

IA

**stichting
mathematisch
centrum**



AFDELING INFORMATICA

IW 30/75

FEBRUARI

P.J.W. TEN HAGEN & C.L. PIPPEL

THE UTOR, A PROGRAM FOR THE CONNECTION OF A NUMBER OF
PERIPHERALS TO A GENERAL PURPOSE COMPUTER

547.847

2e boerhaavestraat 49 amsterdam

BIBLIOTHEEK MATHEMATISCH CENTRUM
AMSTERDAM

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS(MOS) subject classification scheme(1970): 68L20

ACM -Computing Reviews- Category: 4.3, 3.8

The UTOR, a program for the connection of a number of peripherals to a general purpose computer

by

P.J.W. ten Hagen & C.L. Pippel.

ABSTRACT

A real time operating system for a satellite computer (in casu a PDP8/I) is described. The satellite is used as a peripheral processor to a general purpose computer. The real time problems inherent to the servicing of all kinds of peripherals are tackled by splitting up the servicing into the urgent real time part and the less urgent follow-up. The interface between these two kinds of service processes is defined by means of meta-instructions for job handling. The design and implementation of these meta-instructions together with some examples is outlined. The complete operating system-assembly listing is added in the appendix.

KEYWORDS & PHRASES: Real time operating system, peripheral processor,
multi programming

CONTENTS

ABSTRACT	i
1. INTRODUCTION	1
2. THE SUPERVISOR	1
2.0. Task and Form of the Supervisor	3
2.1. The Job mechanism	3
2.2. The Meta-instructions	4
3. THE IMPLEMENTATION OF THE JOB MECHANISM	6
4. THE CONNECTION OF TERMINALS	
AN ILLUSTRATIVE EXAMPLE	10
4.0. Introduction	10
4.1. Specifications for the Use of a Terminal	10
4.2. The Use of a Terminal	11
4.2.1. The Interactive Use	11
4.2.2. The Full-Duplex Use	13
4.2.3. The Off-Line Terminal Use	13
4.3. UTOR Commands	14
4.4. The Implementation	16
4.4.1. The Multiplexer Interrupt Routine	16
4.4.2. The PX- and XP Interrupt Routine	17
4.4.2.1. The PX interrupt routine	17
4.4.2.2. The XP interrupt routine	18
4.4.3. The Main Program	18
4.4.3.1. The Jobroutines of the Main Program	18
Appendix A	
THE UTOR PROGRAM	25

1. INTRODUCTION

In this report we describe an operating system for a small satellite computer (a PDP8/I).

The function of the operating system, called UTOR (Users coordinaTOR), is to establish the connection between a number of peripherals and the central general purpose computer (a Philips-Electrologica EL-X8-computer).

There are several reasons why we use a satellite computer to connect these peripherals instead of connecting them directly to the central computer:

1. The number of connections possible to the computer can be (and in our case is) smaller than the total number of connections required for all peripherals. A satellite counts for only two connections.
2. Some peripherals demand the attention of the CPU with high frequency. Putting a satellite in between, which takes care of all real time problems caused by those peripherals, results in a considerable reduction of overhead for the CPU.
3. The satellite can make each peripheral look like a simple standard machine to the central computer.
4. In many cases the connection between a special purpose computer and a peripheral is much cheaper than the hardware for a direct connection. This is especially true when the central computer is replaced. In this latter case only the connection between the new computer and the satellite has to be reestablished

The control of the UTOR over a peripheral can be divided into two parts:

1. The actions that are to be taken in order to fulfil the real time requirements. These actions have to be completed within a certain time-interval each time a device asks attention.
2. All actions that can safely be delayed if necessary. This set of actions is called *the administrative follow-up*.

The operating system has a set of primitive actions, that can split up the servicing of each peripheral into the real time treatment and the administrative follow-up. In this way all peripherals (including the central

computer) are treated uniformly. The real time treatment takes place in interrupt routines. The follow-up consists of a number of jobs, initially created by the interrupt routines. Each job is created by putting it into the queue of its appropriate priority level. The real time treatment as well as the follow-up is subject to a multilevel priority schedule. This solves the problem of parallel servicing of all connected peripherals. The connection of another device causes no further problem with respect to this subject.

