALUE
00262  0 71 00020      231  QSW5    LDX     QNOW            PICK UP NUMBER THIS PROG
00263  2 35 01172      232          STA     QA,2            \
00264  2 36 01205      233          STB     QB,2             \
00265  0 76 00027      234          LDA     XQSW              \
00266  2 35 01322      235          STA     QX,2               \ SAVE A,B,X,P REGS AND CELL 0
00267  0 76 00252      236          LDA     QSWITC             / FOR THIS PROG FOR NEXT ENTRY
00270  2 35 01274      237          STA     QP,2              /
00271  0 76 00000      238          LDA     0                /
00272  2 35 01157      239          STA     QC,2            /
00273  0 53 40040      240  QSW6    SKN     *QTOPAD         W BUFF QUEUE EMPTY
00274  0 01 00301      241          BRU     $+5              YES.
00275  0 40 21000      242  QSW7    BRTW                     NO. IS W BUFF BUSY
00276  0 01 00301      243          BRU     $+3              YES.
00277  0 71 01074      244  QSW8    LDX     P0               NO. PICK UP W BUFFER DESIGNATION
00300  0 43 00550      245          BRM     CONBIR             CONNECT W BUFF IR'S, RETURN
00301  0 53 40041      246  QSW9    SKN     *QTOPAD+1       Y BUFF QUEUE EMPTY
00302  0 01 00307      247          BRU     $+5              YES.
00303  0 40 22000      248  QSW10   BRTY                     NO. IS Y BUFF BUSY
00304  0 01 00307      249          BRU     $+3              YES.
00305  0 71 01075      250  QSW11   LDX     P1               NO. PICK UP Y BUFF DESIGNATION
00306  0 43 00550      251          BRM     CONBIR             CONNECT Y BUFFER IR'S, RETURN
00307  0 71 00017      252  QSW12   LDX     QITEM           PICK UP QUEUED LIST POINTER
00310  0 41 00312      253          BRX     $+2             INCREMENT LIST. END OF LIST
00311  0 71 00021      254          LDX     QNUM             YES. RECYCLE TO BEGINNING OF LIST.
00312  0 37 00017      255          STX     QITEM            NO. STORE POINTER
00313  2 71 01261      256          LDX     QLIST,2         PICK UP NEXT PROG NUMBER.
00314  0 37 00020      257          STX     QNOW            STORE PROG NUMBER
00315  2 53 01144      258  QSW13   SKN     HLTFLG,2        THIS PROG IN WAIT
00316  0 01 00273      259          BRU     QSW6             YES, SO SKIP IT.
00317  2 76 01335      260  QSW14   LDA     TSTFLG,2         NO. PICK UP PROG TEST REQUEST.
00320  0 72 01067      261          SKA     N1              ACTION REQUIRED
00321  0 01 00333      262          BRU     QSW17            YES
00322  0 71 00020      263  QSW15   LDX     QNOW            \
00323  2 76 01274      264          LDA     QP,2             \
00324  0 35 00252      265          STA     QSWITC             \
00325  2 76 01157      266          LDA     QC,2                \ PICK UP A,B,X,P REGS AND CELL 0
00326  0 35 00000      267          STA     0                   / FOR NEW PROGRAM
00327  2 76 01172      268          LDA     QA,2               /
00330  2 75 01205      269          LDB     QB,2              /
00331  2 71 01322      270          LDX     QX,2             /
00332  0 01 40252      271  QSW16   BRU     *QSWITC         EXIT TO NEW PROGRAM.
00333  2 23 01335      272  QSW17   EXU     TSTFLG,2        PERFORM REQUESTED ACTION. PROG READY
00334  0 01 00273      273          BRU     QSW6             NO, SO SKIP IT
00335  0 46 30003      274          CLR                      YES,
00336  2 35 01335      275          STA     TSTFLG,2          CLEAR TEST REQUEST,
00337  0 01 00322      276          BRU     QSW15              GO ON
                        277  ********************************************************************************
```

Fig. 8 (contd)

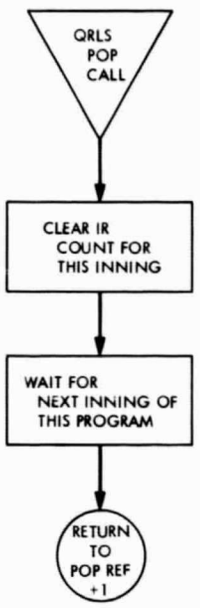
Appendix A

Demonstration Queue Monitor Assembly for SDS-900 Series Computer

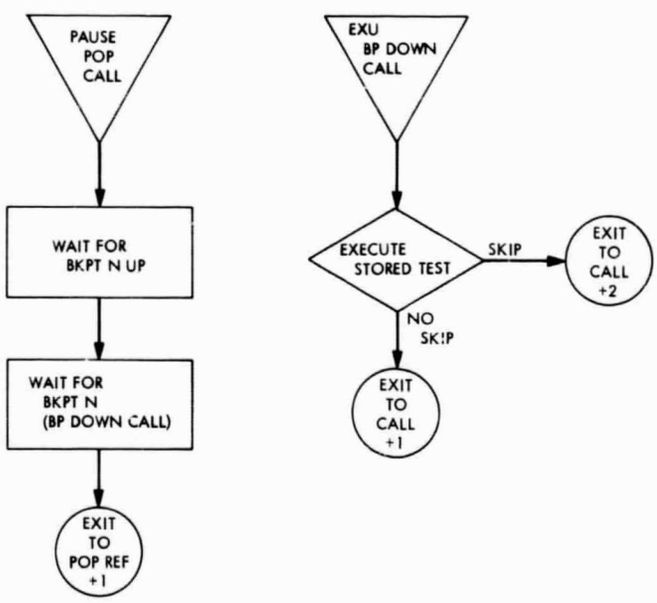

POWER-ON IR MODULE
POWER ON IR
DELAY 100 ms
TEST FOR POWER OFF
OFF
WAIT FOR NEXT POWER-ON IR
ON
DISCONNECT IRS, W AND Y CHANNELS. WAIT FOR TYPEWRITER RDY
TYPE "ERR POWER"
CLEAR IR
EXIT TO REST 1

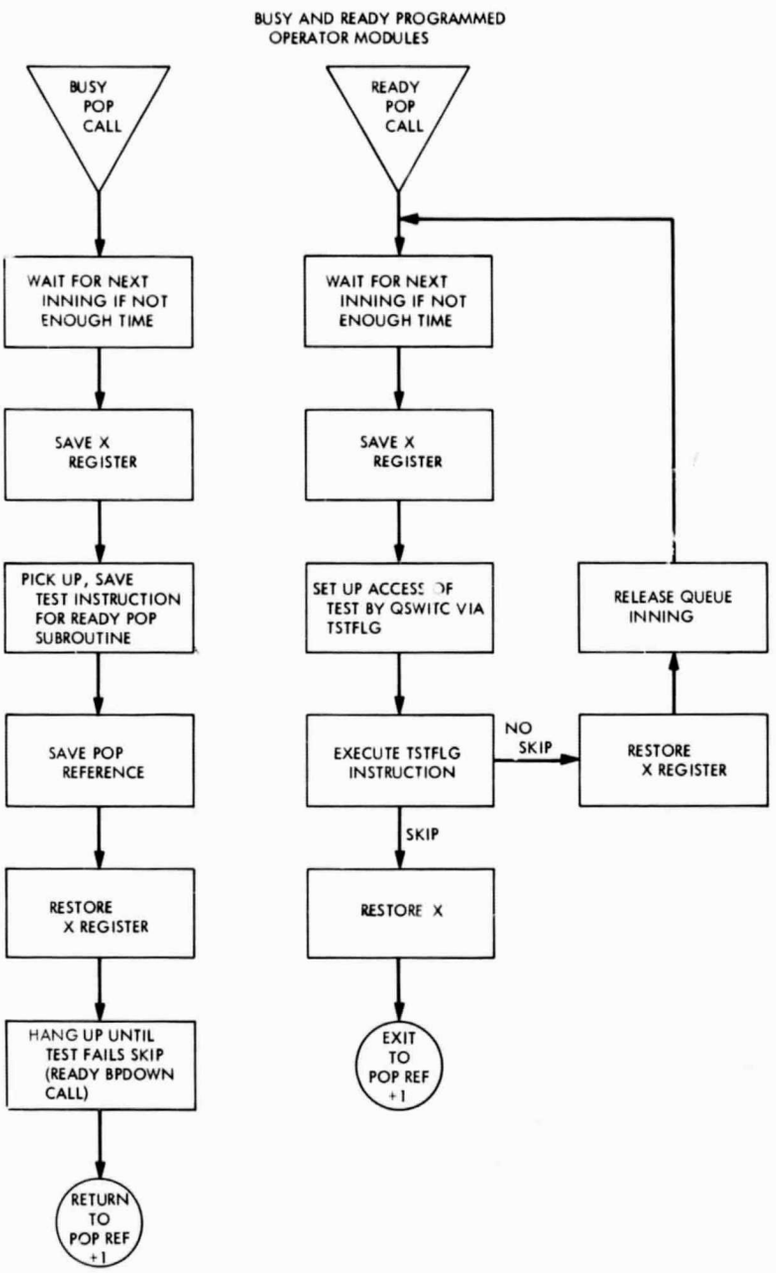

ARM AND DARM PROGRAMMED OPERATOR MODULE
ARM POP CALL
DARM POP CALL
WAIT UNTIL NEXT INNING IF NOT TIME ENOUGH
WAIT UNTIL NEXT INNING IF NOT ENOUGH TIME
SAVE A, B, X REGISTERS
SAVE A, B, X REGISTERS
CREATE ARMING WORD FOR POP CALL
CREATE DISARM WORD FOR POP CALL
MERGE WITH ARM WORD
EXTRACT FROM ARM WORD
ALM IRS
RESTORE A, B, X REGISTERS
EXIT TO POP REF +1

QUEUE RELEASE PROGRAMMED
OPERATOR MODULE

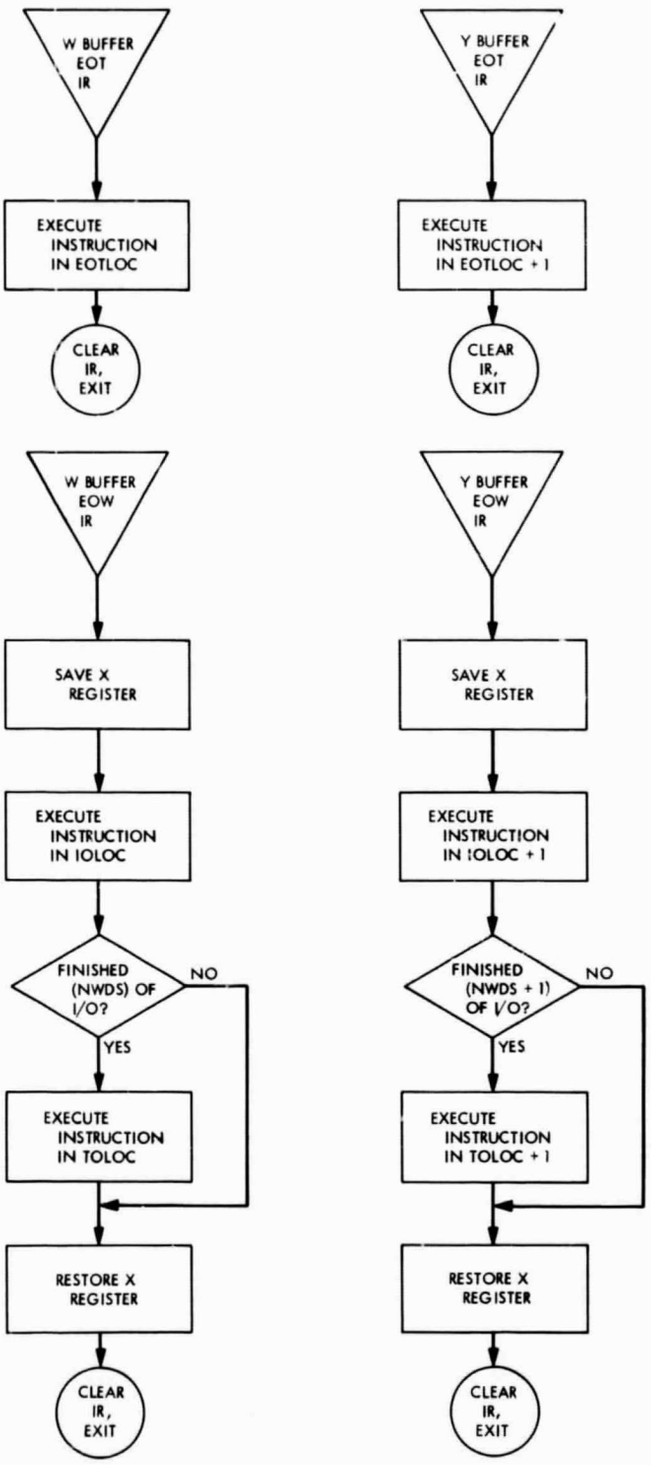


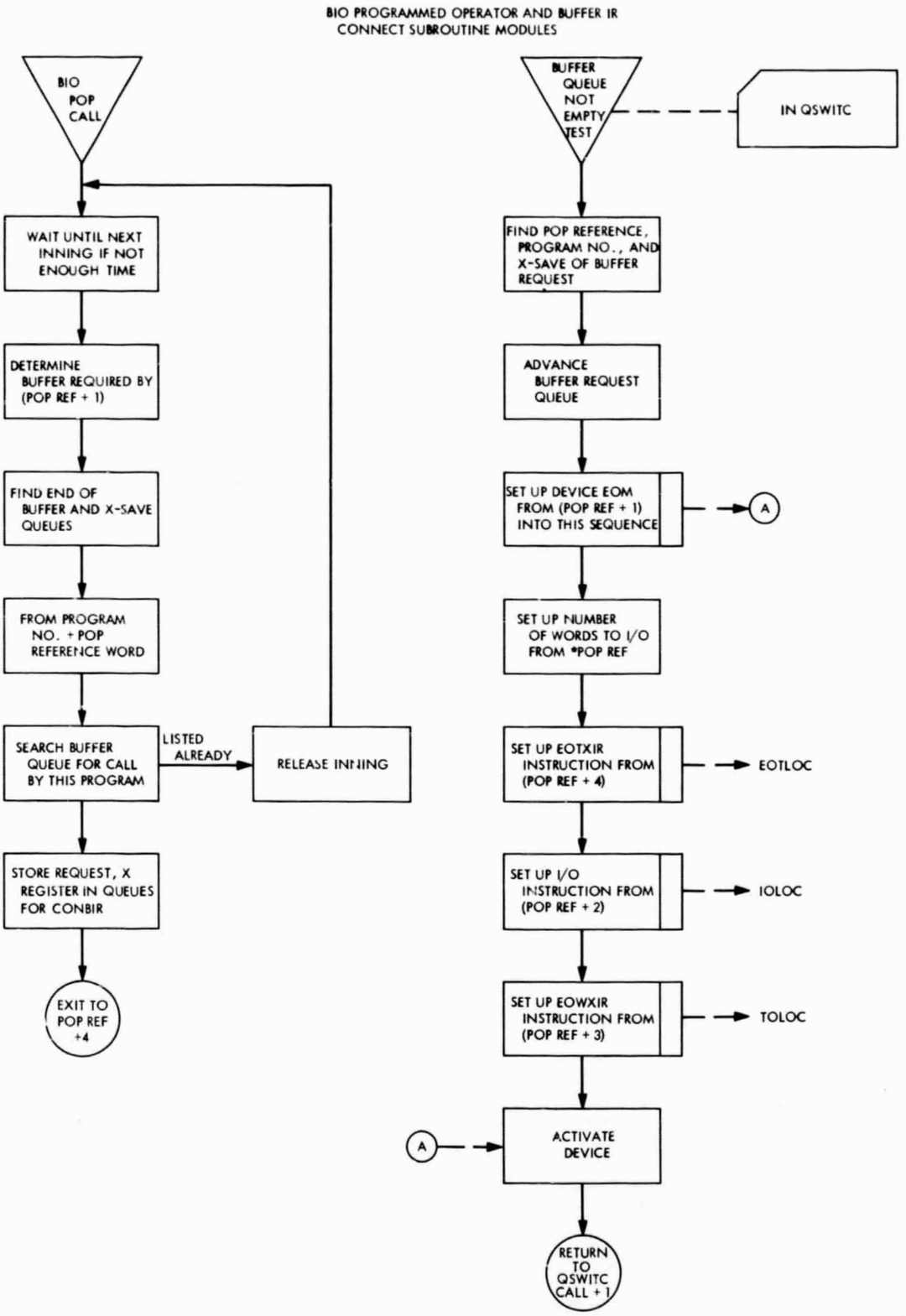
PAUSE PROGRAMMED
OPERATOR AND TEST
INVERTER MODULE



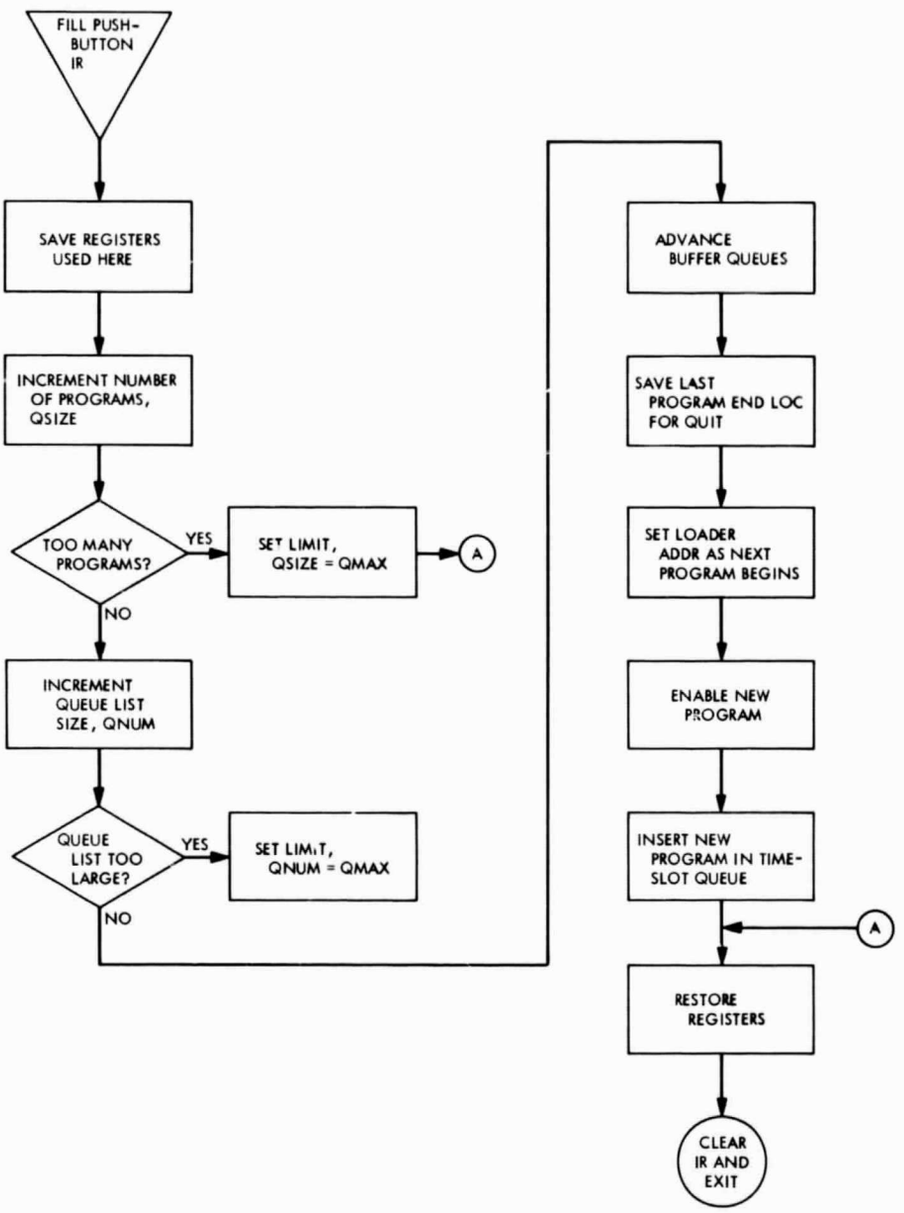

BUSY AND READY PROGRAMMED OPERATOR MODULES
BUSY POP CALL
READY POP CALL
WAIT FOR NEXT INNING IF NOT ENOUGH TIME
WAIT FOR NEXT INNING IF NOT ENOUGH TIME
SAVE X REGISTER
SAVE X REGISTER
PICK UP, SAVE TEST INSTRUCTION FOR READY POP SUBROUTINE
SET UP ACCESS OF TEST BY QSWITC VIA TSTFLG
RELEASE QUEUE INNING
SAVE POP REFERENCE
EXECUTE TSTFLG INSTRUCTION
NO SKIP
RESTORE X REGISTER
SKIP
RESTORE X REGISTER
RESTORE X
HANG UP UNTIL TEST FAILS SKIP (READY BPDOWN CALL)
EXIT TO POP REF +1
RETURN TO POP REF +1

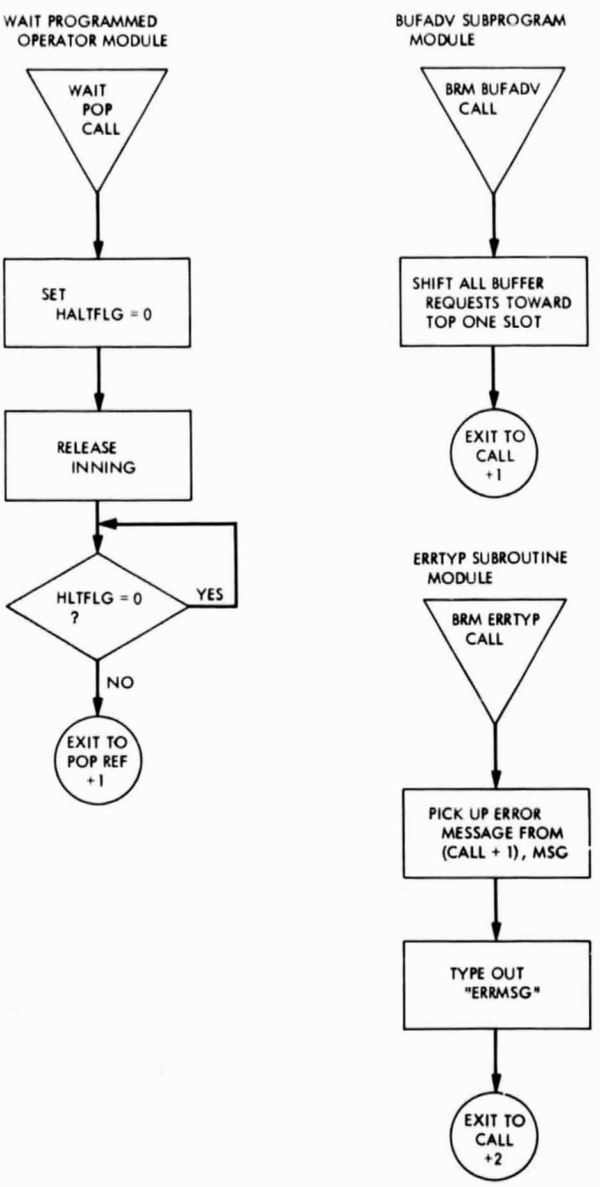
W AND Y BUFFER END-OF-WORD
AND END-OF-TRANSMISSION
INTERRUPT ROUTINES MODULE



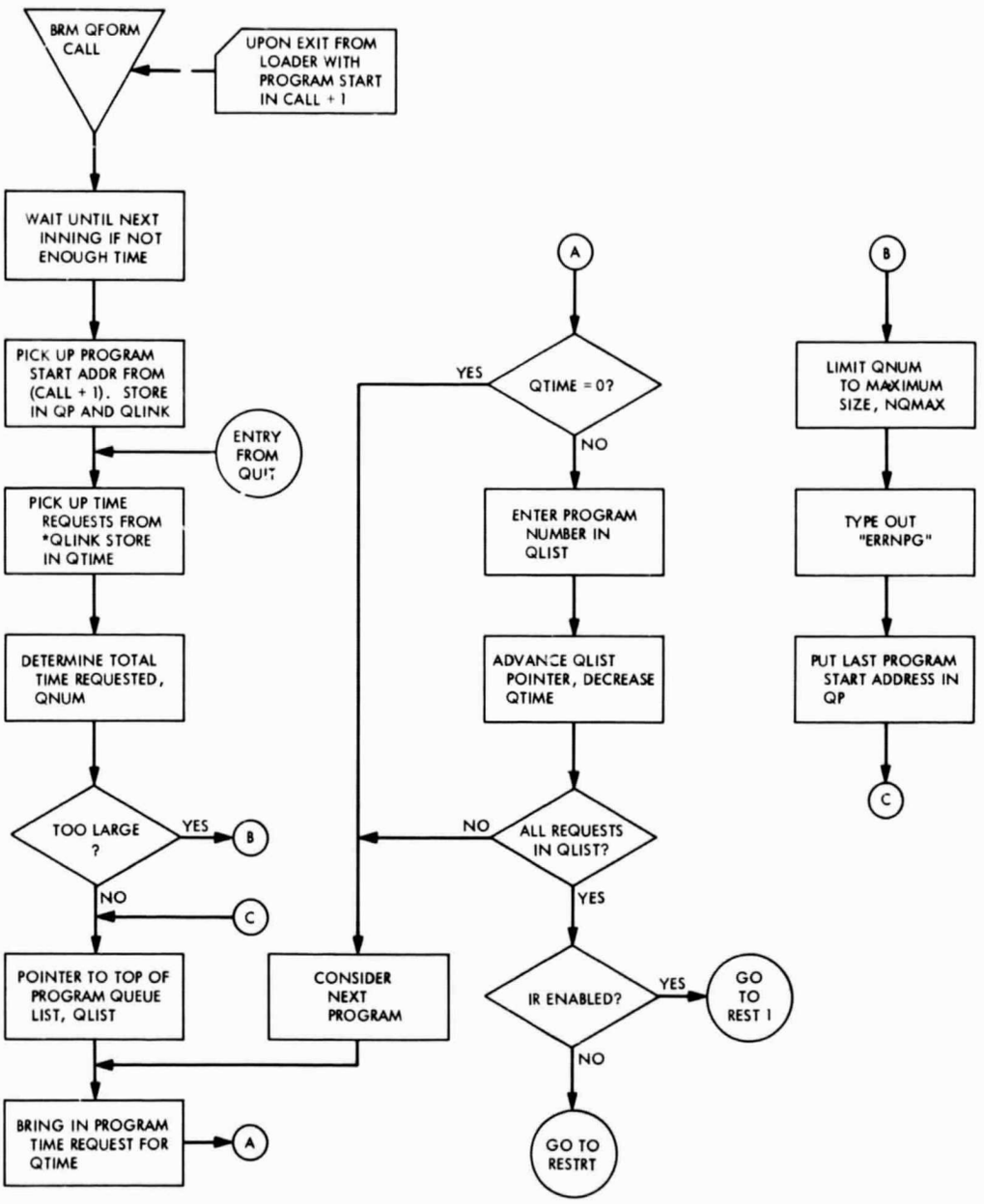

BIO PROGRAMMED OPERATOR AND BUFFER IR CONNECT SUBROUTINE MODULES
BIO POP CALL
WAIT UNTIL NEXT INNING IF NOT ENOUGH TIME
DETERMINE BUFFER REQUIRED BY (POP REF + 1)
FIND END OF BUFFER AND X-SAVE QUEUES
FROM PROGRAM NO. + POP REFERENCE WORD
SEARCH BUFFER QUEUE FOR CALL BY THIS PROGRAM
LISTED ALREADY
RELEASE INNING
STORE REQUEST, X REGISTER IN QUEUES FOR CONBIR
EXIT TO POP REF +4
BUFFER QUEUE NOT EMPTY TEST
IN QSWITC
FIND POP REFERENCE, PROGRAM NO., AND X-SAVE OF BUFFER REQUEST
ADVANCE BUFFER REQUEST QUEUE
SET UP DEVICE EOM FROM (POP REF + 1) INTO THIS SEQUENCE
A
SET UP NUMBER OF WORDS TO I/O FROM *POP REF
SET UP EOTXIR INSTRUCTION FROM (POP REF + 4)
EOTLOC
SET UP I/O INSTRUCTION FROM (POP REF + 2)
IOLOC
SET UP EOWXIR INSTRUCTION FROM (POP REF + 3)
TOLOC
A
ACTIVATE DEVICE
RETURN TO QSWITC CALL + 1

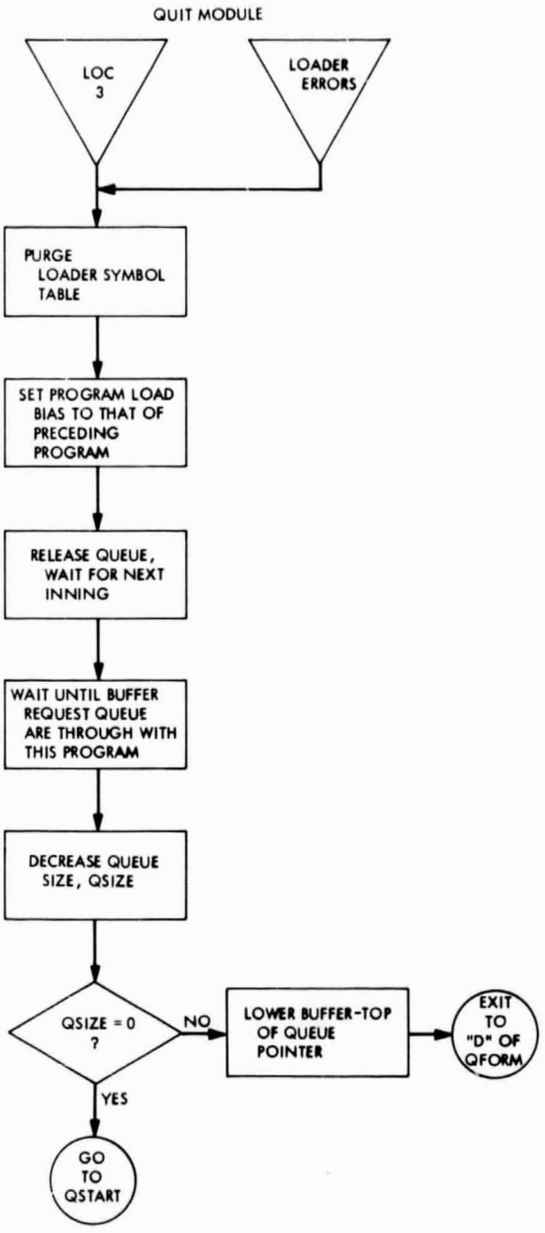
FILL PUSHBUTTON IR MODULE

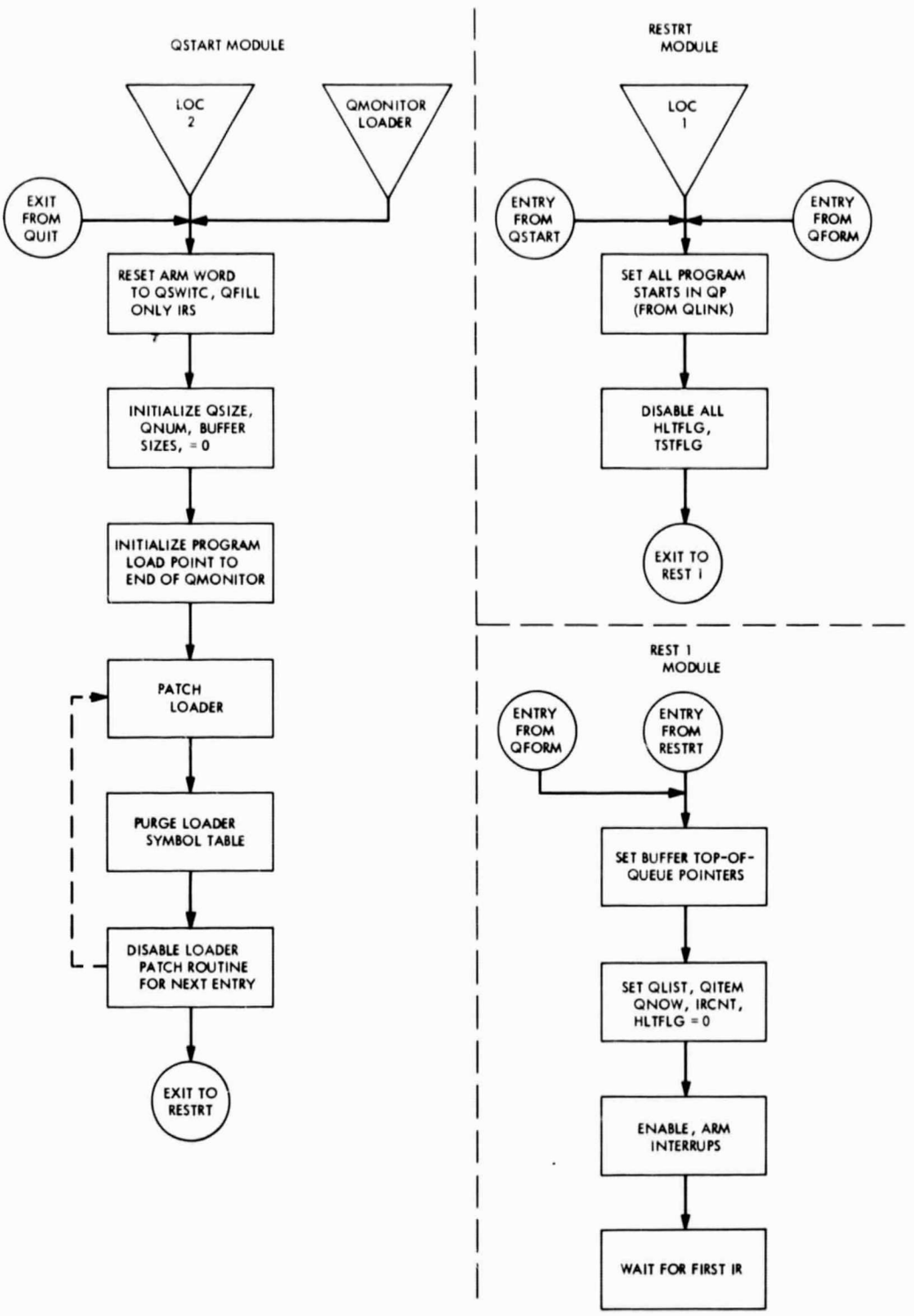


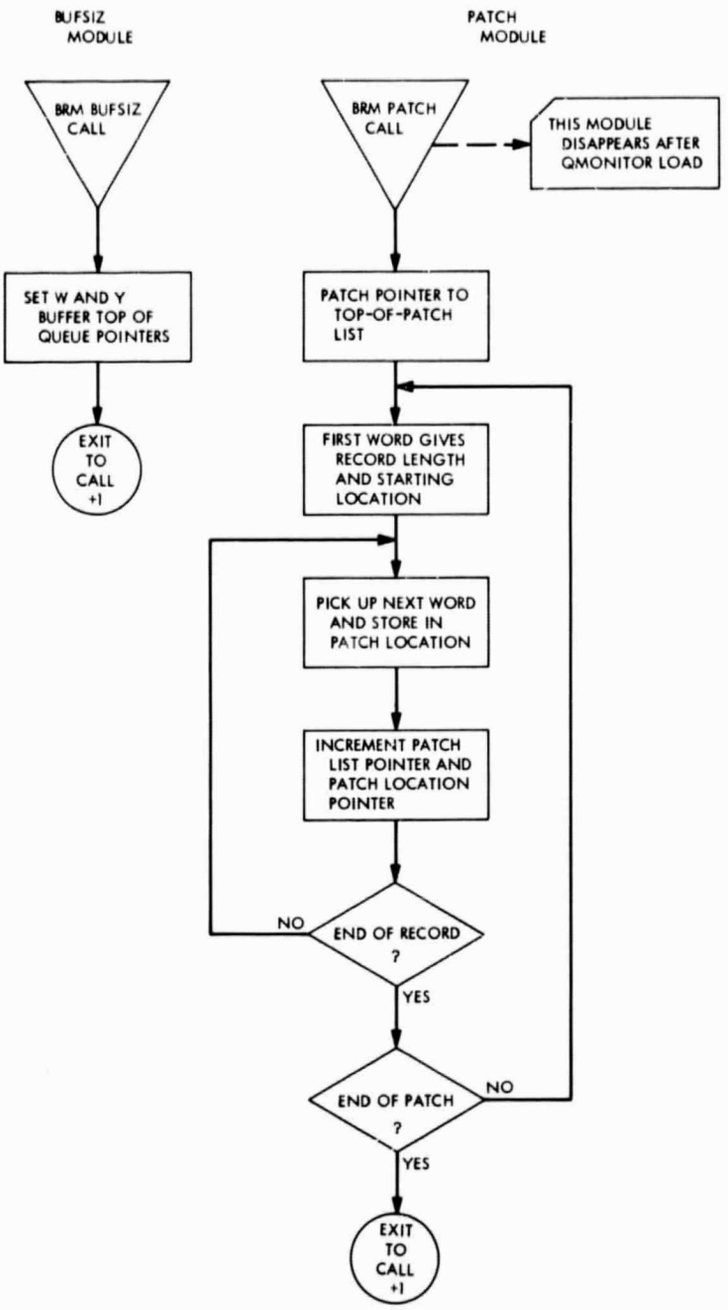

WAIT PROGRAMMED OPERATOR MODULE
WAIT POP CALL
SET HALTFLG = 0
RELEASE INNING
HLTFLG = 0 ?
YES
NO
EXIT TO POP REF +1
BUFADV SUBPROGRAM MODULE
BRM BUFADV CALL
SHIFT ALL BUFFER REQUESTS TOWARD TOP ONE SLOT
EXIT TO CALL +1
ERRTYP SUBROUTINE MODULE
BRM ERRTYP CALL
PICK UP ERROR MESSAGE FROM (CALL + 1), MSG
TYPE OUT "ERRMSG"
EXIT TO CALL +2

QFORM MODULE




QUIT MODULE
LOC 3
LOADER ERRORS
PURGE LOADER SYMBOL TABLE
SET PROGRAM LOAD BIAS TO THAT OF PRECEDING PROGRAM
RELEASE QUEUE, WAIT FOR NEXT INNING
WAIT UNTIL BUFFER REQUEST QUEUE ARE THROUGH WITH THIS PROGRAM
DECREASE QUEUE SIZE, QSIZE
QSIZE = 0 ?
NO
LOWER BUFFER-TOP OF QUEUE POINTER
EXIT TO "D" OF QFORM
YES
GO TO QSTART


QSTART MODULE
LOC 2
QMONITOR LOADER
EXIT FROM QUIT
RESET ARM WORD TO QSWITC, QFILL ONLY IRS
INITIALIZE QSIZE, QNUM, BUFFER SIZES, = 0
INITIALIZE PROGRAM LOAD POINT TO END OF QMONITOR
PATCH LOADER
PURGE LOADER SYMBOL TABLE
DISABLE LOADER PATCH ROUTINE FOR NEXT ENTRY
EXIT TO RESTRT
RESTRT MODULE
LOC 1
ENTRY FROM QSTART
ENTRY FROM QFORM
SET ALL PROGRAM STARTS IN QP (FROM QLINK)
DISABLE ALL HLTFLG, TSTFLG
EXIT TO REST 1
REST 1 MODULE
ENTRY FROM QFORM
ENTRY FROM RESTRT
SET BUFFER TOP-OF-QUEUE POINTERS
SET QLIST, QITEM QNOW, IRCNT, HLTFLG = 0
ENABLE, ARM INTERRUPS
WAIT FOR FIRST IR


BUFSIZ MODULE
PATCH MODULE
BRM BUFSIZ CALL
BRM PATCH CALL
THIS MODULE DISAPPEARS AFTER QMONITOR LOAD
SET W AND Y BUFFER TOP OF QUEUE POINTERS
PATCH POINTER TO TOP-OF-PATCH LIST
EXIT TO CALL +1
FIRST WORD GIVES RECORD LENGTH AND STARTING LOCATION
PICK UP NEXT WORD AND STORE IN PATCH LOCATION
INCREMENT PATCH LIST POINTER AND PATCH LOCATION POINTER
NO
END OF RECORD ?
YES
END OF PATCH ?
NO
YES
EXIT TO CALL +1

```
1  ********************************************************************
2  ********************************************************************
3  *                                                                  *
4  *      PROGRAM TO TIME-SHARE SDS-900 COMPUTER                      *
5  *                                                                  *
6  ********************************************************************
7  *                                                                  *
8  *      1.  LOWEST PRIORITY IR PATCHED TO PERIODIC PULSE AT DESIRED RATE*
9  *      2.  NEXT LOWEST IR PATCHED TO FILL PUSHBUTTON.              *
10 *      3.  ALL USER PROGRAMS ARE PROG 0,EXCEPT FOR ASSIGNED  IR    *
11 *          LOCATIONS AND POP LINKS.                                *
12 *      4.  USER PROGRAMS ARE COMPLETE WITH END TRANSFER.           *
13 *      5.  ALL HALTS ARE ACCOMPLISHED BY A WAIT POP, AND RELEASED IN *
14 *          AN IR ROUTINE BY                                        *
15 *                                                                  *
16 *                          LDX     =-PROGNUM                       *
17 *                          LDA     =-1                             *
18 *                          STA     HLTFLG,2                        *
19 *                                                                  *
20 *          -PROGNUM IS ACCESSED IN USER NON-IR PROGRAM BY  LDX GNOW.*
21 *                                                                  *
22 *      6.  ALL BUFFER I/O IS DONE WITH BIO POP, NON-INTERLACED:    *
23 *                                                                  *
24 *              BIO  WDCNT          SET UP BUFFER IR TO I/O  (WDCNT) WDS*
25 *              EOM  DEVICE         CONNECT THIS DEVICE             *
26 *              XXX  FSTLOC         XIM OR MIX, STARTING  IN FSTLOC *
27 *              XXX                 PERFORM THIS IN EOW IR AFTER FINISH *
28 *              XXX                 PERFORM THIS IN EOT IR          *
29 *                                                                  *
30 *      7.  ALL WAIT-FOR-EVENT TESTS ARE DONE VIA READY AND BUSY POPS:*
31 *                                                                  *
32 *              READY   TEST                                        *
33 *              BUSY    TEST                                        *
34 *                                                                  *
35 *          (TEST)=SKIP TEST NOT USING A,B,X REGISTER               *
36 *                                                                  *
37 *      8.  USER IR-S ARE ARMED/DISARMED VIA ARM/DARM POP:          *
38 *                                                                  *
39 *              ARM     IRNUM       ARM, OR                         *
40 *              DARM    IRNUM       DISARM THE INDICATED IR(0 TO 15)*
41 *                                                                  *
42 ********************************************************************
43 *                                                                  *
44 *      NO OVERFLOW SAVE YET IN MONITOR                             *
45       PAGE
46 ********************************************************************
47 *                                                                  *
48 *      9.  TH  REMAINDER OF A QUEUE INNING IS RELEASED VIA QRLS POP*
49 *      10. PROGRAM WAIT FOR BPN TOGGLE IS THROUGH PAUSE N POP.     *
50 *      11. NO POPS INSIDE IR ROUTINES, BIO POP, OR READY POP.      *
51 *      12. USER PROGRAMS HAVE THE FOLLOWING FORMAT:                *
52 *                                                                  *
53 *                      PROG   0                                    *
54 *              P                                                   *
55 *               R                                                  *
56 *                A                                                 *
57 *              START  NOP    N     PROG GETS AT LEAST N/QMAX OF QUEUE*
58 *                  G                                               *
59 *                   R                                              *
60 *                    A                                             *
61 *                     M                                            *
62 *                     END    START                                 *
63 *                                                                  *
64 *      13. USER PROGRAMS REFER TO QUEUE MONITOR POPS VIA OPD:      *
65 *                                                                  *
66 *              QRLS   OPD  017000000    RELEASE REST OF INNING     *
67 *              ARM    OPD  017100000    ARM IR (0 TO 15)           *
68 *              DARM   OPD  017200000    DISARM IR (0 TO 15)        *
69 *              BUSY   OPD  017300000    PROCEED IF TEST FAILS      *
70 *              READY  OPD  017400000    PROCEED IF TEST PASSES     *
71 *              PAUSE  OPD  017500000    OPT N WAIT                 *
```

```
                        72  *            BIO     APD  017600000    BUFFER I/O                    *
                        73  *            WAIT    APD  017700000    WAIT FOR HALT FLAG            *
                        74  *                                                                     *
                        75  ***********************************************************************
                        76          PAGE
                        77  ***********************************************************************
                        78  *                                                                     *
                        79  *      TO  LOAD:                                                      *
                        80  *                                                                     *
                        81  *      1.  STANDARD FILL QUEUE MONITOR-LOADER TAPE.                   *
                        82  *      2.  LOAD USER TAPES BY FILL PUSHBUTTON.                        *
                        83  *                                                                     *
                        84  ***********************************************************************
                        85          PAGE
                        86  ***********************************************************************
                        87  *                                                                     *
                        88  *      PROGRAM TO SET RELATIVE ORIGIN TO ZERO                         *
                        89  *                                                                     *
                        90  ***********************************************************************
00454                   91          ORG     300
00454  0 46 30003       92  QZERO   CLR
00455  0 01 00000       93          BRU     QUBLDR
       00000454         94          END     QZERO
00455  QUBLDR