Initially the UTOR was meant to connect a number of terminals to the EL-X8. To this end the first version of the UTOR knew a two-level interrupt mechanism to deal with the real time problems, and one low-priority job queue for the follow-up. This version was designed and programmed by the first author of this report. It was presented as a "master thesis" under the title: "DE UTOR, een programma voor de koppeling van een aantal terminals aan een computer". It was carried out under the supervision and mentorship of Prof.Dr. F.E.J. KRUSEMAN ARETZ. The philosophy of how to use a terminal, and the ideas for the implementation of the UTOR were developed during many discussions between the author and his mentor. The author is also much indebted to J.V.M. VAN DER GRINTEN and C. ZUIDEMA from whom he received the first raw versions of the interrupt routines for the multiplexer and the interface with the EL-X8. When new peripherals had to be connected to the UTOR the concept of a multilevel jobmechanism which contains the multilevel interrupt routines was designed together by both authors. This version was implemented by the second author. It is also the version that we describe in this report.

When it was noticed that the UTOR could service all connected peripherals and still remain idly waiting most of the time, the idea arose to modify the UTOR and the PS/8 operating system of the PDP8 in such a way that PS/8-programs could run under supervision of the UTOR. This project was carried out by A.E. BROUWER and the second author. Since the interface between the UTOR and PS/8 was rather complex and all programs to be run had to be modified considerably, the hardware of the PDP8/I was extended to allow disconnecting the keyboard and teletype printer flags from the interrupt bus. The version of the UTOR listed in appendix A is the one with a simplified interface with PS/8, made possible by this hardware change.

2. THE SUPERVISOR

2.0. Task and Form of the Supervisor

The task of the supervisor is to distribute processor time over all peripherals that need servicing for their real time requests or for the administrative follow-up. We assume that all real time requests are initiated by the hardware through a one-level interrupt. (The PDP8/I has a one priority-level interrupt mechanism). The first and most urgent action undertaken by the supervisor is to find out how urgent a given interrupt needs to be serviced. This means that all UTOR routines are interruptable, except for some short (in time!) sections. All critical actions of the cooperating processes take place inside these protected sections. A call of the interrupt service routine always results in the creation of a *job* of a certain priority. Upon exit of the interrupt routine, the job with the highest priority is executed or continued.

The supervisor consists of the interrupt routine mentioned above, and a number of other routines that can be called from the specific peripheral service routines. These calls are referred to as *meta-instructions*. The meta-instructions can be added to the repertoire of machine instructions (in our case PAL8 assembly code). This creates a new machine that, in our experience, is much easier to deal with. Before describing the meta-instructions in detail, we shall give a complete job description in the next paragraph.

2.1. The Job mechanism

A job is an instruction to the supervisor to execute a certain routine with some specified data. The job can result in a change of the given data as well as in the creation of zero or more subsequent jobs.

The execution of a job is an autonomous process. A job can be temporarily interrupted but never be terminated by the supervisor. Termination is accomplished by the job itself through execution of a meta-instruction.

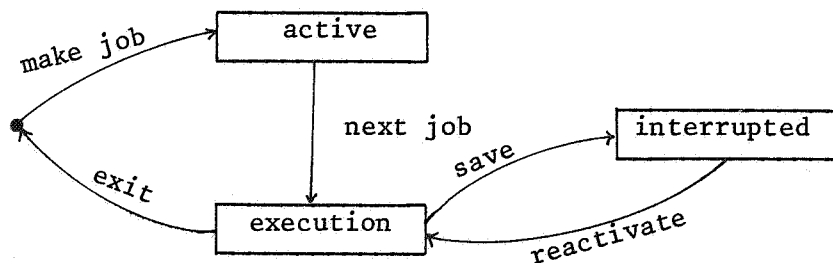
A job, just created, is specified by the following three parameters:

- the routine address
- the priority
- the data pointer.

The data pointer is the core memory address where the data can be found and stored. Two existing jobs may differ in the data pointer only! The priority is a measure for the urgency of a job. For each priority level there exists one queue of jobs. Only the first job of the queue is in execution. New jobs are always put at the end of the appropriate queue. Thus it follows that for each priority level only one job at a time can be in execution. In this way a job may have local variables as long as all incarnations of the same subroutine are of equal priority. Furthermore it follows that jobs of equal priority are executed in order of creation. A job once created can be in one of the following states:

- active, the job is in the queue
- in execution, the routine of the job is in execution
- interrupted, the job is saved on the interrupt stack.