                         1  ***********************************************************************
                         2  *                                                                     *
                         3  *      DEFINITIONS PECULIAR TO THIS ASSEMBLY:                         *
                         4  *         (QUEUE SIZE AND COMPUTER CONFIGURATION                      *
                         5  *                                                                     *
                         6  ***********************************************************************
       00000012          7  IRRATE EQU     10                  NUMBER OF IR*S PER INNING
       00000207          8  LASTIR EQU     0207                LAST IR IN 207
       00010000          9  MEMSIZ EQU     010000              *4096 WORDS MEMORY
       00000012         10  QMAX   EQU     10                  NUMBER OF PERMISSIBLE USER PROGRAMS
                        11  *                                                                     *
                        12  ***********************************************************************
                        13  *                                                                     *
                        14  *      DEFINITIONS OF LOADER LOCATIONS                                *
                        15  *                                                                     *
                        16  ***********************************************************************
       00004000         17  LDZERO EQU     MEMSIZ-04000        LOADER ZERO REFERENCE
       00007646         18  BIAS   EQU     LDZERO+03646        INITIAL BIAS ADDRESS
       00006576         19  QUBLDR EQU     LDZERO+02576
                        20  *                                                                     *
                        21  ***********************************************************************
                        22          PAGE
                        23  ***********************************************************************
                        24  *                                                                     *
                        25  *      PROGRAM START LINKAGES                                         *
                        26  *                                                                     *
                        27  ***********************************************************************
                        28  *                                                                     *
00001                   29          ORG     1
00001  0 01 01016       30          BRU     RESTRT              GO TO START ALL PROGRAMS AT BEGIN
00002  0 01 01000       31          BRU     QSTART              INITIALIZE QUEUE TO NO PROGRAMS
00003  0 01 00753       32          BRU     QUIT                REMOVE LAST PROG
                        33  ***********************************************************************
                        34  *                                                                     *
                        35  *      POWER OFF IR                                          (PWROFF)*
                        36  *                                                                     *
                        37  ***********************************************************************
00004  0 00 00000       38  PWROFF PZE
00005  0 01 00005       39          BRU     *
                        40  *                                                                     *
00006  52255151         41  PWRMSG TEXT    < ERROR >
00007  47665152
                        42  ***********************************************************************
                        43  *                                                                     *
                        44  *      CELLS 010 TO 027 ARE VARIABLE STORAGE                          *
                        45  *                                                                     *
                        46  ***********************************************************************
```

```
                                47          PAGE
                                48  ********************************************************************************
                                49  *                                                                              *
                                50  *       EBN, EBT, PWRER IR LINKAGES                                            *
                                51  *                                                                              *
                                52  ********************************************************************************
00030                           53          ABSG    030
00030  0 43 00454               54          BRM     EBNYIR              EBN Y IR
00031  0 43 00465               55          BRM     EBNWIR              EBN W IR
00032  0 43 00451               56          BRM     EBTYIR              EBT Y IR
00033  0 43 00446               57          BRM     EBTWIR              EBT W IR
00034                           58          RES     2                   DATA STORAGE
00036  0 43 00052               59          BRM     PWRBN               PWR BN IR
00037  0 43 00004               60          BRM     PWRBFF              PWR BFF IR
                                61  ********************************************************************************
                                62  *                                                                              *
                                63  *       CELLS 040 TO 051 ARE STORAGE                                           *
                                64  *                                                                              *
                                65  ********************************************************************************
                                66          PAGE
                                67  ********************************************************************************
                                68  *                                                                              *
                                69  *       PWRED BN IR ROUTINE                                            (PWRBN)*
                                70  *                                                                              *
                                71  ********************************************************************************
00052                           72          ABSG    052
00052  0 00 00000               73  PWRBN   PZE
00053  0 71 01071               74          LDX     N1000               /
00054  0 67 20060               75          LCY     48                  / DELAY 100 MILLISEC
00055  0 41 00054               76          BRX     $-1                 /
00056  0 40 24000               77          SKS     024000              PWRER BFF BCCURED
00057  0 01 00057               78          BRU     $                   YES
00060  0 02 20004               79          DIR
00061  0 02 00000               80          EBM     0                   DSCW
00062  0 02 00100               81          EBM     0100                DSCY
00063  0 40 21000               82          BRTW
00064  0 01 00063               83          BRU     $-1
00065  0 02 02641               84          TYP     0,1,4
00066  0 12 00006               85          MIW     PWRMSG
00067  0 12 00007               86          MIW     PWRMSG+1
00070  0 02 14000               87          TBPW
00071  0 01 40072               88          BRU     *$+1                CLEAR IR
00072  0 01 01025               89          BRU     REST1
                                90  ********************************************************************************
                                91  *                                                                              *
                                92  *       PROGRAMMED OPERATOR TO RELEASE REST OF INNING                  (QRLS)*
                                93  *               DESTROYS B REGISTER.                                           *
                                94  *                                                                              *
                                95  ********************************************************************************
      00000073                  96  RLSPOP  EQU     $
00073  0 75 01074               97          LDB     PC
00074  0 36 00042               98          STB     IRCNT
00075  0 53 00042               99          SKN     IRCNT
00076  0 01 00075              100          BRU     $-1
00077  0 51 00000              101          BRR     0
                               102  ********************************************************************************
                               103          PAGE
                               104  ********************************************************************************
                               105  *                                                                              *
                               106  *       LINKS TO PROGRAMMED OPERATORS                                          *
                               107  *                                                                              *
                               108  ********************************************************************************
      17000000                 109  QRLS    BPD     017000000           QUEUE INNING RELEASE
      17100000                 110  ARM     BPD     017100000           ARM IR
      17200000                 111  DARM    BPD     017200000           DISARM IR
      17300000                 112  BUSY    BPD     017300000           PROCEED IF TEST FAILS
      17400000                 113  READY   BPD     017400000           STOP IF TEST FAILS
      17500000                 114  PAUSE   BPD     017500000           BPT WAIT
      17600000                 115  BIO     BPD     017600000           BUFFER I/O
      17700000                 116  WAIT    BPD     017700000           WAIT FOR HALT FLAG
                               117  *                                                                              *
```

```
00170                    118           ARRG    0170
00170  0 01 00073        119           BRU     RLSPAP              RELEASE REMAINDER OF INNING
00171  0 01 00340        120           BRU     ARMPAP              ARM IR PAP
00172  0 01 00360        121           BRU     DRMPAP              DISARM IR PAP
00173  0 01 00372        122           BRU     BSYPAP
00174  0 01 00406        123           BRU     RDYPAP
00175  0 01 00430        124           BRU     PSEPAP
00176  0 01 00474        125           BRU     RIOPAP
00177  0 01 00625        126           BRU     WATPAP
                         127     *
                         128     ************************************************************************
                         129     *
                         130     *       LINKS TO QUEUE IRS                                             *
                         131     *
                         132     ************************************************************************
00206                    133           ARRG    LASTIR-1
00206  0 43 00210        134           BRM     QFILL
00207  0 43 00253        135           BRM     QSWITC
                         136     ************************************************************************
                         137           PAGE
                         138     ************************************************************************
                         139     *
                         140     *       FILL INTERRUPT ROUTINE                                 (QFILL)*
                         141     *
                         142     ************************************************************************
00210  0 00 00000        143     QFILL   PZE
00211  0 35 01172        144           STA     QFA                 /
00212  0 37 01322        145           STX     QFX                 /SAVE A,X
00213  0 60 00024        146           MDF     QSIZE               INCREMENT NUMBER OF PROGRAMS.
00214  0 20 00000        147           NOP
00215  0 76 01072        148           LDA     NQMAX               /
00216  0 73 00024        149           SKG     QSIZE               /
00217  0 01 00222        150           BRU     $+3                 /
00220  0 35 00024        151           STA     QSIZE               /
00221  0 01 00247        152           BRU     QF2                 /IF TOO MANY PROGS IGNORE AND EXIT
00222  0 60 00021        153           MDF     QNUM                INCREMENT NUMBER OF ITEMS IN QUEUE
00223  0 20 00000        154           NOP
00224  0 73 00021        155           SKG     QNUM                /
00225  0 01 00227        156           BRU     $+2                 /
00226  0 35 00021        157           STA     QNUM                /IF LIST OVERFLOWS LIMIT TO MAX
00227  0 43 01040        158           BRM     BUFSIZ
00230  0 71 01074        159           LDX     P0                  /
00231  0 43 00634        160           BRM     BUFADV              /
00232  0 71 01075        161           LDX     P1                  /
00233  0 43 00634        162           BRM     BUFADV              /ADVANCE W AND Y BUFFER QUEUES
00234  0 71 00024        163           LDX     QSIZE               /
00235  0 76 07646        164           LDA     BIAS
00236  0 35 00014        165           STA     LBIAS               SAVE LAST PROG BIAS
00237  0 16 01065        166           MRG     MBO
00240  2 35 01172        167           STA     QA,2
00241  0 76 01102        168           LDA     QUBLAD              /
00242  2 35 01274        169           STA     QP,2                /START NEW PROGRAM IN LOADER
00243  2 37 01144        170           STX     HLTFLG,2            ENABLE QUEUE SWITCH TO NEW PROGRAM
00244  0 71 00021        171           LDX     QNUM                /
00245  0 76 00024        172           LDA     QSIZE               /
00246  2 35 01261        173           STA     QLIST,2             /INSERT NEW PROGRAM IN QUEUE
       00000247          174     QF2     EQU     $
00247  0 76 01172        175           LDA     QFA                 /
00250  0 71 01322        176           LDX     QFX                 /RESTORE A,X
00251  0 01 40210        177           BRU     *QFILL
                         178     ************************************************************************
                         179           PAGE
                         180     ************************************************************************
                         181     *
                         182     *       QUEUE SWITCHING INTERRUPT MODULE                       (QSWITC)*
                         183     *
                         184     ************************************************************************
                         185     *
                         186     *       FUNCTION: TO GIVE TIME-SLOT TO NEXT PROGRAM IN QUEUE LIST.     *
                         187     *
                         188     ************************************************************************
```

```
                                  189  *                                                                      *
                                  190  *       SYMBOL TABLE DEFINITIONS:                                      *
                                  191  *                                                                      *
                                  192  *               IRCNT               NUMBER OF IR'S LEFT THIS SLOT, NEG I*
                                  193  *               XQSW                X REG TEMP SAVE                    *
                                  194  *               IRMAX               NUMBER OF IR'S PER TIME-SLOT, NEG I *
                                  195  *               QNOW                PROGRAM NOW IN CONTROL, NEG I      *
                                  196  *               QA                  \                                  *
                                  197  *               QB                   \                                 *
                                  198  *               QX                    ARRAYS FOR SAVING A,B,X,P,CELL 0  *
                                  199  *               QP                   /                                 *
                                  200  *               QQ                  /                                  *
                                  201  *               QTOPAD              CELL WITH LOC OF TOP OF W BUFF QUEUE*
                                  202  *               P1                  =+1, I                             *
                                  203  *               QITEM               POINTER TO QUEUED PROG LIST, NEG I *
                                  204  *               QNUM                NUMBER OF ITEMS IN PROG LIST, NEG I *
                                  205  *               QQ                  =+Q , I                            *
                                  206  *               QLIST               ARRAY WITH QUEUE OF PROG NUMBERS   *
                                  207  *               HLTFLG              ARRAY WITH PROGRAM HALTS: GO IF NEG.*
                                  208  *               TSTFLG              ARRAY WITH PROGRAM TEST INSTR:     *
                                  209  *                                    GO IF 0; ELSE EXU TSTFLG; GO IF SKP*
                                  210  *               N1                  =-1, I                             *
                                  211  *                                                                      *
                                  212  ************************************************************************
                                  213  *       ENTRY:  IR TO CELL DESIGNATED DURING COMPILE, BRM QSWIIC        *
                                  214  *       EXIT:   CONTINUES AT PREVIOUS INTERRUPTION OF PROGRAM IN QUEUE  *
                                  215  ************************************************************************
                                  216  *                                                                      *
                                  217  *       SUBROUTINES:                                                   *
                                  218  *               BRM CONBIR          TO CONNECT BUFFER IR'S FOR NEXT I/O;*
                                  219  *                                    RET $+1                           *
                                  220  *                                                                      *
                                  221  ************************************************************************
00000252                          222  QSW0    EQU     $                   ENTRY
00252  0 00 00000                 223  QSWITC  PZE                         CALL SAVE CELL
00253  0 53 00042                 224  QSW1    SKN     IRCNT               END OF INNING
00254  0 01 00257                 225          BRU     $+3                  YES.
00255  0 61 00042                 226  QSW2    MIN     IRCNT                NO. INCREMENT IR COUNT
00256  0 01 40252                 227  QSW3    BRU     *QSWITC                  RETURN TO PRESENT PROGRAM
00257  0 37 00027                 228  QSW4    STX     XQSW                SAVE X REG
00260  0 71 01061                 229          LDX     IRMAX               RESTORE IRCNT
00261  0 37 00042                 230          STX     IRCNT                TO (NEG) MAX VALUE
00262  0 71 00020                 231  QSW5    LDX     QNOW                PICK UP NUMBER THIS PROG
00263  2 35 01172                 232          STA     QA,2                \
00264  2 36 01205                 233          STB     QB,2                 \
00265  0 76 00027                 234          LDA     XQSW                  \
00266  2 35 01322                 235          STA     QX,2                   \ SAVE A,B,X,P REGS AND CELL 0
00267  0 76 00252                 236          LDA     QSWITC                 / FOR THIS PROG FOR NEXT ENTRY
00270  2 35 01274                 237          STA     QP,2                  /
00271  0 76 00000                 238          LDA     0                    /
00272  2 35 01157                 239          STA     QQ,2                /
00273  0 53 40040                 240  QSW6    SKN     *QTOPAD             W BUFF QUEUE EMPTY
00274  0 01 00301                 241          BRU     $+5                  YES.
00275  0 40 21000                 242  QSW7    BRTW                         NO. IS W BUFF BUSY
00276  0 01 00301                 243          BRU     $+3                   YES.
00277  0 71 01074                 244  QSW8    LDX     P0                    NO. PICK UP W BUFFER DESIGNATION
00300  0 43 00550                 245          BRM     CONBIR                  CONNECT W BUFF IR'S, RETURN
00301  0 53 40041                 246  QSW9    SKN     *QTOPAD+1           Y BUFF QUEUE EMPTY
00302  0 01 00307                 247          BRU     $+5                  YES.
00303  0 40 22000                 248  QSW10   BRTY                         NO. IS Y BUFF BUSY
00304  0 01 00307                 249          BRU     $+3                   YES.
00305  0 71 01075                 250  QSW11   LDX     P1                    NO. PICK UP Y BUFF DESIGNATION
00306  0 43 00550                 251          BRM     CONBIR                  CONNECT Y BUFFER IR'S, RETURN
00307  0 71 00017                 252  QSW12   LDX     QITEM               PICK UP QUEUED LIST POINTER
00310  0 41 00312                 253          BRX     $+2                 INCREMENT LIST. END OF LIST
00311  0 71 00021                 254          LDX     QNUM                 YES. RECYCLE TO BEGINNING OF LIST.
00312  0 37 00017                 255          STX     QITEM                NO. STORE POINTER
00313  2 71 01261                 256          LDX     QLIST,2             PICK UP NEXT PROG NUMBER.
00314  0 37 00020                 257          STX     QNOW                STORE PROG NUMBER
00315  2 53 01144                 258  QSW13   SKN     HLTFLG,2            THIS PROG IN WAIT
00316  0 01 00273                 259          BRU     QSW6                 YES, SO SKIP IT.
```

```
00317  2 76 01335     260 QSW14  LDA     TSTFLG,2          NO, PICK UP PROG TEST REQUEST,
00320  0 72 01067     261        SKA     N1                ACTION REQUIRED
00321  0 01 00330     262        BRU     QSW17              YES
00322  0 71 00020     263 QSW15  LDX     QNOW              \
00323  2 76 01274     264        LDA     QP,2               \
00324  0 35 00252     265        STA     QSWITC              \
00325  2 76 01167     266        LDA     QO,2                 \ PICK UP A,B,X,P REGS AND CELL 0
00326  0 35 00000     267        STA     0                    / FOR NEW PROGRAM
00327  2 76 01172     268        LDA     QA,2                /
00330  2 75 01206     269        LDB     QB,2               /
00331  2 71 01322     270        LDX     QX,2              /
00332  0 01 40252     271 QSW16  BRU     *QSWITC           EXIT TO NEW PROGRAM,
00333  2 23 01335     272 QSW17  EXU     TSTFLG,2          PERFORM REQUESTED ACTION, PROG READY
00334  0 01 00273     273        BRU     QSW6               NO, SO SKIP IT
00335  0 46 30000     274        CLR                        YES,
00336  2 35 01335     275        STA     TSTFLG,2            CLEAR TEST REQUEST,
00337  0 01 00322     276        BRU     QSW15                GO ON
                      277 ***************************************************************
                      278        PAGE
                      279 ***************************************************************
                      280 *                                                             *
                      281 *      INTERRUPT ARM/DISARM PROPS.                  (ARM,DARM)*
                      282 *                                                             *
                      283 ***************************************************************
     00000340         284 ARMPRP EQU     $
00340  0 53 00042     285        SKN     IRCNT             CHECK FOR
00341  0 01 00340     286        BRU     $-1                ENOUGH TIME
00342  0 35 00045     287        STA     TMPA              /
00343  0 36 00046     288        STB     TMPB              /
00344  0 37 00025     289        STX     TMPX              SAVE REGS
00345  0 77 40000     290        EAX     *0                /
00346  0 76 01050     291        LDA     ARM0              /
00347  2 66 00000     292        RSH     0,2               /
00350  0 16 00047     293        MRG     ARMG1             1 IN IR BIT + PREVIOUS ARMING
00351  0 35 00047     294 AM1    STA     ARMG1             /
00352  0 02 20020     295        AIR                       /
00353  0 13 00047     296        PAT     ARMG1             ARM IR*S
00354  0 76 00045     297        LDA     TMPA              /
00355  0 75 00046     298        LDB     TMPB              /
00356  0 71 00025     299        LDX     TMPX              /RESTORE,
00357  0 51 00000     300        BRR     0                 RETURN
                      301 *
     00000360         302 DRMPRP EQU     $
00360  0 53 00042     303        SKN     IRCNT             CHECK FOR ENOUGH TIME
00361  0 01 00360     304        BRU     $-1               /
00362  0 35 00045     305        STA     TMPA              /
00363  0 36 00046     306        STB     TMPB              /
00364  0 37 00025     307        STX     TMPX              /SAVE REGS
00365  0 77 40000     308        EAX     *0                /
00366  0 76 01060     309        LDA     DARM0             /
00367  2 66 00000     310        RSH     0,2               /
00370  0 14 00047     311        ETR     ARMG1             0 IN IR BIT
00371  0 01 00351     312        BRU     AM1
                      313 ***************************************************************
                      314        PAGE
                      315 ***************************************************************
                      316 *                                                             *
                      317 *      PRP TO MAKE BUSY TEST, NO WAIT IN QUEUE IF READY.          (BUSY)*
                      318 *           DESTROYS  A,B REGISTERS                           *
                      319 *                                                             *
                      320 ***************************************************************
     00000372         321 BSYPRP EQU     $
00372  0 53 00042     322        SKN     IRCNT             /
00373  0 01 00372     323        BRU     $-1               WAIT FOR ENOUGH TIME
00374  0 37 00025     324        STX     TMPX              SAVE X
00375  0 76 40000     325        LDA     *0                /
00376  0 16 01065     326        MRG     MB0               /
00377  0 71 00020     327        LDX     QNOW              /
00400  2 35 01144     328        STA     HLTFLG,2          SET UP TEST IN INVERTER SUBRTN
00401  0 76 00000     329        LDA     0                 SAVE PRP REF
00402  0 71 00025     330        LDX     TMPX              RESTOREX
```

```
00403  1 74 00441      331          READY   BPDWN               INVERTED READY TEST
00404  0 35 00000      332          STA     0                   RESTORE PRP REF
00405  0 51 00000      333          BRR     0                   RETURN
                        334  ********************************************************************
                        335  *                                                                  *
                        336  *       PRP TO MAKE DEVICE READY TEST. NO WAIT IN QUEUE  IF BUSY,(READY)*
                        337  *               DESTROYS B REGISTER                                *
                        338  *                                                                  *
                        339  ********************************************************************
     00000406           340  RDYPRP  EQU     $
00406  0 53 00042      341          SKN     IRCNT               /
00407  0 01 00406      342          BRU     $+1                 SKIP REST OF INNING IF NOT ENOUGH TIME
00410  0 37 00025      343          STX     TMPX                /
00411  0 75 40000      344          LDB     *0                  /
00412  0 71 00020      345          LDX     QNPA                /
00413  2 36 01335      346          STB     TSTFLG,2            /SET TEST FOR QSWITCH ACCESS
00414  0 36 00415      347          STB     $+1                 /
00415  0 23 00000      348          EXU                         /EXECUTE TEST
00416  0 01 00424      349          BRU     R1                  RETURN IF FAIL
00417  0 75 01074      350          LDB     PO                  /
00420  0 71 00020      351          LDX     QNPA                /
00421  2 36 01335      352          STB     TSTFLG,2            RESTORE INDICATOR IF READY
00422  0 71 00025      353          LDX     TMPX
00423  0 51 00000      354          BRR     0                    AND RETURN
                        355  *                                                                  *
00424  0 71 00025      356  R1      LDX     TMPX                /
00425  0 75 01074      357          LDB     PO
00426  0 36 00042      358          STB     IRCNT
00427  0 01 00406      359          BRU     RDYPRP
                        360  ********************************************************************
                        361          PAGE
                        362  ********************************************************************
                        363  *                                                                  *
                        364  *       PRP TO PAUSE, PROCEED ON BPT N TOGGLE                      (PAUSE)
                        365  *               DESTROYS (A),(B),(X) REGISTERS                     *
                        366  *                                                                  *
                        367  ********************************************************************
     00000430           368  PSEPRP  EQU     $
00430  0 76 00000      369          LDA     0                   SAVE EXIT
00431  0 77 40000      370          EAX     *0
00432  3 74 01051      371          READY   BPTST,2
00433  2 75 01051      372          LDB     BPTST,2
00434  0 71 00020      373          LDX     QNPA                /
00435  2 36 01144      374          STB     HLTFLG,2
00436  1 74 00441      375          READY   BPDWN               /
00437  0 35 00000      376          STA     0                   SET UP EXIT
00440  0 51 00000      377          BRR     0
                        378  *                                                                  *
00441  0 43 00442      379  BPDWN   BRM     $+1
00442  0 00 00000      380          PZE
00443  2 23 01144      381          EXU     HLTFLG,2
00444  0 61 00442      382          MIN     BPDWN+1
00445  0 51 00442      383          BRR     BPDWN+1
                        384  ********************************************************************
                        385          PAGE
                        386  ********************************************************************
                        387  *                                                                  *
                        388  *       END OF WORD AND END OF TRANSMISSION IR ROUTINES             *
                        389  *                                                                  *
                        390  ********************************************************************
                        391  *                                                                  *
00446  0 00 00000      392  EOTWIR  PZE                         END-OF-TRANSMISSION W BUFF IR
00447  0 23 00010      393          EXU     EOTLOC              TERMINATE INST
00450  0 01 40446      394          BRU     *$-2
                        395  *                                                                  *
00451  0 00 00000      396  EOTYIR  PZE                         END-OF-TRANSMISSION Y BUFF IR
00452  0 23 00011      397          EXU     EOTLOC+1            TERMINATE
00453  0 01 40451      398          BRU     *$-2
                        399  *                                                                  *
00454  0 00 00000      400  EOWYIR  PZE                         END-OF-WORD Y BUFF IR
00455  0 37 00026      401          STX     XYIR                SAVE X
```

```
00456  0 71 00016      402          LDX     NWDS+1
00457  0 23 00013      403          EXU     IOLOC+1          I/O EXECUTION
00460  0 41 00462      404          BRX     $+2
00461  0 23 00035      405          EXU     TOLOC+1          TERMINATION INST
00462  0 37 00016      406          STX     NWDS+1
00463  0 71 00026      407          LDX     XYIR             RESTORE X
00464  0 01 40454      408          BRU     *EOWYIR
                        409  *                                                              *
00465  0 00 00000      410  EOWWIR  PZE                      END-OF-WORD W BUFF IR
00466  0 37 01307      411          STX     XWIR             SAVE X
00467  0 71 00015      412          LDX     NWDS
00470  0 23 00012      413          EXU     IOLOC            I/O EXECUTION
00471  0 41 00473      414          BRX     $+2
00472  0 23 00034      415          EXU     TOLOC            TERMINATION INST
00473  0 37 00015      416          STX     NWDS
00474  0 71 01307      417          LDX     XWIR             RESTORE X
00475  0 01 40465      418          BRU     *EOWWIR
                        419  ********************************************************************
                        420
                        421          PAGE
                        422  ********************************************************************
                        423  *                                                                  *
                        424  *       BUFFER I/O POP                                      (BIOP)*
                        425  *            DESTROYS B REGISTER                                   *
                        426  *                                                                  *
                        427  ********************************************************************
     00000476           428  BIOPOP  EQU     $
00476  0 53 00042      429          SKN     IRCNT            IF NOT ENOUGH TIME
00477  0 01 00476      430          BRU     $-1                WAIT FOR NEXT INNING
00500  0 37 00025      431          STX     TMPX             /
00501  0 35 00045      432          STA     TMPA             /
00502  0 71 00000      433          LDX     0                /
00503  2 76 00001      434          LDA     1,2              /
00504  0 14 01103      435          ETR     YBUFF            /
00505  0 71 01075      436          LDX     P1               /
00506  0 73 01074      437          SKG     P0               /
00507  0 71 01074      438          LDX     P0               DETERMINE BUFFER
00510  2 76 01077      439          LDA     QBOTAD,2
00511  0 35 01157      440          STA     QADDR
00512  2 76 01056      441          LDA     BXADDR,2
00513  0 35 00051      442          STA     XADDR
00514  0 46 30003      443          CLR                      /
00515  0 76 00020      444          LDA     QNOW             /
00516  0 66 00011      445          RSH     9                /
00517  0 76 00000      446          LDA     0                /
00520  0 14 01064      447          ETR     MADR1            /
00521  0 36 00044      448          STB     TMP1             /
00522  0 16 00044      449          MRG     TMP1             FORM QNOW,POPREF TO PUT IN QUEUE
00523  0 71 00024      450          LDX     QSIZE            /
00524  0 75 01101      451          LDB     QMASK
00525  0 53 41157      452  B1      SKN     *QADDR           SEARCH BUFFER QUEUE
00526  0 01 00536      453          BRU     B2                FOR
00527  0 70 41157      454          SKM     *QADDR              LISTING
00530  0 41 00525      455          BRX     B1
00531  0 46 30003      456          CLR                      /
00532  0 76 00045      457          LDA     TMPA               RESTORE
00533  0 71 00025      458          LDX     TMPX                 AND
00534  0 36 00042      459          STB     IRCNT
00535  0 01 00476      460          BRU     BIOPOP                 WAIT
00536  0 35 41157      461  B2      STA     *QADDR           LIST REQUEST IN QUEUE
00537  0 71 00020      462          LDX     QNOW             /
00540  0 76 00025      463          LDA     TMPX             /
00541  0 35 40051      464          STA     *XADDR           SAVE X REG FOR CONBIR
00542  0 71 00000      465          LDX     0                /
00543  2 77 00004      466          EAX     4,2              /
00544  0 37 00000      467          STX     0                SET UP RETURN
00545  0 76 00045      468          LDA     TMPA             RESTORE
00546  0 71 00025      469          LDX     TMPX               AND
00547  0 51 00000      470          BRR     0                    RETURN
                        471  ********************************************************************
```

```
                        470         PAGE
                        471 ********************************************************************
                        472 *                                                                  *
                        473 *       SUBPROGRAM TO CONNECT BUFFER I/O ROUTINES                   *
                        474 *                                                                  *
                        475 ********************************************************************
00550  0 00 00000       476 CONBIR PZE
00551  0 37 00027       477         STX     XQSV                SAVE BUFFER
00552  2 76 01056       478         LDA     BXADDR,2
00553  0 35 00051       479         STA     XADDR
00554  2 76 40040       480         LDA     *QTBPAD,2           /
00555  0 14 01064       481         ETR     MADR1               /
00556  0 35 00043       482         STA     PBPREF              PBP REF CELL
00557  2 75 40040       483         LDB     *QTBPAD,2           /
00560  0 76 01067       484         LDA     N1                  /
00561  0 67 00011       485         LSH     9                   /
00562  0 35 00050       486         STA     QNBW1               COMPUTE QNBW OF PBPREF
00563  0 71 00027       487         LDX     XQSV                /
00564  0 43 00634       488         BRM     BUFADV              ACVANCE BUFFER
00565  0 71 00043       489         LDX     PBPREF              /
00566  0 46 30003       490         CLR                         /
00567  2 75 00001       491         LDB     1,2                 /
00570  0 36 00623       492         STB     CB1                 SET UP I/O INITIATION
00571  2 75 00004       493         LDB     4,2                 /
00572  0 71 00050       494         LDX     QNBW1               /
00573  0 71 40051       495         LDX     *XADDR              /
00574  0 54 40043       496         SUB     *PBPREF             /
00575  0 71 00027       497         LDX     XQSV                /
00576  2 35 00015       498         STA     NWDS,2              SET UP NUMBER OF WORDS TO OUTPUT
00577  2 36 00010       499         STB     EBTLBC,2             AND EBT INSTRUCTION
00600  0 71 00043       500         LDX     PBPREF              /
00601  2 75 00003       501         LDB     3,2                 /
00602  2 76 00002       502         LDA     2,2                 /
00603  0 14 01073       503         ETR     BPMSK               /
00604  0 16 01064       504         MRG     MB1                 /
00605  0 71 00050       505         LDX     QNBW1               /
00606  0 71 40051       506         LDX     *XADDR
00607  0 55 40043       507         ADD     *PBPREF             /
00610  0 35 00050       508         STA     IBTMP               /
00611  0 61 00043       509         MIN     PBPREF              /
00612  0 61 00043       510         MIN     PBPREF              /
00613  0 77 40043       511         EAX     *PBPREF             /
00614  0 37 00043       512         STX     PBPREF              /
00615  0 76 00043       513         LDA     PBPREF              /
00616  0 14 01063       514         ETR     MADR                /
00617  0 55 00050       515         ADD     IBTMP               SET UP I/O INSTR, EFFADR+NWDS,2
00620  0 71 00027       516         LDX     XQSV                /
00621  2 35 00012       517         STA     IBLBC,2             STORE I/O INSTR
00622  2 36 00034       518         STB     TBLBC,2              AND TERM I/O INSTR
00623  0 23 00000       519 CB1     EXU                         EXECUTE I/O INITIATION
00624  0 51 00550       520         BRR     CONBIR               AND RETURN
                        521 ********************************************************************
                        522 *                                                                  *
                        523 *       PBP TO HALT PROGRAM N WITHOUT HALTING OTHERS                (WAIT)*
                        524 *             DESTROYS (B), (X)                                    *
                        525 ********************************************************************
00000625                526 WAITBP EQU     $
00625  0 71 00020       527         LDX     QNBW
00626  0 75 01074       528         LDB     PC
00627  2 36 01144       529         STB     HLTFLG,2
00630  0 36 00042       530         STB     IRCNT               SKIP REST INNING
00631  2 53 01144       531         SKN     HLTFLG,2            WAIT UNTIL HALT FLAG IS MADE NEG.
00632  0 01 00631       532         BRU     $-1
00633  0 51 00000       533         BRR     0
                        534 ********************************************************************
                        535         PAGE
                        536 ********************************************************************
                        537 *                                                                  *
                        538 *       PROGRAM TO ADVANCE BUFFER REQUEST QUEUE                     (BUFADV)*
                        539 *                                                                  *
                        540 ********************************************************************
```

```
00634  0 00 00000      543  BUFADV PZE
00635  2 71 00040      544         LDX     QTBPAD,2
00636  0 76 00024      545         LDA     QSIZE
00637  2 75 00001      546  BUF1   LDB     1,2
00640  2 36 00000      547         STB     0,2
00641  0 41 00642      548         BRX     $+1
00642  0 55 01075      549         ADD     P1
00643  0 72 01065      550         SKA     MBC
00644  ' 01 00637      551         BRU     BUF1
00645  0 76 01074      552         LDA     P0
00646  2 35 00000      553         STA     0,2
00647  0 51 00634      554         BRR     BUFADV
                        555  ********************************************************************************
                        556  *                                                                              *
                        557  *       PROGRAM TO LIST ERROR TEXT                                    (ERRTYP)*
                        558  *                                                                              *
                        559  ********************************************************************************
00650  0 00 00000      560  ERRTYP PZE
00651  0 61 00650      561         MIN     ERRTYP
00652  0 75 40650      562         LDB     *ERRTYP
00653  0 36 00663      563         STB     ERRTXT+1
00654  1 76 01074      564         BIO     P2
00655  0 02 02641      565         TYP     0,1,4
00656  0 12 00662      566         MIW     ERRTXT
00657  0 02 14000      567         TBP,
00660  0 20 00000      568  NOP    NOP
00661  0 51 00650      569         BRR     ERRTYP
                        570  *                                                                              *
00662  52255151        571  ERRTXT TEXT    < ERR>
00663  0 00 00000      572         PZE
                        573  ********************************************************************************
                        574         PAGE
                        575  ********************************************************************************
                        576  *                                                                              *
                        577  *       PROGRAM TO FORM QUEUE EXECUTION ORDER                         (QFORM)*
                        578  *                                                                              *
                        579  ********************************************************************************
00664  0 00 00000      580  QFORM  PZE
00665  0 61 00664      581         MIN     QFORM                   /
00666  0 75 40664      582         LDB     *QFORM
00667  0 71 00024      583         LDX     QSIZE                   /
00670  0 53 00042      584         SKN     IRCNT                   /
00671  0 01 00670      585         BRU     $-1                     IF NOT TIME, SKIP REST OF INNING
00672  2 36 01274      586         STB     QP,2                    /
00673  2 36 01246      587         STB     QLINK,2
       00000674        588  QFM1B  EQU     $
00674  0 53 00042      589         SKN     IRCNT                   /
00675  0 01 00674      590         BRU     $-1                     IF NOT TIME, SKIP REST OF INNING
00676  0 46 00003      591         CLR
00677  2 76 41246      592  QFM1A  LDA     *QLINK,2                /
00700  0 14 01063      593         ETR     MADR                    /
00701  2 35 01307      594         STA     QTIME,2                 /
00702  0 76 01074      595         LDA     P0                      /
00703  2 54 01307      596         SUB     QTIME,2                 /
00704  2 35 01307      597         STA     QTIME,2                 /
00705  0 46 10012      598         BAC                             /
00706  2 55 01307      599         ADD     QTIME,2                 /
00707  0 46 20005      600         ABC                             /
00710  0 41 00677      601         BRX     QFM1A                   /
00711  0 46 00014      602         XAB                             /SAVE QUEUE TIME ALLOCATIONS
00712  0 35 00021      603         STA     QNUM                    /COMPUTE NEW NUMBER OF QUEUE ITEMS
00713  0 35 00022      604         STA     QNMDMY
00714  0 54 01072      605         SUB     NQMAX                   /
00715  0 73 01074      606         SKG     P0                      /
00716  0 01 00742      607         BRU     Q2BIG
       00000717        608  QFM0   EQU     $
00717  0 76 01062      609         LDA     LSTLOC                  /
00720  0 55 00021      610         ADD     QNUM                    /
00721  0 35 00023      611         STA     QPBINT                  /INITIALIZE POINTER TO TOP OF LIST
00722  0 71 00024      612  QFM1   LDX     QSIZE                   /
00723  2 76 01307      613  QFM2   LDA     QTIME,2                 /
```

```
00724  0 72 01067     614          SKA     N1                  /
00725  0 01 00727     615          BRU     $+2                 /
00726  0 01 00735     616          BRU     QFM3                /
00727  0 37 40023     617          STX     *QPRINT             /
00730  2 61 01307     618          MIN     QTIME,2             /
00731  0 61 00023     619          MIN     QPRINT              /
00732  0 61 00022     620          MIN     QNMDMY              /
00733  0 53 00022     621          SKN     QNMDMY              /
00734  0 01 00737     622          BRU     $+3
00735  0 41 00723     623  QFM3    BRX     QFM2                /
00736  0 01 00722     624          BRU     QFM1                /FARM QUEUE LIST
00737  0 40 20004     625          IFT
00740  0 01 01016     626          BRU     RESTRT              IF IR NOT ENABLED, THEN RESTART
00741  0 01 01025     627          BRU     REST1               /
                      628  *
00742  0 76 01072     629  Q2BIG   LDA     NQMAX
00743  0 35 00021     630          STA     QNUM                /
00744  0 35 00022     631          STA     QNMDMY              /SET QUEUE SIZE MAXIMUM
00745  0 43 00650     632          BRM     ERRTYP              /
00746  45472752       633          TEXT    <NPG >
00747  0 71 00024     634          LDX     QSIZE               /
00750  2 75 01246     635          LDB     QLINK,2             /
00751  2 36 01274     636          STB     QP,2                /RESTART ADDRESS IN QUEUE
00752  0 01 00717     637          BRU     QFM0
                      638  *                                                              *
                      639  ********************************************************************
                      640          PAGE
                      641  ********************************************************************
                      642  *                                                                  *
                      643  *       PROGRAM TO REMOVE LAST PROGRAM FROM QUEUE          (QUIT)*
                      644  *                                                                  *
                      645  ********************************************************************
     00000753         646  QUIT    EQU     $
00753  0 43 06742     647          BRM     PURGE               CLEAR LOADER SYMBOL TABLE
00754  0 76 00014     648          LDA     LBIAS               /
00755  0 35 07646     649          STA     BIAS                /RESTORE BIAS TO LAST LOCATION
00756  0 76 00024     650          LDA     QSIZE               /
00757  0 67 00017     651          LSH     15                  /
     00000760         652  QU1     EQU     $
00760  1 70 00000     653          QRLS
00761  0 75 01101     654          LDB     QMASK               /
00762  0 71 00024     655          LDX     QSIZE
00763  2 70 01220     656  QU2     SKM     QBBTW,2             /
00764  0 01 00766     657          BRU     $+2                 /
00765  0 01 00760     658          BRU     QU1                 /
00766  2 70 01233     659          SKM     QBBTY,2             /
00767  0 01 00771     660          BRU     $+2                 /
00770  0 01 00760     661          BRU     QU1                 /
00771  0 41 00763     662          BRX     QU2                 WAIT UNTIL I/O IS COMPLETE
00772  0 61 00024     663          MIN     QSIZE               REDUCE NUMBER OF PROGS
00773  0 53 00024     664          SKN     QSIZE
00774  0 01 01000     665          BRU     QSTART
00775  0 43 01040     666          BRM     BUFSIZ
00776  0 71 00024     667          LDX     QSIZE
00777  0 01 00674     668          BRU     QFM18
                      669  ********************************************************************
                      670          PAGE
                      671  ********************************************************************
                      672  *                                                                  *
                      673  *       ENTRY TO SET UP QUEUE SWITCH                               *
                      674  *                                                                  *
                      675  ********************************************************************
     00001000         676  QSTART  EQU     $
01000  0 76 01051     677          LDA     ARMQ                /
01001  0 35 00047     678          STA     ARMQ1               RESET ALL IR*S
01002  0 46 30003     679          CLR                         /
01003  0 35 00021     680          STA     QNUM                /
01004  0 35 00024     681          STA     QSIZE               /
01005  0 35 01220     682          STA     QBBTW
01006  0 35 01233     683          STA     QBBTY               CLEAR BUFFER QUEUES
01007  0 76 01336     684          LDA     MONEND              /
01010  0 35 07646     685          STA     BIAS                /SET INITIAL LOAD ADDRESS
```

```
                00001011    686  QST1   EQU    $
01011  0 43 01337   687                BRM    PATCH                MODIFY LOADER
01012  0 43 06742   688                BRM    PURGE                CLEAR LOADER SYMBOL TABLE
01013  0 76 00660   689                LDA    NOP
01014  0 35 01011   690                STA    QST1
01015  0 01 01016   691                BRU    RESTRT
                    692  ********************************************************************************
                    693                PAGE
                    694  ********************************************************************************
                    695  *                                                                              *
                    696  *       ENTRY TO RESTART PROGRAMS                                    (RESTRT)*
                    697  *                                                                              *
                    698  ********************************************************************************
01016  0 71 01072   699  RESTRT LDX    NQMAX                /
01017  0 46 30003   700                CLR                  /
01020  2 76 01246   701  RS1    LDA    QLINK,2              /
01021  2 35 01274   702                STA    QP,2           /
01022  2 37 01144   703                STX    HLTFLG,2
01023  2 36 01335   704                STB    TSTFLG,2
01024  0 41 01020   705                BRX    RS1            /
                    706  ********************************************************************************
                    707  *                                                                              *
                    708  *       ENTRY AFTER PROGRAM LOAD                                               *
                    709  *                                                                              *
                    710  ********************************************************************************
                00001025    711  REST1  EQU    *
01025  0 43 01040   712                BRM    BUFSIZ
01026  0 46 30003   713                CLR
01027  0 35 01261   714                STA    QLIST
01030  0 35 00017   715                STA    QITEM
01031  0 35 00020   716                STA    QNOW
01032  0 35 01144   717                STA    HLTFLG           SET UP SWITCH HANGUP UNTIL FILL
01033  0 35 00042   718                STA    IRCNT            ENABLE QUEUE SWITCH
01034  0 02 20002   719                EIR
01035  0 02 20020   720                AIR
01036  0 13 00047   721                PRT    ARMQ1            ENABLE PROG IR*S, QUEUE
01037  0 01 01037   722                BRU    *                WAIT FOR QUEUE TO BEGIN
                    723  ********************************************************************************
                    724                PAGE
                    725  ********************************************************************************
                    726  *                                                                              *
                    727  *       PROGRAM TO SET BUFFER QUEUE SIZE                                       *
                    728  *                                                                              *
                    729  ********************************************************************************
01040  0 00 00000   730  BUFSIZ PZE
01041  0 71 01070   731                LDX    N2               /
01042  2 76 01101   732  QF1    LDA    QBGTAD+2,2           /
01043  0 55 00024   733                ADD    QSIZE            /
01044  0 14 01063   734                ETR    MADR             /
01045  2 35 00042   735                STA    QTGPAD+2,2       /
01046  0 41 01042   736                BRX    QF1              /INCREMENT BUFFER QUEUE SIZE
01047  0 51 01040   737                BRR    BUFSIZ
                    738  ********************************************************************************
                    739                PAGE
                    740  ********************************************************************************
                    741  *                                                                              *
                    742  *       DATA TABLE                                                             *
                    743  *                                                                              *
                    744  ********************************************************************************
01050  00100000     745  ARMO   DATA   0100000
01051  00601400     746  ARMQ   DATA   0600000+0140000*/(0201-LASTIR)
                00001051    747  BPTST  EQU    $-1
01052  44020400     748                DATA   044020400        BPT 1
01053  44020200     749                DATA   044020200        BPT 2
01054  44020100     750                DATA   044020100        BPT 3
01055  44020040     751                DATA   044020040        BPT 4
01056  2 00 01116   752  BXADDR PZE    BIOSV0,2             LINKS TO
01057  2 00 01131   753                PZE    BIOSV1,2
01060  77677777     754  DARMO  DATA   077677777
01061  77777766     755  IRMAX  DATA   -IRRATE               NUMBER OF IR*S PER INNING
01062  0 00 01261   756  LSTLOC PZE    QLIST
```