The state of a job changes through execution of one of the meta-instructions as specified in the following transition diagram:



2.2. The Meta-instructions

The meta-instructions can be added to the program in the form of subroutine calls. We specify them below:

- MAKEJOB

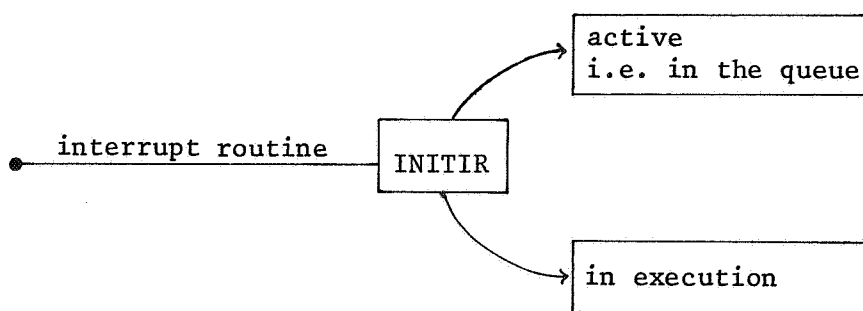
This is a call to the subroutine MAKEJB which creates the job specified by three parameters:

routine address (in accumulator)
 job status word (in JSWMJ)
 data pointer (in DTPNMJ)

The parameters are stored at fixed locations. Therefore, the interrupt must be disabled before storing, and may be enabled only after the call.

- INITIR

This is a call to the subroutine INTRAP by an interrupt routine, to ask for processor time. The accumulator contains the only parameter, the priority level. INTRAP decides whether the interrupted job must be continued or must be saved because of the higher priority of the interrupting routine. In the former case it executes MAKEJOB (calling address+1, priority, skip), followed by a return to the interrupted job; in the latter it executes SAVE, and continues the interrupt routine



- EXIT

This is a jump to DECIS, executed by a job when it has completed its activities. DECIS then either continues the last interrupted job or starts a job from the most urgent nonempty job queue, whichever has the highest priority.

- SAVE

This is a call to the subroutine STACK, which saves accumulator, link, data field, instruction field, program counter, data pointer, and job status word of the interrupted job on the interrupt stack.

The job status word contains the priority of the interrupted job. After execution of SAVE it can be found at the top of the stack. This makes the selection of the most urgent job by DECIS more easy.

- REACTIVATE

This is a call to the subroutine UNSTACK to restore registers and variables saved by STACK. In particular the program counter is restored, which means that the interrupted program is continued.

- NEXTJOB

Called by DECIS to start the next job from the job chain (if its priority is greater than that of the interrupt stack). In the listing it is represented by the label NEXTJB.

3. THE IMPLEMENTATION OF THE JOBMECHANISM

In this chapter we will describe in more detail how the jobmechanism is implemented. Illustrations are taken from the existing implementations on the PDP8/I. All terms used in this description, that directly refer to PDP8/I-hardware or the assembly listing (appendix A), will be denoted in capital letters (e.g., AC, LINK or L). For their precise definition, the reader is referred to the PDP8/I programming manual.

The status of a job can be defined as the way the specifying parameters are stored. These parameters contain all information to (re)start the job. The parameters are:

- routine address, consisting of PC (program counter) and
IF (instruction field)
- priority, 3-bits register, thus allowing 7 levels of priority (at the moment only 4 levels are used)
- data pointer, actual data pointer
- registers, AC (accumulator), 12-bits and L (link), 1-bit.

According to the transition diagram (c.f. 2.1) a job can be in one of three states:

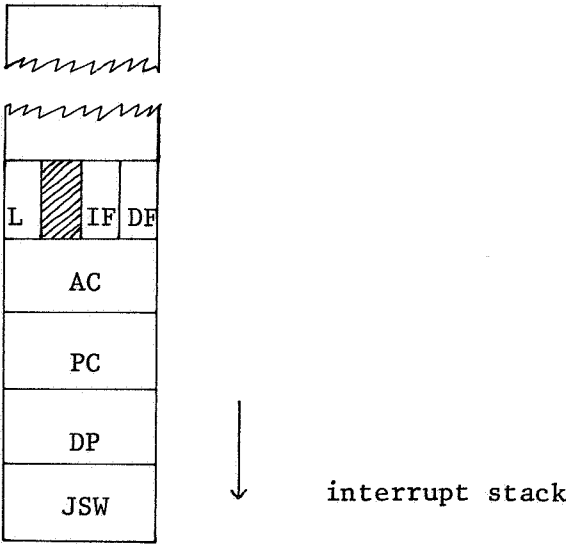
- in execution

The parameters denoted in capital letters are stored in the corresponding hardware registers. The "priority" and "actual data pointer" can be found in two memory locations named JSW and DATAPN respectively.

- interrupted

The status of the interrupted job is saved on the interrupt stack, as depicted in fig. 3.1. The JWS of the job with highest priority among all interrupted jobs containing this priority lies on the top of the stack. The priority can be considered also as the priority of the stack as a whole. (Note that the priority of job in execution is always higher than that of the stack; therefore, if such a job is interrupted it is pushed on top of the stack).