```
01063  00037777        757  MADR    DATA    037777              ADDRESS MASK
01064  00077777        758  MADR1   DATA    077777              ADDRESS MASK WITH *BIT
01065  40000000        759  MB0     DATA    040000000           SIGN BIT
01066  20000000        760  MB1     DATA    020000000           INDEX TAG BIT
01067  77777777        761  M1      DATA    -1
01070  77777776        762  M2      DATA    -2
01071  77776030        763  M1000   DATA    -1000
01072  77777766        764  NQMAX   DATA    -QMAX
01073  07700000        765  OPMSK   DATA    07700000            OP CODE MASK
01074  00000000        766  P0      DATA    0
01075  00000001        767  P1      DATA    +1
01076  00000002        768  P2      DATA    +2
01077  2 00 01220      769  QBUFAD  PZE     QBUFW,,2
01100  2 00 01233      770          PZE     QBUFY,,2
01101  77700000        771  QMASK   DATA    077700000
01102  0 00 06576      772  QUBLAD  PZE     QUBLDR              LOADER ORIGIN
01103  00000100        773  YBUFF   DATA    0100                Y BUFFER BIT, 0=W ,1=Y
                       774  ********************************************************************
                       775          PAGE
                       776  ********************************************************************
                       777  *                                                                  *
                       778  *       STORAGE TABLE                                              *
                       779  *                                                                  *
                       780  ********************************************************************
01104                  781          RES     QMAX
01116                  782  BIASV0  RES     QMAX+1              BIO X REG SAVE, W BUFF
01131                  783  BIASV1  RES     QMAX+1              BIO X REG SAVE, Y BUFF
01144                  784  $HLTFLG RES     QMAX+1              HALT/IDLE FLAGS
01157                  785  QO      RES     QMAX+1              SAVE O
01172                  786  QA      RES     QMAX+1              SAVE A REGISTER
01205                  787  QB      RES     QMAX+1              SAVE B REGISTER
01220                  788  QBUFW   RES     QMAX+1              W BUFFER REQUEST QUEUE
01233                  789  QBUFY   RES     QMAX+1              Y BUFFER REQUEST QUEUE
01246                  790  QLINK   RES     QMAX+1              LINKS TO PROGRAM STARTS
01261                  791  QLIST   RES     QMAX+1              QUEUE ORDER OF EXECUTION LIST
01274                  792  QP      RES     QMAX+1              SAVE P REGISTER
01307                  793  QTIME   RES     QMAX+1              SAVE QUEUE TIME ALLOCATION
01322                  794  QX      RES     QMAX+1              SAVE X REGISTER
01335  77777777        795  TSTFLG  DATA    -1                  READY-TEST FLAGS
       00000047        796  ARM01   EQU     047
       00001205        797  ASAVE   EQU     QB                  POP REF SAVE CELL
       00001274        798  BUFF    EQU     QP                  BIO BUFFER SAVE CELL
       00000010        799  EOTLOC  EQU     010                 E-O-T INST LOC, W AND Y BUFFS
       00000012        800  IOLOC   EQU     012                 I/O INSTRUCTION
       00000050        801  IOTMP   EQU     050                 I/O INSTR TMP STORAGE
       00000042        802  IRCNT   EQU     042                 NO OF IR*S YET THIS INNING
       00000014        803  LBIAS   EQU     014                 LAST BIAS VALUE
       00000015        804  NWDS    EQU     015                 #-NO WDS TO I/O, W AND Y BUFFS
       00000043        805  POPREF  EQU     043                 CONBIR TMP STORAGE
       00001157        806  QADDR   EQU     QO                  BIO BUFFER PRIORITY SET-UP
       00001172        807  QFA     EQU     QA                  QFILL A SAVE
       00001322        808  QFX     EQU     QX                  QFILL X SAVE
       00000017        809  QITEM   EQU     017                 POINTER TO ITEM IN QUEUE PROCESS LIST
       00000020        810  $QNOW   EQU     020                 PROG NOW IN PROCESS
       00000050        811  QNOW1   EQU     IOTMP               BIO QNOW SAVE
       00000021        812  QNUM    EQU     021                 LIST SIZE
       00000022        813  QNMDMY  EQU     022                 DUMMY LIST SIZE
       00000023        814  QPOINT  EQU     023                 POINTER FOR LIST SET-UP
       00000024        815  QSIZE   EQU     024                 NO PROGRAMS IN QUEUE
       00000040        816  QTOPAD  EQU     040                 BUFF QUEUE TOP ADDRESS
       00000045        817  TMPA    EQU     045                 POP TMP STORAGE
       00000046        818  TMPB    EQU     046                 POP TMP STORAGE
       00000025        819  TMPX    EQU     025                 POP TMP STORAGE
       00000044        820  TMP1    EQU     044                 BIO TEMP STORAGE
       00001274        821  TMP2    EQU     QP                  LOADER MOD TEMP STO
       00000034        822  TOLOC   EQU     034                 I/O INSTRUCTIONS, W,Y BUFF
       00000051        823  XADDR   EQU     051                 X SAVE LINK IN BIO
       00000027        824  XQSW    EQU     027                 QSWITC SAVE CELL
       00001307        825  XWIR    EQU     QTIME               FOXWIR  SAVEX
       00000026        826  XYIR    EQU     026                 FOXYIR  SAVE X
01336  0 00 01337      827  MONEND  PZE     *+1
                       828  ********************************************************************
```

```
                        829          PAGE
                        830  ********************************************************************
                        831  *                                                                  *
                        832  *       DEFINITION OF LOADER PATCH LOCATIONS                       *
                        833  *                                                                  *
                        834  ********************************************************************
        00007674        835  CARD1   EQU     LDZERO+03674
        00006734        836  CARRTN  EQU     LDZERO+02734
        00006665        837  CHKERR  EQU     LDZERO+02665
        00007727        838  CN      EQU     LDZERO+03727
        00006606        839  FLAG    EQU     LDZERO+02606
        00007720        840  FLG     EQU     LDZERO+03720
        00007722        841  MES1    EQU     LDZERO+03722
        00007677        842  MESP    EQU     LDZERO+03677
        00007633        843  MESPT   EQU     LDZERO+03633
        00006657        844  MULDEF  EQU     LDZERO+02657
        00006720        845  P40     EQU     LDZERO+02720        *+40
        00006742        846  PURGE   EQU     LDZERO+02742
        00006662        847  RECFMT  EQU     LDZERO+02662
        00006670        848  SKIP    EQU     LDZERO+02670
        00007721        849  SKPFLG  EQU     LDZERO+03721
        00006731        850  SQFERR  EQU     LDZERO+02731
        00000006        851  TABSIZ  EQU     6
        00007632        852  TMP1L   EQU     LDZERO+03632
        00006721        853  THRU1   EQU     LDZERO+02721
        00006725        854  THRU2   EQU     LDZERO+02725
                        855  ********************************************************************
                        856  *                                                                  *
                        857  *       PROGRAM TO MAKE LOADER WORK ON INTERRUPTS                  *
                        858  *                                                                  *
                        859  ********************************************************************
01337  0 00 00000       860  PATCH   PZE
01340  0 71 01074       861          LDX     PO
01341  0 77 01363       862          EAX     LDRPCH              /
01342  0 37 00023       863          STX     QPOINT              /POINTER TO TOP OF LIST
01343  0 76 40023       864  PCH1    LDA     *QPOINT             /
01344  0 14 01063       865          ETR     MASK                /
01345  0 35 00044       866          STA     TMP1                STARTCELL
01346  0 76 40023       867          LDA     *QPOINT             /
01347  0 66 00017       868          RSH     15
01350  0 35 01274       869          STA     TMP2                NO WORDS THIS RECORD
01351  0 53 01274       870          SKN     TMP2
01352  0 51 01337       871          BRR     PATCH               FINISHED MODS
01353  0 71 01274       872          LDX     TMP2
01354  0 61 00023       873  PCH2    MIN     QPOINT
01355  0 76 40023       874          LDA     *QPOINT
01356  0 35 40044       875          STA     *TMP1               /
01357  0 61 00044       876          MIN     TMP1                /
01360  0 41 01354       877          BRX     PCH2                /PICK UP WORD, STORE
01361  0 61 00023       878          MIN     QPOINT
01362  0 01 01343       879          BRU     PCH1                /GO TO NEXT RECORD
                        880  ********************************************************************
                        881          PAGE
                        882  ********************************************************************
                        883  *                                                                  *
                        884  *       LOADER PATCHES                                             *
                        885  *                                                                  *
                        886  ********************************************************************
                        887  *                                                                  *
                        888  RECORD FORM    9,15                 *NOWORDS, STARTLOC
                        889  *                                                                  *
                        890  ********************************************************************
01363  774 06700       891  LDRPCH RECORD  *2,LDZERO+02700
01364  0 20 00000       892          NOP                         /
01365  0 43 00664       893          BRM     QFORM               /LINK TO PROGRAM QUEUE
                        894  *                                                                  *
01366  777 06600       895          RECORD  *1,LDZERO+02600
01367  0 01 06614       896          BRU     LDZERO+02614
                        897  *                                                                  *
01370  76' 06603       898          RECORD  *9,LDZERO+02603
01371  0 43 06742       899          BRM     PURGE
```

```
01372  0 01 06614      900          BRU     LDZERR+02614
01373                  901          RES     1
01374                  902          RES     1                  FLAG
01375                  903          RES     5
                       904  *
01402  777 06635       905          RECORD  *1,LDZERR+02635
01403  0 20 00000      906          NOP
                       907  *
01404  765 06656       908          RECORD  *11,LDZERR+02656
01405  0 01 00753      909          BRU     QUIT
01406  0 43 00650      910          BRM     ERRTYP
01407  44242652        911          TEXT    <MDF >             MULTIPLE DEFS
01410  0 01 00753      912          BRU     QUIT
01411  0 43 00650      913          BRM     ERRTYP
01412  26446352        914          TEXT    <FMT >             ILLEGAL FORMAT
01413  0 01 00753      915          BRU     QUIT
01414  0 43 00650      916          BRM     ERRTYP
01415  23624452        917          TEXT    <CSM >             CHECKSUM ERROR
01416  0 01 00753      918          BRU     QUIT
01417  0 53 06606      919          SKN     FLAG
                       920  *
01420  777 06674       921          RECORD  *1,LDZERR+02674
01421  0 01 06657      922          BRU     MULDEF             MULTIPLE DEFS
                       923  *
01422  731 06703       924          RECORD  *39,LDZERR+02703
01423  0 00 00000      925          PZE                        READ
01424  0 76 01067      926          LDA     N1                 /
01425  0 35 06606      927          STA     FLAG               SET FINISH FLAG
01426  1 76 06720      928          BIR     P40                /
01427  0 02 02604      929          RPT     0,1,4              /
01430  0 32 07727      930          WIM     CW                 /
01431  0 61 06606      931          MIN     FLAG               /
01432  0 61 06606      932          MIN     FLAG               /
01433  1 73 06670      933          BUSY    SKIP               /
01434  0 40 20010      934          BETW                       /
01435  0 01 07674      935          BRU     CARD1              BUFFER ERROR TEST
01436  0 01 06774      936          BRU     LDZERR+02776
01437                  937          RES     1                  /
01440  00000050        938          DATA    40                 P40
01441                  939          RES     8                  /
01451  0 43 00650      940          BRM     ERRTYP             SOFERR, SYMBOL TABLE OVERFLOW
01452  46264352        941          TEXT    <OFL >             /
01453  0 01 00753      942          BRU     QUIT               /
01454  1 76 01075      943          BIR     P1
01455  0 02 02641      944          TYP     0,1,4
01456  0 12 07722      945          MIW     MES1
01457  0 02 14000      946          TAPW
01460  0 20 00000      947          NOP
01461  0 51 07677      948          BRR     MESP
01462  0 00 00000      949          PZE                        PURGE LOADER SYMBOL TABLE
01463  0 71 07612      950          LDX     LDZERR+03612       /
01464  0 75 01074      951          LDB     P0
01465  2 36 07663      952          STB     LDZERR+03663,2     /
01466  0 41 06745      953          BRX     LDZERR+02745
01467  0 75 07625      954          LDB     LDZERR+03625       /
01470  0 36 07656      955          STB     LDZERR+03656       /
01471  0 51 06742      956          BRR     PURGE              RETURN
                       957  *
01472  777 07006       958          RECORD  *1,LDZERR+03006
01473  0 01 06662      959          BRU     RECFMT             ILLEGAL RECORD FORMAT
                       960  *
01474  777 07035       961          RECORD  *1,LDZERR+03035
01475  0 01 06665      962          BRU     CHKERR             CHECKSUM ERROR
                       963  *
01476  776 07067       964          RECORD  *2,LDZERR+03067
01477  1 74 01053      965          READY   BPTST+2
01500  0 20 00000      966          NOP
                       967  *
```

```
01501  777 07436      968          RECORD  =1,LDZERR+03436
01502  0 01 06731     969          BRU     SBFERR          SYMBOL TABLE OVERFLOW
                      970  *                                                      *
01503  777 07625      971          RECORD  =1,LDZERR+03625
01504  0 00 04565     972          PZE     LDZERR+02573-TABSIZ
                      973  *                                                      *
01505  775 07674      974          RECORD  =3,CARD1
01506  0 43 00650     975          BRM     ERRTYP          /
01507  63214752       976          TEXT    <TAP >          /
01510  0 01 00753     977          BRU     QUIT            TYPE *ERRTAP*, QUIT
                      978  *                                                      *
01511  763 07705      979          RECORD  =13,LDZERR+03705  /
01512  0 35 07632     980          STA     TMP1L           NUMBER OF WORDS TO TYPE
01513  0 76 01067     981          LDA     N1
01514  0 35 07720     982          STA     FLG             SET UP EXIT FLAG
01515  1 76 07632     983          BIR     TMP1L           /
01516  0 02 02641     984          TYP     0,1,4           /
01517  0 12 47633     985          MIW     *MESPT          /
01520  0 02 14000     986          TAPW                    /
01521  0 61 07720     987          MIN     FLG
01522  1 73 07721     988          BUSY    SKPFLG
01523  0 01 06734     989          BRU     CARRTN
01524                 990          RES     1
01525                 991          RES     1               FLG
01526  0 53 07720     992          SKN     FLG             SKPFLG
                      993  *                                                      *
01527  0 00 00000     994          PZE
                      995  ************************************************************
       00001000       996          END     QSTART
```

SDS SCIENTIFIC DATA SYSTEMS

SDS PROGRAM LIBRARY
PROGRAM DESCRIPTION

 Catalog No. 000020

IDENTIFICATION: UNIVERSAL LOADER

AUTHOR: Programmatics Incorporated/R. Tausworthe (changes)

ACCEPTED: February 3, 1965/July 18, 1968 (changes)

COMPUTER CONFIGURATION: Any SDS 900 series computer with a ~~card reader and/or~~ photo reader and a typewriter. Loader ~~exists on cards and paper tape and~~ loads programs which exist ~~either~~ on ~~cards or~~ paper tape.

PURPOSE: To load one or more programs produced by SYMBOL or META-SYMBOL and presented to the loader on ~~either punched cards or~~ paper tape. This loader has essentially the same capabilities as the SDS MONARCH Loader but it functions independently of MONARCH.

PROGRAMMED OPERATIONS: None

SUBROUTINES REQUIRED: None

STORAGE: Uses last ~~01204~~ 1212 locations of memory plus storage for the loader's symbol table which is constructed in the locations immediately preceding the loader with each unique symbol (external label or POP item) requiring 3 words of memory.

~~This loader uses locations 0 through 0132_8 temporarily while it is being loaded into upper memory. However, once the loader commences to load a user's program, these cells are available and the user's program can be loaded over these locations.~~

TIMING: ~~N/A~~ Non-reentrant

ASSEMBLY LANGUAGE USED: SYMBOL

LOADING PROCEDURE:

~~Loading of the Universal Loader is a three step process involving a pre-bootstrap, a bootstrap loader, and the Universal Loader. The pre-bootstrap (first card of the deck or first record of the paper tape) bootstraps itself into memory, computes memory size, stores the initial relocation bias in the next to last location in memory and simulates a card or paper tape FILL.~~

Loaded as part of QMONITOR by standard paper tape fill.

~~When loading the loader from cards, the pre-bootstrap simulates a card FILL to bring in the Two Card Loader (Catalog No. 030007) which then loads the Universal Loader. When loading the loader from paper tape the pre-bootstrap simulates a paper tape FILL to bring in the Basic Paper Tape Loader* (Catalog No. 000019) which then loads the Universal Loader.~~

~~Listings of pre-bootstrap for card and paper tape are included with listing of Universal Loader.~~

~~The Two Card Loader is included in the binary deck and is the second and third cards of the deck. Also, the Basic Paper Tape Loader is included on the paper tape version and is the second record of the tape.~~

~~PAPER TAPE~~

~~To load the Universal Loader from paper tape:~~

~~1. Insert loader in photoreader.~~

~~2. Set (A) = initial relocation bias (ignored when loading absolute object programs).**~~

~~*The halt in the Basic Paper Tape Loader has been removed.~~

~~**i. e., its value remains unchanged during loading of absolute object programs.~~

3. Set breakpoint switches as described below under "Options".

4. Execute paper tape PHLL.

CARD

To load Universal Loader from cards:

1. Insert loader in card reader.

2. Set (A) = initial relocation bias (ignored when loading absolute object programs).*

3. Set breakpoint switches as described below under "Options".

4. Execute (or simulate) card PHLL.

To simulate card PHLL on SDS 910/920:

a. Set location (1) = WIM 2 (022 00002)

b. Press START

c. Set (A) = initial relocation bias

d. Set (C) = EOM 03606 (002 03606)

e. Press RUN

*i.e., its value remains unchanged during loading of absolute object programs.

USE:

Options

~~Several~~ loader options are controlled by the setting of the four Breakpoint switches.

Breakpoint Switch	Setting	Function
~~1~~	~~down~~	~~a) The loader will halt before execution to allow insertion of programs to be loaded in the device selected by Breakpoint 4.~~ ~~b) The A register can be set equal to the initial load relocation bias. If the A register is not modified, the initial contents of A are used.~~
~~1~~	~~up~~	~~Loader commences loading from device indicated by Breakpoint 4 as soon as loader itself has been loaded. The initial contents of the A register (bits 10-23) are used as the initial load relocation bias.~~
2	down	Loader will halt when an End Record without transfer address is encountered. ~~The A register contains the load relocation bias, as computed by the loader for the next program to be loaded.~~
2	up	Loader will not halt when End Record without a transfer address is encountered.

3	down	Loader will halt when End Record with transfer address is encountered. The A register contains the load relocation bias, as computed by the loader, for the next program to be loaded. The G register contains the unconditional branch (BRU) to the transfer address.
3	up	Loader will not halt when End Record with transfer address is encountered. An unconditional branch to the transfer address is executed with (A)$_{10-23}$ = load relocation bias, as computed by the loader, for the next program to be loaded.
4	down	Input device is card reader 1 on the W buffer.
4	up	Input device is paper tape reader 1 on the W buffer.

Use as a Subroutine

Once the loader has been loaded into memory, it can be used by programs in memory as a loader subroutine to load other programs. The calling sequence is:

BRM QUBLDR

where $(A)_{10-23}$ is interpreted as the initial load relocation bias and:

$(A)_0 = 1 \rightarrow$ loader's symbol table is retained

$(A)_0 = 0 \rightarrow$ loader's symbol table is purged

and where QUBLDR is an external label which is always present in the loader's symbol table (i.e., never purged).

Cannot be used to return control to calling program.

The loader returns control to the program ~~which called it only if an End Record with a~~ transfer address ~~is encountered and the transfer address specifies an address in the program which called the loader~~ (i.e., a "BRR" return to the program is not made).

Halts

There are several halt instructions within the loader. Some of these halts are termed "scheduled" halts since they occur only when specifically requested by the user. The remaining halts are "unscheduled" or "error" halts.

(a) Scheduled Halts

C Register Contents	Indicates
~~0 20 11111~~	~~Breakpoint 1 down when control is transferred to loader~~.
~~0 20 22222~~	Breakpoint 2 is down when End Record without transfer address is encountered.
~~0 01 aaaaa~~	~~Breakpoint 3 is down when End Record with transfer address is encountered. "aaaaa" is tranfer address.~~

(b) Unscheduled or "Error" Halts

C Register Contents	Indication and Corrective Action, If Any
~~0 20 00001~~	Buffer error on attempt to read a record from input medium ~~specified by Breakpoint 4. Re~~ position offending record for re-read, then IDLE RUN.
(paper tape reader)	

(c) Blank COMMON references and a definition. Blank COMMON references should be preceded by a Blank COMMON definition, but external references and definitions (label or POP) need not be supplied in any particular order.

The term "program" is used in this description of the Universal Loader to mean a sequence of:

(a) One or more data records (Record Type 0).

and/or (b) One or more external reference or definition records (Record Type 1).

and/or (c) One or more programmed operator reference or definition records (Record Type 2).

and (d) Terminated by an end record (Record Type 3) with or without a transfer address.

The last, or only program, in a series of programs to be loaded should have an end record (Type 3) with a transfer address and all of the programs preceding it should have end records (Type 3) without transfer addresses.

If there are unsatisfied label or POP references at the time the end record with a transfer address is encountered, the loader indicates that an error condition exists. ~~The programmer can then supply a subroutine or POP library as additional input to the loader. The loader will search the library for subroutines which contain external label or POP definitions which satisfy any of the unsatisfied external references. The loader concludes searching and/or loading from the library when all references are satisfied or an end-of-file (17170000) is encountered, i.e. when a card with 1717 punched* in column one is encountered, or in the case of paper tape when a one word record consisting of 17170000 is encountered.~~

~~*Produced by multi-punching 0-1-2-3-6-7-8-9 in column one.~~

See Appendix J of "SYMBOL and META-SYMBOL REFERENCE MANUAL", SDS publication number 900506A, July 1964 (and revisions) for a description of the data formats accepted by this loader. Note, however, that the Universal Loader does not accept labeled common definitions or references (Record Type 2, Item Types 1 and 3) and will treat labeled common references as format errors.

Programs to be loaded can be introduced on ~~punched cards or~~ paper tape. The input/output subroutines within the Universal Loader ~~do not~~ use interrupts and do not use interlace.