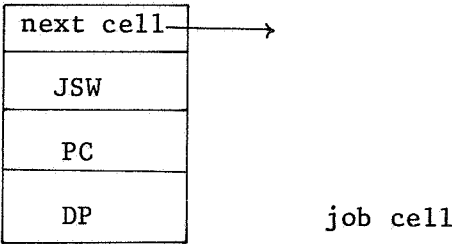
fig. 3.1.



- active

The active job status fills a cell in the job chain. This cell contains (cf. fig. 3.2) only the job status word, the PC and the data pointer. All other hardware registers will be initialized in a standard way [i.e., $AC := \emptyset$; $L := \emptyset$; $IF := \emptyset$, $DF := 3$], when execution is started (NEXTJOB) and need not to be stored.

fig. 3.2.



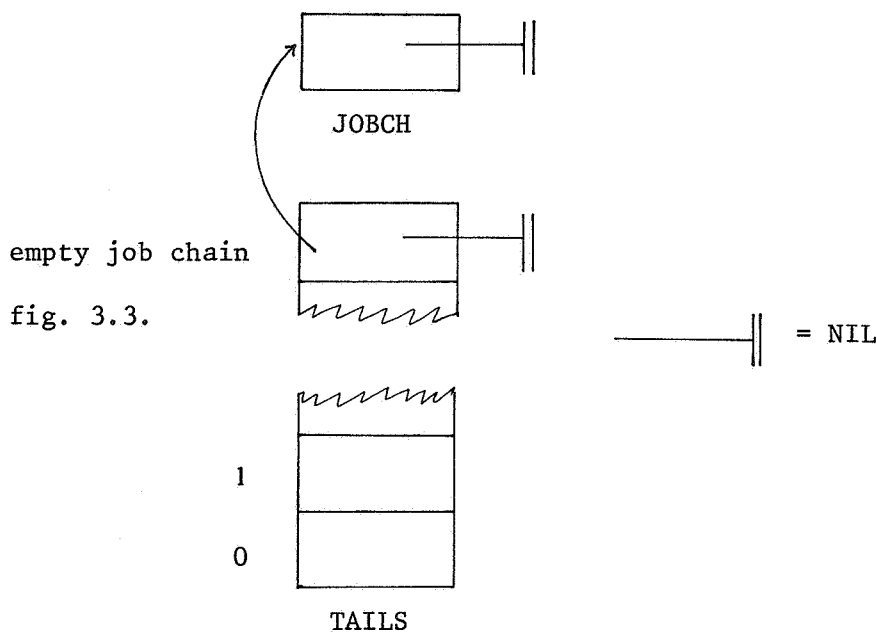
The so called job chain consists of several sub-chains, i.e., one for each priority level. A sub-chain of higher priority precedes a sub-chain of lower priority. The elements of the sub-chain are the job cells (fig. 3.2). The NEXT-pointer of the last cell in a sub-chain points to the first element of the next nonempty sub-chain, or to NIL if no such sub-chain exists. In the latter case it is also the last cell of the job chain.

The first cell in the job chain can be found through the pointer JOBCH. This pointer is used by NEXTJOB to find the active job that has to be brought in execution.

MAKEJOB puts a job of priority i at the end of the i -th sub-chain. In order to locate this position without a search of undefined length the supervisor disposes of an array of pointers, namely:

TAILS[0: N] (N is the highest priority level).

TAILS[i] points to the last active job of priority i , or to NIL when the sub-chain is empty; however, TAILS[N] points to JOBCH when the job chain is empty. JOBCH can be considered as the NEXT-field of the "job cell" that precedes the first existing job in the chain. The empty chain is depicted in fig. 3.3. Observe that the first defined pointer from TAILS [i] locates the position where an active job of priority i has to be inserted. This is also the case when the job chain is empty: The fact that a (sub-) chain is empty constitutes no exceptional situation.



After one action of MAKEJOB for priority i , the empty chain turns into the chain of fig. 3.4. The reverse effect would be caused by an action of NEXTJOB. It is clear that both meta-instructions carry out some pointer setting.