```
                                1 *              UNIVERSAL LOADER  - SDS 900 SERIES COMPUTERS
        00003776                2 TBIAS   EQU     $+03776
02573                           3         ORG     02573
        00000000                4 S9300   EQU     0
        00000000                5 X0      EQU     S9300
        00000002                6 X2      EQU     2-S9300
                                7 MDE     OPD     06000000
        00020400                8 BP10FF  EQU     020400
        00020200                9 BP20FF  EQU     020200
        00020100               10 BP30FF  EQU     020100
        00020040               11 BP40FF  EQU     020040
02573   50642243               12 ITBL    DATA    *QUBL*
02574   24516060               13         DATA    *DR  *
02575   4 00 0 02576           14         PZE     QUBLDR,4
02576   0 20 0 00000           15 QUBLDR  NOP
02577   0 72 0 03604           16         SKA     MBO
02600   0 01 06614             17         BRU     LDZERO+02614
02601   0 01 0 02603           18         BRU     LDR1            PURGE LOADER'S SYMBOL TABLE
02602   0 76 0 03776           19 LOADER  LDA     TBIAS
02603   0 43 06742             20 LDR1    BRM     PURGE
02604   0 01 06614             21         BRJ     LDZERO+02614
02605                          22         RES     1
02606                          23 FLAG    RES     1
02607                          24         RES     5
02610   0 36 0 03656           25         STD     ETBL
02611   0 40 0 20400           26 LDR2    SKS     BP10FF
02612   0 00 0 11111           27         HLT     011111          *WAIT- BREAKPOINT 1 ON (DOWN)
02613   0 20 0 11111           28         NOP     011111
02614   0 14 0 03621           29         ETR     MADR
02615   0 35 0 03646           30         STA     BIAS
02616   0 46 30003             31         CLR
02617   0 35 0 03650           32         STA     MLOC
        00002620               33 STRT    EQU     $
02620   0 43 0 02703           34 LOOP    BRM     READ            READ NEXT RECORD, AND VERIFY
02621   0 43 0 03047           35         BRM     LOAD            PROCESS VERIFIED RECORD
02622   0 76 0 03645           36         LDA     T               LOAD TYPE CODE OF RECORD JUST PROCESS
02623   0 75 0 03605           37         LDB     N1
02624   0 70 0 03567           38         SKM     P3
```

```
                                    1 *              UNIVERSAL LOADER  - SDS 900 SERIES COMPUTERS
          00003776                  2 TBIAS   EQU     $+03776
02573                               3         ORG     02573
          00000000                  4 S9300   EQU     0
          00000000                  5 X0      EQU     S9300
          00000002                  6 X2      EQU     2-S9300
                                    7 MDE     OPD     06000000
          00020400                  8 BP1OFF  EQU     020400
          00020200                  9 BP2OFF  EQU     020200
          00020100                 10 BP3OFF  EQU     020100
          00020040                 11 BP4OFF  EQU     020040
02573     50642243                 12 ITBL    DATA    *QUBL*
02574     24516060                 13         DATA    *DR  *
02575     4 00 0 02576             14         PZE     QUBLDR,4
02576     0 20 0 00000             15 QUBLDR  NOP
02577     0 72 0 03604             16         SKA     MSO
02600     0 01 0 06614             17         BRU     LDZERO+02614
02601     0 01 0 02603             18         BRU     LDR1            PURGE LOADER'S SYMBOL TABLE
02602     0 76 0 03776             19 LOADER  LDA     TBIAS
02603     0 43 0 06742             20 LDR1    BRM     PURGE
02604     0 01 0 06614             21         BRJ     LDZERO+02614
02605                              22         RES     1
02606                              23 FLAG    RES     1
02607                              24         RES     5
02610     0 36 0 03656             25         STO     ETBL
02611     0 40 0 20400             26 LDR2    SKS     BP1OFF
02612     0 00 0 11111             27         HLT     011111          *WAIT- BREAKPOINT 1 ON (DOWN)
02613     0 20 0 11111             28         NOP     011111
02614     0 14 0 03621             29         ETR     MADR
02615     0 35 0 03646             30         STA     BIAS
02616     0 46 30003               31         CLR
02617     0 35 0 03650             32         STA     MLOC
          00002620                 33 STRT    EQU     $
02620     0 43 0 02703             34 LOOP    BRM     READ            READ NEXT RECORD, AND VERIFY
02621     0 43 0 03047             35         BRM     LOAD            PROCESS VERIFIED RECORD
02622     0 76 0 03645             36         LDA     T               LOAD TYPE CODE OF RECORD JUST PROCESS
02623     0 75 0 03605             37         LDB     N1
02624     0 70 0 03567             38         SKM     P3
```

```
02625  0 01 0 02620      39          BRU     LABP
02626  0 75 0 03670      40 TSTRR    LDB     SEQNR
02627  0 36 0 03646      41          STB     BIAS
02630  0 76 0 03651      42 TSTRR2   LDA     RCTR          TEST FOR UNSATISFIED REFS
02631  0 16 0 03653      43          MRG     PRCTR
02632  0 73 0 03564      44          SKG     P0
02633  0 01 0 02671      45          BRU     TSTDD         NO UNSATISFIED REFS
02634  0 53 0 03657      46          SKN     LIB           SKIP IF LOADER IS IN LIBRARY MODE
02635  0 20 00000        47          NOP
                          48 MESG     FORM    3,6,15
02636  0 43 0 03677      49 MESG1    BRM     MESP
02637  0 05 03722        50          MESG    0,5,MES1
02640  0 71 0 03625      51          LDX     LTBL,X0
02641  2 53 0 00002      52 MZG1     SKN     2,X2
02642  0 01 0 02644      53          BRU     $+2
02643  0 01 0 02646      54          BRU     MZG2
02644  0 43 0 03677      55          BRM     MESP
02645  2 02 00000        56          MESG    2,2,0
02646  2 77 0 37775      57 MZG2     EAX     -3,X2
02647  0 37 0 03634      58          STX     TMP3
02650  0 76 0 03634      59          LDA     TMP3
02651  0 14 0 03621      60          ETR     MADR
02652  0 35 0 03634      61          STA     TMP3
02653  0 76 0 03656      62          LDA     ETBL
02654  0 73 0 03634      63          SKG     TMP3
02655  0 01 0 02641      64          BRU     MZG1
02656  0 01 00753        65          BRJ     QUIT
02657  0 43 00650        66 MULDEF   BRM     ERRTYP
02660  44242652          67          TEXT    <MDF >        MULTIPLE DEFS
02661  0 01 00753        68          BRJ     QUIT
02662  0 43 00650        69 RECFMT   BRM     ERRTYP
02663  26446352          70          TEXT    <FMT >        ILLEGAL FORMAT
02664  0 01 00753        71          BRJ     QUIT
02665  0 43 00650        72 CHKERR   BRM     ERRTYP
02666  23624452          73          TEXT    <CSM >        CHECKSUM ERROR
02667  0 01 00753        74          BRJ     QUIT
02670  0 53 06606        75 SKIP     SKN     FLAG
02671  0 76 0 03652      76 TSTDD    LDA     DCTR
02672  0 16 0 03654      77          MRG     PDCTR
```

```
02673  0 72 0 03605     78          SKA     N1          SKIP IF NO DUPLICATE DEFS
02674  0 01 06657       79          BRU     MULDEF      MULTIPLE DEFS
02675  0 20 0 00010     80          NOP     8
02676  0 01 0 03514     81          BRU     LDAPCT
                        82 *
02677                   83 EXITL2   RES     0
02677  0 76 0 03646     84          LDA     BIAS
02700  0 20 00000       85          NOP                 /
02701  0 43 00664       86          BRM     QFORM       /LINK TO PROGRAM QUEUE
02702  00000000         87 EXIT     DATA    0
                        88 *
02703  0 00 00000       89 READ     PZE                 READ
02704  0 76 01067       90 CARD3    LDA     N1          /
02705  0 35 06606       91          STA     FLAG        SET FINISH FLAG
02706  1 76 06720       92          BIO     P40         /
02707  0 02 02604       93          RPT     0,1,4       /
02710  0 32 07727       94          WIM     CW          /
02711  0 61 06606       95          MIN     FLAG        /
02712  0 61 06606       96 CARD     MIN     FLAG        /
02713  1 73 06670       97          BUSY    SKIP        /
02714  0 40 00010       98          SETM                /
02715  0 01 07674       99 CARD2    BRU     CARD1       BUFFER ERROR TEST
02716  0 01 06776      100          BRU     LDZERO+02776
02717                  101          RES     1           /
02720  00000050        102 P40      DATA    40          P40
02721                  103          RES     8           /
02722  0 01 0 03674    104          BRU     CARD1       CARD FEED CHECK
02723  0 53 0 03657    105          SKN     LIB         SKIP IF LIBRARY MODE SWITCH ON
02724  0 01 0 02776    106          BRU     *-2
02725  0 76 0 03727    107          LDA     CW
02726  0 75 0 03603    108          LDB     N1
02727  0 70 0 03672    109          SKM     DECP        END OF LIBRARY
02730  0 01 0 02772    110          BRU     *+2
02731  0 43 00650      111 SOFERR  BRM     ERRTYP      SOFERR, SYMBOL TABLE OVERFLOW
02732  46264352        112          TEXT    <OFL >      /
02733  0 01 00753      113          BRU     QUIT        /
02734  1 76 01075      114 CARRTN  BIO     P1
02735  0 02 02641      115          TYP     0,1,4
02736  0 12 07722      116          MIN     MES1
```

```
02737  0 02 14000          117          TBR
02740  0 20 00000          118          NRP
02741  0 51 07677          119          BRR     *ESP
02742  0 00 00000          120 PURGE    PZE                      PURGE LOADER SYMBOL TABLE
02743  0 71 07612          121          LDX     LDZER0+03612     /N12
02744  0 75 01074          122          LDB     P0
02745  2 36 07663          123          STB     LDZER0+03663,2   /SDCTR
02746  0 41 06745          124          BRX     LDZER0+02745
02747  0 75 07625          125          LDB     LDZER0+03625     /LTBL
02750  0 36 07656          126          STB     LDZER0+03656     /ETBL
02751  0 51 06742          127          BRR     PURGE            RETURN
                            128 *
02752  0 53 0 03660        129 LMT12    SKN     LOADM            SKIP IF LOAD MODE SWITCH IS ON
02753  0 01 0 03040        130          BRU     ENDTST           TEST FOR END RECORD
02754  0 76 0 03727        131          LDA     CW
02755  0 6600 025          132          RSH     21
02756  0 14 0 03570        133          ETR     P7
02757  0 73 0 03564        134          SKG     P0
02760  0 01 0 02763        135          BRU     $+3               T=0  OR  T<0
02761  0 73 0 03566        136          SKG     P2
02762  0 01 0 02776        137          BRU     NRP               T=1 OR  T=2
02763  0 53 0 03661        138          SKN     PRGTST           SKIP IF PURGE TEST SWITCH IS ON
02764  0 01 0 02776        139          BRU     NRP
02765  0 35 0 03661        140          STA     PRGTST           RESET PURGE TEST SWITCH
02766  0 53 0 03671        141          SKN     NRS              SKIP IF NO REFS WERE SATISFIED
02767  0 01 0 02776        142          BRU     NRP              AT LEAST ONE REF WAS SATISFIED
02770  0 71 0 03607        143          LDX     N5,X0             P
02771  2 76 0 03670        144          LDA     SDCTR+5,X2        U
02772  2 35 0 03657        145          STA     DCTR+5,X2         R
02773  0 41 0 02771        146          BRX     $-2,X0            G
02774  0 35 0 03660        147          STA     LOADM             E - RESET LOAD MODE SWITCH
02775  0 01 0 02704        148          BRU     READ+1           SKIP TO NEXT ID RECORD
02776  0 46 30003          149 NRP      CLR
02777  0 75 0 03727        150          LDB     CW
03000  0 6700 003          151          LSH     3
03001  0 35 0 03645        152          STA     T                RECORD TYPE CODE
03002  0 76 0 03564        153          LDA     P0
03003  0 6700 006          154          LSH     6
03004  0 73 0 03571        155          SKG     P37              SKIP IF WORD COUNT TOO LARGE
```

```
03005  0 01 0 03011      156         BRU     READ1
03006  0 01 06662        157 READ2   BRJ     RECFMT          ILLEGAL RECORD FORMAT
03007  0 20 0 00003      158         NOP     3               ILLEGAL INPUT FORMAT
03010  0 01 0 02704      159         BRU     READ+1          REREAD
03011  0 54 0 03566      160 READ1   SUB     P2
03012  0 73 0 03605      161         SKG     N1              SKIP IF RECORD HAS AT LEAST 1 DATA WO
03013  0 01 0 03006      162         BRU     READ2
03014  0 35 0 03634      163         STA     WDCT            STORE WORD COUNT LESS 2
03015  0 35 0 03643      164         STA     CT
03016  0 76 0 03627      165         LDA     LDW1
03017  0 35 0 03630      166         STA     WDAD
03020  0 76 0 03727      167         LDA     CW
03021  0 14 0 03620      168         ETR     UPMSK
03022  0 17 0 03616      169         EOR     M7777
03023  0 17 1 03630      170 READ3   EOR     *WDAD           ACCUMULATE CHECKSUM
03024  0 60 0 03634      171         MDE     WDCT
03025  0 61 0 03630      172         MIN     WDAD            INCREMENT STORAGE ADDRESS
03026  0 53 0 03634      173         SKN     WDCT            SKIP IF LAST WORD CHECKSUMMED
03027  0 01 0 03023      174         BRU     READ3
03030  0 35 0 03630      175         STA     WDAD
03031  0 6600 014        176         RSH     12
03032  0 17 0 03630      177         EOR     WDAD
03033  0 75 0 03616      178         LDB     M7777
03034  0 70 0 03727      179         SKM     CW              SKIP IF CHECKSUMS AGREE
03035  0 01 06665        180         BRJ     CHKERR          CHECKSUM ERROR
03036  0 20 0 00004      181         NOP     4
03037  0 51 0 02703      182         BRR     READ            EXIT
                          183 *
                          184 *               TEST SKIPPED RECORD FOR END RECORD
03040  0 76 0 03727      185 ENDTST  LDA     CW
03041  0 75 0 03601      186         LDB     BTMSK
03042  0 70 0 03602      187         SKM     ENDMSK          SKIP IF END RECORD
03043  0 01 0 02704      188         BRU     READ+1          READ NEXT RECORD
03044  0 76 0 03564      189         LDA     PO              SET NEW
03045  0 35 0 03673      190         STA     NEWR                    ROUTINE FLAG.
03046  0 01 0 02704      191         BRU     READ+1          READ NEXT RECORD
                          192 *
                          193 *
                          194 *               PROCESS VERIFIED RECORDS
```

```
03047  0 00 0 00000   195 LOAD   PZE                          ENTRY
03050  0 76 0 03627   196        LDA     LDW1
03051  0 35 0 03630   197        STA     WD1                  INITIALIZE DATA WORD ADDR
03052  0 71 0 03645   198        LDX     T,X0                 (X2) := TYPE CODE
03053  2 01 1 03554   199        BRU     *TYPE,X2             BRANCH TO PROCESSING ROUTINE
                      200 *              PROCESS END RECORD
03054  0 76 0 03643   201 END    LDA     CT
03055  0 75 0 03564   202        LDB     PO
03056  0 36 0 03673   203        STB     NEWR                 SET NEW ROUTINE FLAG
03057  0 72 0 03571   204        SKA     P37                  SKIP IF NO TRANSFER WORD
03060  0 01 0 03074   205        BRU     END1
03061  0 76 0 03730   206        LDA     DW1
03062  0 55 0 03646   207        ADD     BIAS
03063  0 14 0 03621   208        ETR     MADR
03064  0 53 0 03657   209        SKN     LIB
03065  0 01 0 03067   210        BRU     $+2
03066  0 01 0 03071   211        BRU     $+3_                 DONT HALT IF IN LIBRARY MODE
03067  1 74 01053     212        READY   BPTST+2              BKPT 2 WAIT =ND RECORD
03070  0 20 00000     213        NOP
03071  0 20 0 22222   214        NOP     022222
03072  0 35 0 03646   215        STA     BIAS
03073  0 01 0 02620   216        BRU     STRT                 LOAD NEXT PROGRAM
                      217 *              PROCESS END RECORD WITH TRANSFER ADDRESS
03074  0 76 0 03730   218 END1   LDA     DW1
03075  0 55 0 03646   219        ADD     BIAS
03076  0 14 0 03621   220        ETR     MADR
03077  0 35 0 03670   221        STA     SEQNR
03100  0 76 0 03730   222        LDA     DW1
03101  0 14 0 03617   223        ETR     NAMSK
03102  0 16 0 03626   224        MRG     LEXIT
03103  0 35 0 03730   225        STA     DW1
                      226 *              PROCESS DATA RECORD
                      227 *              PROCESS MODIFIER WORDS
03104  0 76 0 03642   228 DATA   LDA     LSREL
03105  0 35 0 03633   229        STA     TLREL                INITIALIZE ADDRESS OF MODIFIERS
03106  0 76 0 03730   230        LDA     DW1                  (A) = LOAD-ADDRESS WORD
03107  0 71 0 03606   231        LDX     N4
03110  0 37 0 03634   232 RLOOP  STX     CTR
03111  0 75 0 03564   233        LDB     PO
```

```
03112  0 6700 001          234          LSH     1
03113  0 73 0 03605        235          SKG     N1                  SKIP IF NO MODIFIER WORD PRESENT
03114  0 01 0 03116        236          BRU     $+2
03115  0 01 0 03121        237          BRU     $+4
03116  0 71 0 03643        238          LDX     CT,X0
03117  0 60 0 03643        239          MDE     CT                  REDUCE WORD COUNT
03120  2 75 0 03730        240          LDB     DW1,X2              (B) := MODIFIER WORD
03121  0 36 1 03633        241          STB     *TLREL              STORE MODIFIER WORD OR ZERO
03122  0 61 0 03633        242          MIN     TLREL
03123  0 71 0 03634        243          LDX     CTR
03124  0 41 0 03110        244          BRX     RLOOP
                            245 *                                   CT = NR OF DATA WORDS
                            246 *       PROCESS LOAD ADDRESS
03125  0 76 0 03730        247          LDA     DW1
03126  0 72 0 03614        248          SKA     LRF
03127  0 55 0 03646        249          ADD     BIAS
03130  0 14 0 03622        250          ETR     MADR1               (A) = ADJUSTED LOAD ADDRESS
03131  0 35 0 03644        251          STA     LOC                 STORE MODIFIED LOAD ADDRESS
                            252 *       PROCESS DATA WORDS
03132  0 60 0 03643        253          MDE     CT                  (CT) = NR OF DATA WORDS LESS 1
03133  0 76 0 03631        254          LDA     LDW2
03134  0 35 0 03635        255          STA     IWD                 INDIRECT ADDRESS OF 1ST DATA WORD
                            256 *
03135  0 76 1 03635        257 LLDA     LDA     *IWD                (A) = CURRENT DATA WORD
03136  0 53 0 03636        258          SKN     SREL                SKIP IF (A) REQUIRES SPECIAL RELOC
03137  0 01 0 03170        259          BRU     LDBB
                            260 *       PROCESS SPECIAL I/O OPERATION WORDS
03140  0 75 0 03646        261 SPRLC    LDB     BIAS
03141  0 53 0 03641        262          SKN     LREL                SKIP IF LOAD RELOCATION SPECIFIED
03142  0 01 0 03144        263          BRU     $+2
03143  0 01 0 03147        264          BRU     $+4
03144  0 75 0 03647        265          LDB     CBIAS
03145  0 53 0 03640        266          SKN     CREL                SKIP IF COMMON RELOCATION SPECIFIED
03146  0 01 0 03202        267          BRU     LSTA                EXIT
03147  0 36 0 03632        268          STB     TMP1
03150  0 14 0 03574        269          ETR     MSKA                ISOLATE BASE ADDRESS
03151  0 55 0 03632        270          ADD     TMP1                ADD SPECIFIED BIAS
03152  0 35 0 03632        271          STA     TMP1
03153  0 73 0 03574        272          SKG     MSKA                SKIP IF (A) > 037777
```

```
03154  0 01 0 03164        273          BRU     SPRLC1              NO CHANGE TO 15TH BIT OF ADDRESS
03155  0 14 0 03574        274          ETR     MSKA
03156  0 35 0 03632        275          STA     TMP1                SAVE LOWER 14 BITS OF ADJUSTED ADDR
03157  0 60 0 03644        276          MDE     LOC                 CURRENT LOAD ADDRESS LESS 1
03160  0 76 1 03644        277          LDA     *LOC
03161  0 17 0 03573        278          EOR     INVRT               COMPLEMENT 15TH BIT OF ADDRESS
03162  0 35 1 03644        279          STA     *LOC                STORE ADJUSTED I/O OPERATION WORD
03163  0 61 0 03644        280          MIN     LOC
03164  0 76 1 03635        281 SPRLC1   LDA     *IWD                (A) = CURRENT DATA WORD
03165  0 14 0 03600        282          ETR     MSKC                ISOLATE COUNT
03166  0 16 0 03632        283          MRG     TMP1                MRG LOWER 14 BITS OF ADJUSTED ADDRESS
03167  0 01 0 03202        284          BRU     LSTA                EXIT
                            285 *
                            286 *
03170  0 75 0 03646        287 LDBB     LDB     BIAS                (B) = LOAD RELOCATION BIAS
03171  0 53 0 03641        288          SKN     LREL                SKIP IF (A) LOAD RELOCATABLE
03172  0 01 0 03175        289          BRU     $+3
03173  0 43 0 03223        290          BRM     RELOC               ADJUST RELOCATABLE ADDRESS
03174  0 01 0 03524        291          BRU     LLDB
                            292 *
03175  0 75 0 03647        293          LDB     CBIAS               (B) = COMMON RELOCATION BIAS
03176  0 53 0 03640        294          SKN     CREL                SKIP IF (A) COMMON RELOCATABLE
03177  0 01 0 03201        295          BRU     $+2
03200  0 43 0 03223        296          BRM     RELOC               ADJUST RELOCATABLE ADDRESS
03201  0 01 0 03524        297          BRU     LLDB
                            298 *
03202  0 35 1 03644        299 LSTA     STA     *LOC                STORE ADJUSTED DATA WORD
03203  0 61 0 03644        300          MIN     LOC                 INCREMENT LOAD ADDRESS
03204  0 61 0 03635        301          MIN     IWD                 INCREMENT DATA WORD ADDRESS
                            302 *                CYCLE RELOCATION WORDS
03205  0 71 0 03606        303          LDX     N4,X0
03206  2 75 0 03642        304          LDB     SREL+4,X2
03207  0 6700 001          305          LSH     1
03210  2 36 0 03642        306          STB     SREL+4,X2
03211  0 41 0 03206        307          BRX     $-3,X0
                            308 *
03212  0 60 0 03643        309          MDE     CT
03213  0 20 0 00000        310          NOP
03214  0 53 0 03643        311          SKN     CT                  SKIP IF ALL DATA WORDS PROCESSED
```

```
03215  0 01 0 03135        312          BRU     LLDA
                           313 *
03216  0 76 0 03650        314          LDA     MLOC
03217  0 73 0 03644        315          SKG     LOC
03220  0 76 0 03644        316          LDA     LOC
03221  0 35 0 03650        317          STA     MLOC                STORE CURRENT MAXIMUM LOAD ADDRESS +1
03222  0 51 0 03047        318          BRR     LOAD                EXIT
                           319 *
                           320 *                PERFORM LOAD OR COMMON RELOCATION
03223  0 00 0 00000        321 RELOC    PZE     0                   ENTRY
03224  0 36 0 03632        322          STB     TMP1                SAVE BIAS
03225  0 14 0 03621        323          ETR     MADR
03226  0 55 0 03632        324          ADD     TMP1                ADD SPECIFIED BIAS
03227  0 14 0 03621        325          ETR     MADR                (A) = ADJUSTED ADDRESS
03230  0 35 0 03632        326          STA     TMP1
03231  0 76 1 03635        327          LDA     *IWD                (A) = CURRENT DATA WORD
03232  0 14 0 03623        328          ETR     MRDA
03233  0 16 0 03632        329          MRG     TMP1                (A) = ADJUSTED DATA WORD
03234  0 51 0 03223        330          BRR     RELOC               EXIT
                           331 *
                           332 *                APPLY SPECIFIED RELOCATION BIASES TO ADDRESS IN A REGISTE
                           333 *                PROCESS SPECIAL I/O OPERATION WORDS
                           334 *
                           335 *                ROUTINES FOR PROCESSING EXTERNAL REFS AND DEFS
                           336 *                RECORD TYPES 1 AND 2
03235  0 76 0 03567        337 NEXTI    LDA     P3
03236  0 55 0 03630        338          ADD     WD1
03237  0 35 0 03630        339          STA     WD1                 INCREMENT ITEM ADDRESS
03240  0 76 0 03643        340          LDA     CT
03241  0 54 0 03567        341          SUB     P3
03242  0 35 0 03643        342          STA     CT                  REDUCE WORD COUNT
03243  0 73 0 03564        343          SKG     P0                  SKIP IF AT LEAST 3 MORE WORDS
03244  0 51 0 03047        344          BRR     LOAD                EXIT
                           345 *
                           346 *                PROCESS NEXT ITEM, HEREAFTER CALLED 'C'
03245  0 71 0 03630        347 EXT      LDX     WD1,X0              (X0) = LOC. OF 1ST WORD OF ITEM
03246  0 75 0 03645        348          LDB     T                   (B) = RECORD TYPE CODE
03247  0 6700 027          349          LSH     23
03250  0 36 0 03636        350          STB     TT                  (TT) NEGATIVE IF (T) = 1
```

```
03251  2 76 0 00002      351          LDA     2,X2          (A) = CHAIN, VALUE, OR POP ADDR WORD
03252  0 14 0 03575      352          ETR     LRMADR
03253  0 72 0 03614      353          SKA     LRF
03254  0 55 0 03646      354          ADD     BIAS
03255  0 14 0 03621      355          ETR     MADR          (A) = ADJUSTED VALUE OF CHAIN ADDRESS
03256  0 35 0 03644      356          STA     LOC           SAVE ADJUSTED ADDRESS
03257  2 75 0 00002      357          LDB     2,X2
03260  0 36 0 03640      358          STB     STYP          SET STYP + FOR REF, - FOR DEF
03261  0 6700 002        359          LSH     2
03262  0 14 0 03567      360          ETR     P3
03263  0 35 0 03637      361          STA     ST            (ST) = SUBTYPE CODE SCALED ZERO
03264  0 76 0 03564      362          LDA     P0
03265  0 6700 006        363          LSH     6
03266  0 35 0 03670      364          STA     SEQNR         SEQUENCE NR IF C IS A POP ITEM
03267  2 76 0 00000      365          LDA     0,X2
03270  2 75 0 00001      366          LDB     1,X2          (A,B) = 8 CHARACTER SYMBOL
03271  0 71 0 03637      367          LDX     ST,XO         (X2) = 2 BIT SUBTYPE CODE
03272  0 53 0 03636      368          SKN     TT
03273  2 01 1 03541      369          BRU     *POPTYP,X2    BRANCH TO POP SUBTYPE ROUTINE
03274  2 01 1 03545      370          BRU     *SUBTYP,X2    BRANCH TO LABEL SUBTYPE ROUTINE
                          371 *
03275  0 43 0 03405      372 DEF      BRM     SRCH
03276  0 53 0 03636      373          SKN     TT            SKIP IF C IS A LABEL ITEM
03277  0 01 0 03356      374          BRU     PNTCH         T=2
03300  2 53 0 00002      375          SKN     2,X2          SKIP IF X IS A DEF
03301  0 01 0 03307      376          BRU     TXR           X IS A REF
03302  0 76 0 03644      377          LDA     LOC           X IS A DEF
03303  0 53 0 03640      378          SKN     STYP          SKIP IF C IS A DEF
03304  0 01 0 03320      379          BRU     TXDR          C IS A REF
03305  0 61 0 03652      380          MIN     DCTR          DUPLICATE DEF
03306  0 01 0 03235      381          BRU     NEXTI
                          382 *
                          383 *                 X IS A REF
03307  0 53 0 03640      384 TXR      SKN     STYP          SKIP IF C IS A DEF
03310  0 01 0 03337      385          BRU     TXRR          C IS A REF
03311  0 60 0 03651      386 TXRD     MDE     RCTR          C IS A DEF, DECREMENT REF COUNTER
03312  0 61 0 03671      387          MIN     NRS           INCREMENT SATISFIED REF. COUNTER
03313  0 76 0 03644      388          LDA     LOC           VALUE OF C
03314  0 16 0 03604      389          MRG     MBO           SET DEF FLAG
```

```
03315  2 75 0 00002      390          LDB     2,X2
03316  2 35 0 00002      391          STA     2,X2
03317  0 46 00014        392          XAB                     REF X BECOMES DEF X
03320  0 14 0 03621      393 TXDR     ETR     MADR
03321  0 35 0 03634      394 TXRD1    STA     CHAIN           ADDR OF A REF
03322  0 76 1 03634      395          LDA     *CHAIN
03323  0 35 0 03632      396          STA     M
03324  0 14 0 03623      397          ETR     MRDA
03325  0 35 0 03633      398          STA     TMP2
03326  2 76 0 00002      399          LDA     2,X2
03327  0 14 0 03621      400          ETR     MADR            (A) = VALUE OF SYMBOL X
03330  0 16 0 03633      401          MRG     TMP2
03331  0 35 1 03634      402          STA     *CHAIN          STORE COMPLETED REF
03332  0 76 0 03632      403          LDA     M
03333  0 14 0 03621      404          ETR     MADR
03334  0 73 0 03564      405          SKG     P0              SKIP IF NOT LAST REF IN CHAIN
03335  0 01 0 03235      406          BRU     NEXTI           EXIT 1  LAST REF IN CHAIN
03336  0 01 0 03321      407          BRU     TXRD1
                          408 *               X AND C ARE BOTH REFS
03337  0 76 0 03644      409 TXRR     LDA     L0C             ADDRESS OF 1ST REF IN NEW CHAIN
03340  0 35 0 03634      410 TXRR1    STA     CHAIN
03341  0 76 1 03634      411          LDA     *CHAIN
03342  0 14 0 03621      412          ETR     MADR
03343  0 72 0 03605      413          SKA     N1
03344  0 01 0 03340      414          BRU     TXRR1
03345  0 76 0 03644      415          LDA     L0C             NEW 1ST LINK
03346  0 16 0 03603      416          MRG     MB1             SET REF FLAG
03347  2 75 0 00002      417          LDB     2,X2
03350  2 35 0 00002      418          STA     2,X2
03351  0 46 00014        419          XAB                     STORE NEW FIRST LINK IN TABLE
03352  0 14 0 03621      420          ETR     MADR            REMOVE REF FLAG FROM OLD 1ST LINK
03353  0 16 1 03634      421          MRG     *CHAIN          ADDRESS FIELD WAS ZERO
03354  0 35 1 03634      422          STA     *CHAIN          LINK NEW CHAIN TO OLD CHAIN
03355  0 01 0 03235      423          BRU     NEXTI           EXIT 3
                          424 *
                          425 *               MATCHED POP PROCESSING
03356  2 76 0 00002      426 PMTCH    LDA     2,X2            (A) = WORD 3 OF ITEM X
03357  2 53 0 00002      427          SKN     2,X2            SKIP IF X IS A DEF
03360  0 01 0 03366      428          BRU     PXR               X IS A REF
```

```
                                429 *                MATCHED DEF
03361   0 53 0 03640            430 PXD     SKN      STYP          X IS A DEF, SKIP IF C IS A DEF
03362   0 01 0 03500            431         BRU      PXDCR           C IS A REF
03363   0 61 0 03654            432 PXDCD   MIN      PDCTR           C IS A DEF, ALBEIT A DUPLICATE DEF
03364   2 35 0 00002            433         STA      2,X2          SET DUP FLAG , RETAIN PREVIOUS VALU
03365   0 01 0 03500            434         BRU      PXDCR         FROM THIS POINT PROCESS AS A REF
                                435 *
                                436 *                MATCHED REF
03366   0 53 0 03640            437 PXR     SKN      STYP          SKIP IF CIS A DEF
03367   0 01 0 03500            438         BRU      PXRCR           C IS A REF
03370   0 17 0 03603            439 PXRCD   EOR      MB1             C IS A DEF
03371   0 61 0 03671            440         MIN      NRS           INCREMENT SATISFIED REF. COUNTER
03372   0 16 0 03604            441         MRG      MBC           CHANGE X FROM A REF TO A DEF
03373   0 16 0 03644            442 PXRCD1  MRG      LOC           (LOC)=ACTUAL POP SUBR ORIGIN
03374   2 35 0 00002            443         STA      2,X2          SET WORD3 OF ITEM X
03375   0 35 0 03632            444         STA      TMP1          SAVE WORD3 OF ITEM X
03376   0 6600 001              445         RSH      1
03377   0 14 0 03576            446         ETR      OPMSK
03400   0 6600 017              447         RSH      15            (A) = ACTUAL 6-BIT POP CODE FROM X
03401   0 35 0 03634            448         STA      TMP3
03402   0 71 0 03634            449         LDX      TMP3
03403   0 60 0 03653            450         MDE      PRCTR         DECREMENT POP REF COUNTER
03404   0 01 0 03473            451         BRU      SUBORG
                                452 *
                                453 *                SYMBOL TABLE SEARCH SUBROUTINE
03405   0 00 0 00000            454 SRCH    PZE      0             ENTRY
03406   0 71 0 03625            455         LDX      LTBL,XO       (X2) = ORIGIN OF SYMBOL TABLE
03407   0 36 0 03633            456         STB      TMP2
03410   0 35 0 03632            457         STA      TMP1
03411   0 75 0 03605            458         LDB      N1
03412   0 76 0 03632            459 SRCH1   LDA      TMP1          (A) = 1ST 4 CHARS OF C
03413   2 70 0 00000            460         SKM      0,X2          SKIP IF C(1 THRU 4) = X(1 THRU 4)
03414   0 01 0 03427            461         BRU      SRCH2
03415   0 76 0 03633            462         LDA      TMP2          (A) = 2D 4 CHARS OF C
03416   2 70 0 00001            463         SKM      1,X2          SKIP IF C(5 THRU 8) = X(5 THRU 8)
03417   0 01 0 03427            464         BRU      SRCH2
                                465 *                              TEST FOR MATCH ON TYPE
03420   0 76 0 03564            466         LDA      PO
03421   0 53 0 03636            467         SKN      TT            SKIP IF T =1 (I.E. X IS A LABEL ITEM)
```

```
03422  0 76 0 03614      468          LDA     LRF                T=2 (I.E., X IS A POP ITEM)
03423  2 17 0 00002      469          EOR     2,X2
03424  0 72 0 03614      470          SKA     LRF                SKIP IF C AND X HAVE SAME TYPE CODE
03425  0 01 0 03427      471          BRU     SRCH2
03426  0 51 0 03405      472          BRR     SRCH               EXIT ON MATCH,SYMBOLS AND TYPES AGREE
                         473 *
03427  2 77 0 37775      474 SRCH2    EAX     -3,X2              (X2) := (X2) - 3
03430  0 37 0 03634      475          STX     TMP3
03431  0 76 0 03656      476          LDA     ETBL
03432  0 73 0 03634      477          SKG     TMP3               SKIP IF END OF TABLE REACHED
03433  0 01 0 03412      478          BRU     SRCH1              TEST NEXT TABLE ENTRY
                         479 *
                         480 *
                         481 *                                  INSERT NEW ENTRY IN SYMBOL TABLE
03434  0 76 0 03634      482 NSRT     LDA     TMP3
03435  0 73 0 03650      483          SKG     MLOC               (ETBL)-3 : (MLOC)
03436  0 01 06731        484          BRU     SOFERR             SYMBOL TABLE OVERFLOW
03437  0 20 0 00006      485          NOP     6
03440  0 35 0 03656      486          STA     ETBL               ADDRESS OF LAST TABLE ENTRY
03441  0 71 0 03656      487          LDX     ETBL
03442  0 76 0 03632      488          LDA     TMP1
03443  0 75 0 03633      489          LDB     TMP2
03444  2 35 0 00000      490          STA     0,X2               STORE 1ST LABEL WORD
03445  2 36 0 00001      491          STB     1,X2               STORE 2D LABEL WORD
03446  0 75 0 03637      492          LDB     ST
03447  0 6700 056        493          LSH     46                 (A) = SUB TYPE CODE SCALED 22 LEFT
03450  0 16 0 03644      494          MRG     LOC
03451  2 35 0 00002      495          STA     2,X2
03452  0 53 0 03636      496          SKN     TT                 SKIP IF TYPE CODE (T) IS 1
03453  0 01 0 03457      497          BRU     NSRTP
03454  2 53 0 00002      498          SKN     2,X2               SKIP IF C IS AN EXTERNAL DEF
03455  0 61 0 03651      499          MIN     RCTR
03456  0 01 0 03235      500          BRU     NEXTI
                         501 *                                  INSERT ACUTAL POP CODE AND TYPE INDICATOR IN WORD 3
03457  0 75 0 03655      502 NSRTP    LDB     POPCTR
03460  0 6700 050        503          LSH     40                 (A) = ACTUAL POP CODE SCALED 16 LEFT
03461  2 16 0 00002      504          MRG     2,X2               (WORD 3 ALREADY CONTAINED (LOC))
03462  0 16 0 03614      505          MRG     LRF                SET TYPE INDICATOR (L/P)
03463  2 35 0 00002      506          STA     2,X2               SET WORD 3 OF C IN SYMBOL TABLE
```

```
03464  2 53 0 00002          507          SKN     2,X2                SKIP IF C IS A P0P DEF
03465  0 61 0 03653          508          MIN     PRCTR               INCREMENT EXTERNAL P0P REF COUNTER
03466  0 76 0 03655          509 IP0PD    LDA     P0PCTR              (A) = P0P C0DE FR0M C
03467  0 71 0 03655          510          LDX     P0PCTR              (X0) = P0P C0DE FR0M C
03470  0 6700 020            511          LSH     16
03471  0 35 0 03632          512          STA     TMP1                SAVE SCALED P0P C0UNTER
                             513 * P0P TABLE      * INSERT ACTUAL P0P SUBR0UTINE ADDRESS INT0 P0P
03472  0 61 0 03655          514          MIN     P0PCTR              INCREMENT P0P C0UNTER
03473  2 76 0 00100          515 SUB0RG   LDA     64,X2
03474  0 14 0 03576          516          ETR     0PMSK
03475  0 16 0 03644          517          MRG     L0C
03476  2 35 0 00100          518          STA     64,X2               SET P0P SUBR 0RIGIN IN P0P TABLE
                             519 * P0P RELOC TABLE * INSERT ACTUAL 6-BIT P0P C0DE INT0 P0P RELOC T
03477  0 76 0 03632          520          LDA     TMP1                (A) = W0RD3 0F ITEM X 0R (P0PCTR)SC16
       00003500              521 PXDCR    EQU     $
03500  0 6600 001            522 PXRCR    RSH     1
03501  0 14 0 03576          523          ETR     0PMSK               (A) = ACTUAL P0P C0DE SCALED 15 LEFT
                             524 *
03502  0 71 0 03670          525          LDX     SEQNR,X0            (SEQNR)+64 IS P0P RELOC TABLE ADDRESS
03503  0 43 0 03505          526          BRM     NSRT0P              SET P0P C0DE IN P0P RELOC TABLE ENTRY
03504  0 01 0 03235          527          BRU     NEXTI               PR0CESS NEXT P0P ITEM
                             528 *
03505  0 00 0 00000          529 NSRT0P   PZE                         ENTRY
03506  0 35 0 03632          530          STA     TMP1
03507  2 76 0 00100          531          LDA     64,X2
03510  0 14 0 03577          532          ETR     N0PMSK
03511  0 16 0 03632          533          MRG     TMP1
03512  2 35 0 00100          534          STA     64,X2
03513  0 51 0 03505          535          BRR     NSRT0P              EXIT
                             536 *
                             537 *                SUBR0UTINE T0 SET "BRU" INSTR C0DE IN P0P TABLES ENTRIES
03514  0 76 0 03655          538 LDAPCT   LDA     P0PCTR
03515  0 73 0 03564          539          SKG     P0
03516  0 01 0 02677          540          BRU     EXITL2
03517  0 60 0 03655          541          MDE     P0PCTR
03520  0 71 0 03655          542          LDX     P0PCTR
03521  0 76 0 03624          543          LDA     P0PBRU
03522  0 43 0 03505          544          BRM     NSRT0P
03523  0 01 0 03514          545          BRU     LDAPCT
```

```
                                546 *
03524  0 53 0 03637             547 LLDB    SKN     PREL                SKIP IF (A) HAS PROGRAMME OPERATOR OP
03525  0 01 0 03202             548         BRU     LSTA
                                549 *               PROGRAMMED OPERATOR RELOCATION
03526  0 14 0 03577             550         ETR     NOPMSK              SET LOWER 6 BITS OF OP FIELD TO ZERO
03527  0 35 0 03632             551         STA     TMP1
03530  0 76 1 03635             552         LDA     *IWD                (A) = CURRENT DATA WORD
03531  0 14 0 03576             553         ETR     OPMSK
03532  0 6600 017               554         RSH     15                  (A) = POP SEQUENCE NR
03533  0 35 0 03634             555         STA     TMP3
03534  0 71 0 03634             556         LDX     TMP3
03535  2 76 0 00100             557         LDA     64,X2               (A) = POP TABLE ENTRY
03536  0 14 0 03576             558         ETR     OPMSK               (A) = (POP INSTR CODE - 0100)*2**15
03537  0 16 0 03632             559         MRG     TMP1                INSERT ACTUAL POP CODE IN DATA WORD
03540  0 01 0 03202             560         BRU     LSTA
                                561 *
03541  0 00 0 03466             562 POPTYP  PZE     IPOPD               0-INTERNAL POP DEF
03542  0 00 0 03275             563         PZE     XPOPR               1-EXTERNAL POP REF
03543  0 00 0 03275             564         PZE     XPOPD               2-EXTERNAL POP DEF
03544  0 00 0 03006             565         PZE     READ2               3-NOT ASSIGNED
                                566 *
       00003275                 567 REF     EQU     DEF
       00003275                 568 XPOPR   EQU     DEF
       00003275                 569 XPOPD   EQU     DEF
       00003245                 570 POP     EQU     EXT
                                571 *
       00003006                 572 COMREF  EQU     READ2
03545  0 00 0 03551             573 SUBTYP  PZE     COMDEF
03546  0 00 0 03275             574         PZE     REF
03547  0 00 0 03275             575         PZE     DEF
03550  0 00 0 03006             576         PZE     COMREF
                                577 *               POP SUBTYPES
                                578 *
                                579 *
                                580 *
                                581 *               PROCESS BLANK COMMON DECLARATION
03551  0 76 0 03644             582 COMDEF  LDA     LOC                 CURRENT VALUE OF LOCATION COUNTER
03552  0 35 0 03647             583         STA     CBIAS               SET ORIGIN OF BLANK COMMON
03553  0 01 0 03235             584         BRU     NEXTI
```

```
                                585 *
                                586 *                TABLE OF PROCESSING ROUTINE ADDRESSES
03554   0 00 0 03104            587 TYPE    PZE      DATA                    DATA RECORD ROUTINE
03555   0 00 0 03245            588         PZE      EXT                     EXTERNAL REF AND DEF ROUTINE
03556   0 00 0 03245            589         PZE      POP
03557   0 00 0 03054            590         PZE      END                     END RECORD ROUTINE
03560   0 00 0 03006            591         PZE      READ2
03561   0 00 0 03006            592         PZE      READ2
03562   0 00 0 03006            593         PZE      READ2
03563   0 00 0 03006            594         PZE      READ2
                                595 *
03564   00000000                596 P0      DATA     0
03565   00000001                597 P1      DATA     01
03566   00000002                598 P2      DATA     02
03567   00000003                599 P3      DATA     03
03570   00000007                600 P7      DATA     07
03571   00000037                601 P37     DATA     037
03572   00000077                602 P77     DATA    077
03573   00000040                603 INVRT   DATA     040
03574   00037777                604 MSKA    DATA     037777
03575   00177777                605 LRMADR  DATA     0177777
03576   07700000                606 OPMSK   DATA     007700000
03577   70077777                607 NOPMSK  DATA     070077777
03600   77740000                608 MSKC    DATA     077740000
03601   70070000                609 BTMSK   DATA     070070000
03602   30050000                610 ENDMSK  DATA     030050000
03603   20000000                611 MB1     DATA     020000000
03604   40000000                612 MB0     DATA     040000000
03605   77777777                613 N1      DATA     077777777
                                614 INDEX   FORM     9,15
03606   001 77774               615 N4      INDEX    1,-4
03607   001 77773               616 N5      INDEX    1,-5
03610   001 77772               617 N6      INDEX    1,-6
03611   00077770                618 N8      DATA    077770
03612   001 77764               619 N12     INDEX    1,-12
03613   00077730                620 N40     DATA    077730
03614   00100000                621 LRF     DATA     0100000
03615   00200000                622 CRF     DATA     0200000
03616   00007777                623 M7777   DATA     07777
```

```
03617  76000000          624 NAMSK   DATA    076000000
03620  77770000          625 UPMSK   DATA    077770000
03621  00037777          626 MADR    DATA    037777*/S9300++1
03622  4 00 0 37777      627 MADR1   HLT     -1,4
03623  77740000          628 MPDA    DATA    077740000*/S9300          TABSIZ EQU
03624  0 01 0 00000      629 POPBRU  ERU     0
       00002573          630 TBL     EQU     ITBL
03625  0 00 0 06565      631 LTBL    PZE     LDZER8+02573-TABSIZ
03626  4 00 0 02702      632 LEXIT   PZE     EXIT,4
03627  0 00 0 03730      633 LDW1    PZE     DW1
03630  0 00 0 03730      634 WD1     PZE     DW1
       00003630          635 WDAD    EQU     WD1
03631  0 00 0 03731      636 LDW2    PZE     DW1+1
03632  00000000          637 TMP1    DATA    0                  TMP1L  EQU     LDZER8+03632
03633  00000000          638 TMP2    DATA    0
03634  00000000          639 TMP3    DATA    0
       00003634          640 CHAIN   EQU     TMP3
       00003634          641 ID      EQU     TMP3
03635  0 00 0 00000      642 IWD     PZE     0
03636  00000000          643 SREL    DATA    0
03637  00000000          644 PREL    DATA    0
03640  00000000          645 CREL    DATA    0
03641  00000000          646 LREL    DATA    0
03642  0 00 0 03636      647 LSREL   PZE     SREL
       00003633          648 TLREL   EQU     TMP2
       00003633          649 MESPT   EQU     TMP2
       00003634          650 CTR     EQU     TMP3
       00003637          651 ST      EQU     PREL
       00003640          652 STYP    EQU     CREL
       00003634          653 WDCT    EQU     TMP3
03643  00000000          654 CT      DATA    0
03644  00000000          655 LOC     DATA    0
03645  00000000          656 T       DATA    0
03646  00000000          657 BIAS    DATA    0
03647  0 00 0 00000      658 CBIAS   PZE     0                          COMMON STORAGE ORIGIN
03650  00000000          659 MLOC    DATA    0
03651  00000000          660 RCTR    DATA    0
03652  00000000          661 DCTR    DATA    0
03653  00000000          662 PRCTR   DATA    0
```

```
03654  00000000              663 PDCTR   DATA    0
03655  00000000              664 P0PCTR  DATA    0
03656  0 00 0 02573          665 ETBL    PZE     TBL
03657  00000000              666 LIB     DATA    0
03660  00000000              667 L0ADM   DATA    0
03661  00000000              668 PRGTST  DATA    0
03662  00000000              669 SRCTR   DATA    0
03663  00000000              670 SDCTR   DATA    0
03664  00000000              671 SPRCTR  DATA    0
03665  00000000              672 SPDCTR  DATA    0
03666  00000000              673 SP0PCR  DATA    0
03667  00000000              674 SETBL   DATA    0
03670  00000000              675 SEQNR   DATA    0
03671  77777777              676 NRS     DATA    -1
       00003632              677 M       EQU     TMP1
       00003636              678 TT      EQU     SREL
03672  17170000              679 DE0F    DATA    017170000
03673  00000000              680 NEWR    DATA    0
                             681 *
03674  0 43 00650            682 CARD1   BRM     ERRTYP          /
03675  63214752              683         TEXT    <TAP >          /
03676  0 01 00753            684 *       BRU     QUIT            TYPE *ERRTAP*, QUIT
                             685
                             686 *
                             687 *
03677  0 20 0 00000          688 MESP    N0P     0               MESSAGE PRINT
03700  0 61 0 03677          689         MIN     MESP
03701  0 76 1 03677          690         LDA     *MESP
03702  0 35 0 03633          691         STA     MESPT
03703  0 6600 017            692         RSH     15
03704  0 14 0 03572          693         ETR     P77
03705  0 35 07632            694         STA     TMP1L           NUMBER 9F W0RDS T9 TYPE
03706  0 76 01067            695         LDA     N1
03707  0 35 07720            696         STA     FLG             SET UP EXIT FLAG
03710  1 76 07632            697 MESP1   BIS     TMP1L           /
03711  0 02 02641            698         TYP     0,1,4           /
03712  0 12 47633            699         MIN     *MESPT          /
03713  0 02 14000            700         T0P     0               /
03714  0 61 07720            701         MIN     FLG
```

```
03715  1 73 07721          702          BUSY    SKPFLG
03716  0 01 06734          703          BRU     CARRTN
03717                       704          RES     1
03720                       705 FLG      RES     1                FLG
03721  0 53 07720           706 SKPFLG   SKN     FLG              SKPFLG
03722  12121252             707 MES1     DATA    012121252
03723  44316262             708          DATA    044316262
03724  31452712             709          DATA    031452712
03725  24252662             710          DATA    024252662
03726  52121212             711          DATA    052121212
                            712 *
03727                       713 CW       RES     1
03730                       714 DW1      RES     050
04000                       715          RES     0
                            716 *
                            717 *
       00002602             718          END     LOADER
```

Appendix C

QMONITOR, Loader, and User Program Dump Module

PRECEDING PAGE BLANK NOT FILMED.

```
                                    1  ********************************************************************
                                    2  *                                                                  *
                                    3  *       MONITOR, LOADER AND USER PROGRAM DUMP                      (DUMP)*
                                    4  *                                                                  *
                                    5  ********************************************************************
        00010000                    6  MEMSIZ EQU     010000
        17100000                    7  ARM     OPD     017100000
00000                               8          RORG    0
00000   00000000                    9  LAST    DATA    *
00001                              10  CHKSUM  RES     1
00002   0 20 00001                 11  DUMP    NOP     1
00003   0 02 00004                 12          DIR
00004   0 46 00003                 13          CLR
00005   0 54 00000                 14          SUB     LAST
00006   0 35 00000                 15          STA     LAST
00007   0 71 00107                 16          LDX     BTSIZE
00010   0 02 00644                 17          PTL     0,1,4           /
00011   2 12 00107                 18          MIM     BTSIZE,2
00012   0 41 00011                 19          BRX     *-1             /
00013   0 02 14000                 20          TRBM                    /
00014   0 40 01000                 21          BRT*
00015   0 01 00014                 22          BRJ     *-1
00016   0 20 00000                 23          NOP                     /PUNCH BOOTSTRAP
00017   0 71 00112                 24          LDX     *-01212         /
00020   0 46 00003                 25          CLR                     /
00021   0 02 00644                 26          PTL     0,1,4           /
00022   2 12 07777                 27          MIM     MEMSIZ-1,2
00023   2 54 07777                 28          SUB     MEMSIZ-1,2
00024   0 41 00022                 29          BRX     *-2
00025   0 35 07777                 30          STA     MEMSIZ-1        /FORM LOADER CHECKSUM
00026   0 12 07777                 31          MIM     MEMSIZ-1
00027   0 02 14000                 32          TRBM                    /
00030   0 40 01000                 33          BRT*
00031   0 01 00030                 34          BRJ     *-1
00032   0 20 00000                 35          NOP                     /PUNCH LOADER
00033   0 71 00000                 36          LDX     LAST            /
00034   0 46 00003                 37          CLR                     /
00035   0 02 00644                 38          PTL     0,1,4           /
00036   2 12 00001                 39          MIM     CHKSUM,2
00037   2 54 00001                 40          SUB     CHKSUM,2        /
00040   0 41 00036                 41          BRX     *-2
00041   0 35 00001                 42          STA     CHKSUM          /FORM CHECK SUM ON MON AND USER
00042   0 12 00001                 43          MIM     CHKSUM
00043   0 02 14000                 44          TRBM
00044   0 40 01000                 45          BRT*
00045   0 01 00044                 46          BRJ     *-1
00046   0 20 00000                 47          NOP                     /PUNCH MON, USER PROGS
00047   0 02 00002                 48          EIR
00050   1 71 00021                 49          ARM     17
00051   0 01 00004                 50          BRJ     3               REMOVE THIS PROGRAM
                                   51  *                                                                  *
00052   2 32 00012                 52  BOOT    WIM     012,2           /
00053   0 41 00052                 53          BRX     2               /
00054   0 71 00011                 54          LDX     011             /
00055   2 32 00000                 55          WIM     0,2             /
00056   0 40 01000                 56          BRT*                    /
00057   0 41 00005                 57          BRX     5               /
00060   0 01 00062                 58          BRJ     LOAD            /
00061   0 00 40062                 59          PZE     *$+1            /BOOTSTRAP LOAD
                                   60  *                                                                  *
00062   0 40 01000                 61  LOAD    BRT*
00063   0 01 00062                 62          BRJ     *-1
00064   0 71 00105                 63          LDX     NEXT
00065   0 46 00003                 64  LD1     CLR                     /
00066   0 02 02604                 65          RPT     0,1,4
00067   2 61 00105                 66          MIN     NEXT,2          /
00070   2 32 40105                 67          WIM     *NEXT,2         /
00071   2 58 40105                 68          ADD     *NEXT,2         /
00072   0 40 01000                 69          BRT*                    /
00073   0 01 00067                 70          BRJ     *-4             /FILL, ACCUMULATE CHECK SUM
00074   0 72 00106                 71          SKA     N1
```

```
00075  0 00 00000        72          HLT                         HALT IF
00076  0 20 00004        73          NRP     4                    CHECK SUM ERROR
00077  0 41 00065        74          BRX     LD1
00100  0 02 00002        75          EIR
00101  1 71 00021        76          ARM     17
00102  0 01 00003        77          BRJ     3
                         78  *                                                    *
00103  00006564          79          DATA    MEMSIZ-01214
00104  00000000          80          DATA    0
00105  77777776          81  NEXT    DATA    -2
00106  77777777          82  N1      DATA    -1
00107  77777743          83  BTSIZE  DATA    300T-9
                         84  ********************************************************************
       00000002          85          END     DUMP
00110  77776566
```

Appendix D

Sample User Programs

PRECEDING PAGE BLANK NOT FILMED.

```
                                                1
                                                2  ********************************************************************************
                                                3  *                                                                              *
                                                4  *       SDS TO TELETYPE TRANSLATOR                                             *
                                                5  *                                                                              *
                                                6  ********************************************************************************
      17400000                                  7  READY   BRD     017400000
00000                                           8          RBRG    0
00000  0 80 00001                               9  TTYOUT  NBR     1
00001  0 76 00247                              10          LDA     =-4
00002  0 35 00124                              11          STA     LETFLG
00003  0 43 00104                              12          BRX     LTR
00004  0 71 00250                              13  LINE    LDX     =-12
00005  2 76 00247                              14  NXTWD   LDA     LIST+12,2
00006  0 37 00125                              15          STX     SAVEX
00007  0 75 00247                              16          LDB     =-4
00010  0 36 00126                              17          STB     COUNT
00011  0 46 00005                              18  NXTCHR  ABC
00012  0 67 00004                              19          LSH     6
00013  0 36 00127                              20          STB     WORD
00014  0 35 00130                              21          STA     TMP
00015  0 71 00130                              22          LDX     TMP
00016  2 76 00133                              23          LDA     CHRTBL,2
00017  0 46 00005                              24          ABC
00020  0 67 00003                              25          LSH     3
00021  0 35 00130                              26          STA     TMP
00022  0 71 00130                              27          LDX     TMP
00023  2 43 00066                              28          BRM     *TYPE,2
00024  0 46 10012                              29          BAC
00025  0 66 00003                              30          RSH     3
00026  0 43 00042                              31          BRM     CHAR
00027  0 61 00126                              32  UPDATE  MIN     COUNT
00030  0 53 00126                              33          SKN     COUNT
00031  0 01 00004                              34          BRU     $+3
00032  0 76 00127                              35          LDA     WORD
00033  0 01 00011                              36          BRU     NXTCHR
00034  0 71 00125                              37          LDX     SAVEX
00035  0 41 00005                              38          BRX     NXTWD
00036  0 43 00060                              39          BRM     CR
00037  0 01 00004                              40          BRU     LINE
                                                41  *                                                                            *
00040  0 00 00000                              42  CHAR    PZE                                                                  *
00041  0 35 00131                              43          STA     OUT
00042  0 36 00130                              44          STB     TMP
00043  1 74 00123                              45          READY   TTYRDY
00044  0 80 00000                              46          NBR
00045  0 75 00130                              47          LDB     TMP
00046  2 02 70010                              48          EAM     *030010,2
00047  0 13 00131                              49          PBT     OUT
00050  2 02 70020                              50          EAM     *030020,2
00051  0 51 00040                              51          BRR     CHAR
                                                52  *                                                                            *
00052  0 00 00000                              53  TAB     PZE
00053  0 71 00251                              54          LDX     =-5
00054  0 76 00252                              55          LDA     =4
00055  0 43 00040                              56          BRM     CHAR
00056  0 41 00054                              57          BRX     $-2
00057  0 51 00052                              58          BRR     TAB
                                                59  *                                                                            *
00060  0 00 00000                              60  CR      PZE
00061  0 76 00253                              61          LDA     =010
00062  0 43 00040                              62          BRM     CHAR
00063  0 76 00254                              63          LDA     =2
00064  0 43 00040                              64          BRM     CHAR
00065  0 51 00060                              65          BRR     CR
                                                66  *
00066                                           67  TYPE    RES     1
00067  0 01 00073                              68          BRU     FIG
00070  0 01 00104                              69          BRU     LTR
00071  0 01 00114                              70          BRU     BOTH
00072  0 01 00114                              71          BRU     SPEC
```

```
                                72  *                                                                      *
00073  0 00 00000               73  FIG     PZE
00074  0 53 00124               74          SKN     LETFLG
00075  0 01 00077               75          BRU     $+2
00076  0 51 00073               76          BRR     FIG
00077  0 76 00255               77          LDA     =27
00100  0 43 00040               78          BRM     CHAR
00101  0 76 00256               79          LDA     =-1
00102  0 35 00124               80          STA     LETFLG
00103  0 51 00073               81          BRR     FIG
                                82  *                                                                      *
00104  0 00 00000               83  LTR     PZE
00105  0 53 00124               84          SKN     LETFLG
00106  0 51 00104               85          BRR     LTR
00107  0 76 00257               86          LDA     =31
00110  0 43 00040               87          BRM     CHAR
00111  0 76 00132               88          LDA     PO
00112  0 35 00124               89          STA     LETFLG
00113  0 51 00104               90          BRR     LTR
                                91  *                                                                      *
00114  0 00 00000               92  BOTH    PZE
00115  0 51 00114               93          BRR     $-1                                                    *
                                94  *
00116  0 00 00000               95  SPEC    PZE
00117  0 66 00003               96          RSH     3
00120  0 36 00130               97          STB     TMP
00121  0 43 40130               98          BRM     *TMP
00122  0 01 00027               99          BRU     UPDATE                                                 *
                               100  *
00123  2 40 70010              101  TTYRDY  SKS     *030010,2
00124                          102  LETFLG  RES     1
00125                          103  SAVEX   RES     1
00126                          104  COUNT   RES     1
00127                          105  WORD    RES     1
00130                          106  TMP     RES     1
00131                          107  OUT     RES     1
00132  00000000                108  PO      DATA    0
                               109          PAGE
                               110  ***************************************************************************
                               111  *                                                                         *
                               112  *       TRANSLATOR TABLE                                                  *
                               113  *                                                                         *
                               114  ***************************************************************************
                               115  T       FORM    3,21                    FIRST FIELD=1(FIGS)
                               116  *                                                   =2(LETTERS)
                               117  *                                                   =3(EITHER LETTER OR FIG)
                               118  *                                                   =4(SPECIAL CHAR)
00133  1 00000026              119  CHRTBL  T       1,026                   0
00134  1 00000027              120          T       1,027                   1
00135  1 00000023              121          T       1,023                   2
00136  1 00000001              122          T       1,1                     3
00137  1 00000012              123          T       1,012                   4
00140  1 00000020              124          T       1,020                   5
00141  1 00000025              125          T       1,025                   6
00142  1 00000007              126          T       1,7                     7
00143  1 00000006              127          T       1,6                     8
00144  1 00000030              128          T       1,030                   9
00145  3 00000004              129          T       3,4                     SPACE
00146  1 00000024              130          T       1,024                   =
00147  1 00000013              131          T       1,013                   '
00150  1 00000016              132          T       1,016                   :
00151  1 00000021              133          T       1,021                   >
00152  1 00000031              134          T       1,031                   )
00153  1 00000032              135          T       1,032                   +
00154  2 00000003              136          T       2,3                     A
00155  2 00000031              137          T       2,031                   B
00156  2 00000016              138          T       2,016                   C
00157  2 00000011              139          T       2,011                   D
00160  2 00000001              140          T       2,1                     E
00161  2 00000015              141          T       2,015                   F
00162  2 00000032              142          T       2,032                   G
00163  2 00000024              143          T       2,024                   H
```

```
00164  2 0000006      144          T       2,6              I
00165  3 0000010      145          T       3,010            BACKSPACE
00166  1 0000034      146          T       1,034            *
00167  1 0000022      147          T       1,022            )
00170  1 0000017      148          T       1,017            (
00171  1 0000021      149          T       1,021            <
00172  1 0000031      150          T       1,031            STOP CODE
00173  1 0000003      151          T       1,3              +
00174  2 0000013      152          T       2,013            J
00175  2 0000017      153          T       2,017            K
00176  2 0000022      154          T       2,022            L
00177  2 0000034      155          T       2,034            M
00200  2 0000014      156          T       2,014            N
00201  2 0000030      157          T       2,030            O
00202  2 0000026      158          T       2,026            P
00203  2 0000027      159          T       2,027            Q
00204  2 0000012      160          T       2,012            R
00205  4 00 00060     161          PZE     CR,4             C/R
00206  1 0000011      162          T       1,011            $
00207  1 0000015      163,         T       1,015            *
00210  1 0000022      164          T       1,022            )
00211  1 0000036      165          T       1,036            ]
00212  1 0000031      166          T       1,031            ^
00213  3 0000004      167          T       3,4              BLANK(060)
00214  1 0000035      168          T       1,035            /
00215  2 0000005      169          T       2,5              S
00216  2 0000020      170          T       2,020            T
00217  2 0000007      171          T       2,7              U
00220  2 0000036      172          T       2,036            V
00221  2 0000023      173          T       2,023            W
00222  2 0000035      174          T       2,035            X
00223  2 0000025      175          T       2,025            Y
00224  2 0000021      176          T       2,021            Z
00225  4 00 00052     177          PZE     TAB,4            TAB
00226  1 0000014      178          T       1,014            ,
00227  1 0000017      179          T       1,017            (
00230  1 0000031      180          T       1,031            CR%
00231  1 0000031      181          T       1,031            \
00232  1 0000031      182          T       1,031            DELETE
00233  63254325       183  $LIST   TEXT    <TELETYPE MESSAGE OUTPUT PROGRAM NOW IN QUEUE**>
00234  63704725
00235  60442562
00236  62212725
00237  60466463
00240  47546360
00241  47514627
00242  51214460
00243  45466460
00244  31456350
00245  64256425
00246  54546060

                      184  ************************************************************************
       00000000       185          END     TTYOUT
00247  77777774
00250  77777764
00251  77777773
00252  00000004
00253  00000010
00254  00000002
00255  00000033
00256  77777777
00257  00000037

                        1
                        2  ************************************************************************
                        3  *                                                                      *
                        4  *       PROGRAM TO READ CARD WHEN PROG 2 READY                         *
                        5  *                                                                      *
                        6  ************************************************************************
00000                   7          PORG    0
       17600000         8  BIO     SPD     017600000
       17400000         9  READY   SPD     017400000
00000  0 20 00001      10  PROG1   NOP     1
```

```
00001  1 74 00014      11          READY   CRTY
00002  0 20 00000      12          NOP
00003  0 46 00003      13          CLR
00004  0 35 00014      14          STA     FLAG
00005  1 74 00015      15          READY   SKIP
00006  1 76 00141      16          BIO     *80
00007  0 02 02704      17          EOM     02706
00010  0 30 00020      18          XIM     TABLE+1
00011  0 20 00000      19          NOP
00012  0 20 00000      20          NOP
00013  0 01 00000      21          BRU     PROG1
00014                  22  *FLAG   RES     1
00015  0 53 00014      23  SKIP    SKN     FLAG
00016  0 40 12106      24  CRTY    SKS     012106
00017  52606060        25  *TABLE  TEXT    < >
00020  63212243        26          TEXT    <TABLE ORIGINALLY FILLED WITH THIS MESSAGE >
00021  25604651
00022  31273145
00023  21434270
00024  60263143
00025  43252460
00026  66316330
00027  60633031
00030  62604425
00031  62622127
00032  25526060
00033                  27          RES     70
       00000000        28          END     PROG1
00141  00000120
                        1
                        2  ********************************************************************
                        3  *                                                                  *
                        4  *       AUXILIARY TYPE OUT PROGRAM                                 *
                        5  *                                                                  *
                        6  ********************************************************************
00000                   7          RORG    0
       17500000         8  PAUSE   OPD     017500000
       17600000         9  BIO     OPD     017600000
00000  0 20 00003      10  PROG3   NOP     3
00001  1 76 00010      11          BIO     N
00002  0 02 02641      12          TYP     0,1,4
00003  0 12 00011      13          MIW     TXT
00004  0 02 14000      14          TOPW
00005  0 20 00000      15          NOP
00006  1 75 00003      16          PAUSE   3
00007  0 01 00000      17          BRU     PROG3
00010  00000014        18  N       DATA    PRGEND-TXT
00011  52633144        19  TXT     TEXT    < TIME-SHARED TYPEWRITER OUTPUT, BPT 3 INITIATED>
00012  25406230
00013  21512524
00014  60637047
00015  25665131
00016  63255160
00017  46646367
00020  64637360
00021  22476360
00022  03603145
00023  31633121
00024  63252460
00025                  20  PRGEND  RES     0
       00000000        21          END     PROG3
                        1
                        2  ********************************************************************
                        3  *                                                                  *
                        4  *       PROGRAM TO TYPE OUT CONTENTS OF CARD                       *
                        5  *                                                                  *
                        6  ********************************************************************
00000                   7          RORG    0
       17600000         8  BIO     OPD     017600000
00000  0 20 00002       9  PROG2   NOP     2
00001  1 76 00011      10          BIO     P13
```

```
00033  0 76 00073      43          LDA     =C40000000
00034  0 35 00070      44          STA     XOUT
00035  0 43 00046      45          BRM     MOVE
00036  0 43 00055      46          BRM     WAIT
00037  0 02 36711      47          PEN
00040  0 76 00066      48          LDA     YDEC
00041  0 55 00064      49          ADD     XDEC
00042  0 35 00066      50          STA     YDEC
00043  0 40 20040      51          BPT     4
00044  0 01 00013      52          BRJ     PL1A
00045  0 01 00020      53          BRJ     PL2
                        54  *                                                    *
00046  0 00 00000      55  MOVE    PZE
00047  0 02 33711      56          PBSX
00050  0 13 00070      57          PBT     XOUT
00051  0 02 35711      58          PBSY
00052  0 13 00071      59          PBT     YOUT
00053  0 02 37611      60          CONVT
00054  0 51 00046      61          BRR     MOVE
                        62  *                                                    *
00055  0 00 00000      63  WAIT    PZE
00056  0 76 00065      64          LDA     NTIME
00057  0 35 00067      65          STA     XSAVE
00060  0 53 00067      66          SKN     XSAVE
00061  0 51 00055      67          BRR     WAIT
00062  0 61 00067      68          MIN     XSAVE
00063  0 01 00060      69          BRJ     *-3
00064  00000400        70  XDEC    DATA    0400
00065  77555410        71  NTIME   DATA    -75000
00066  00002000        72  YDEC    DATA    00002000
00067                  73  XSAVE   RES     1
00070                  74  XOUT    RES     1
00071                  75  YOUT    RES     1
                        76  ************************************************************
       00000000        77          END     PLOT
00072  77777776
00073  40000000
00074  37770000
00002  0 02 02641      11          TYP     0,1,4
00003  0 12 00000      12          MIW     TABLE
00004  0 02 14000      13          TOPW
00005  0 20 00000      14          NOP
00006  0 76 00012      15          LDA     *+1
00007  0 35 00000      16          STA     FLAG
00010  0 01 00000      17          BRJ     PROG2
00011  00000015        18  P13     DATA    13
       00000000        19          END     PROG2
00012  77777777
00003  TABLE
00007  FLAG
                        1
                        2  ************************************************************
                        3  *                                                          *
                        4  *       PLOTTING  PROGRAM                                  *
                        5  *                                                          *
                        6  ************************************************************
       17500000         7  PAUSE   OPD     017500000
       00233711         8  PBSX    OPD     00233711        ALERT X
       00235711         9  PBSY    OPD     00235711        ALERT Y
       00237611        10  CONVT   OPD     00237611        CONVERT X AND Y VALUES
       00237311        11  PENUP   OPD     00237311        RAISE PEN
       00236711        12  PEN     OPD     00236711        LOWER PEN
       00237111        13  CHART   OPD     00237111        ADVANCE CHART
                        14  *                                                    *
00000                   15          PROG    0
00000  0 20 00001       16  PLOT    NOP     1
00001  0 46 30003       17          CLR                     /
00002  0 71 00072       18          LDX     *+2             /
00003  0 35 00070       19  PL1     STA     XOUT            /
00004  0 35 00071       20          STA     YOUT            /
00005  0 37 00067       21          STX     XSAVE
00006  0 43 00046       22          BRM     MOVE            /
```

```
00007  1 75 00004      23          PAUSE   4                   /
00010  0 71 00067      24          LDX     XSAVE               /
00011  0 76 00073      25          LDA     =040000000          /
00012  0 41 00003      26          BRX     PL1                 /CALIBRATE PLOTTER
00013  0 02 37111      27  PL1A    CHART
00014  0 43 00055      28          BRM     WAIT
00015  0 02 36711      29          PEN
00016  0 76 00064      30          LDA     XDEC
00017  0 35 00066      31          STA     YDEC
00020  0 76 00070      32  PL2     LDA     XOUT
00021  0 55 00064      33          ADD     XDEC
00022  0 35 00070      34          STA     XOUT
00023  0 76 00071      35          LDA     YOUT
00024  0 55 00066      36          ADD     YDEC
00025  0 35 00071      37          STA     YOUT
00026  0 43 00046      38          BRM     MOVE
00027  0 76 00070      39          LDA     XOUT
00030  0 73 00074      40          SKG     =037770000
00031  0 01 00020      41          BRJ     PL2
00032  0 02 37311      42          PENUP
```