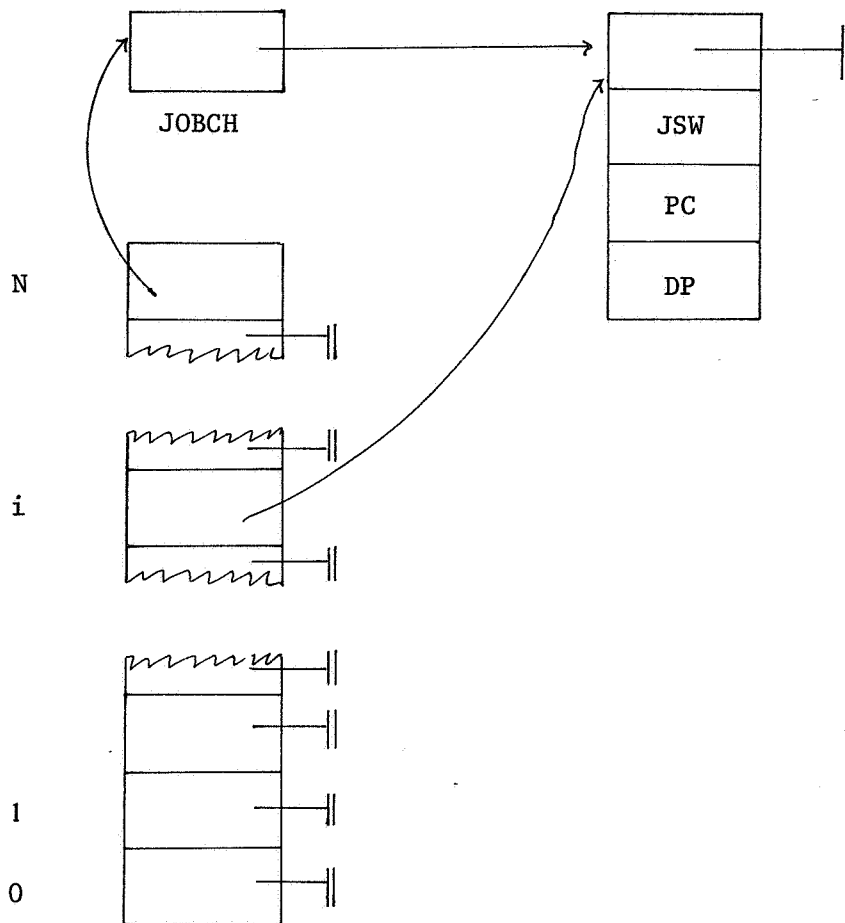


fig. 3.4.

The supervisor contains a routine called DECIS which determines whether a job from the stack will be restarted or a new job from the job chain will be executed, i.e., whether a REACTIVATE or a NEXTJOB will be executed. DECIS chooses REACTIVATE whenever the priority of the stack is equal to or greater than the priority of the job chain, or when the job chain is empty. In the latter case one might expect stack underflow. Stack underflow will not occur as long as the following condition remains valid: DECIS is executed only following an execution of the EXIT-instruction by an active job. This job has interrupted another job, or the background program, which never terminates, and always is started first (upon initialization of the whole system). Hence in both cases the stack is not empty. Observe that stack overflow will not occur due to the fact that a job can only be interrupted by a job of higher priority. There can be at most $N+1$ jobs interrupted. ($N = \text{max priority}$). So the stack will never grow over $(N+1)*5$ locations.

4. THE CONNECTION OF TERMINALS, AN ILLUSTRATIVE EXAMPLE

4.0. Introduction

To illustrate the programming of a peripheral in the UTOR, we describe in this chapter the connection of a number of terminals. In the first part a set of specifications for the use of terminals is justified. This leads to a number of rules for the information traffic between a user at the terminal and the central computer. In the second part (cf. section 4.4) the primitive actions defined in chapter 2 are applied to solve the real time problems as well as to implement these rules.

4.1. Specifications for the Use of a Terminal

The operating system in the central computer communicates with a user at a terminal through messages. To specify the way the UTOR processes these messages we don't need any knowledge about the task of the system. However, a little has to be known about its behaviour. The UTOR accepts messages from the central system for each terminal. These messages are treated in three stages. In the first stage the message is collected in a buffer and the UTOR requests permission to print it. In the second stage, when the permission is granted, the message is handed over to the multiplexer as a character string. In the third stage the multiplexer sends each character as a bitstream to the terminal.

In the opposite direction the multiplexer transforms the bitstream into characters. This constitutes the third stage. Each character is given the appropriate treatment in the second stage. This means that it might be used to build up a message for the central system. Complete messages are handed over to the central system in the first stage.

At each stage, the total number of work is divided into a number of jobs of equal priority. On the third stage the priority is high due to the real time requirements: the multiplexer must sample (put) the next bit from (on) the line within a relatively small interval of time. The priority level for the interface with the central computer is chosen below that for the multiplexer, though high enough to guarantee a high-speed message transfer. A relatively low priority level remains for the second stage jobs.

The treatment of the messages at the second stage is decisive for the

way in which both user at the terminal and central system will communicate with each other. In the next section we will explain how the user can and should be able to use the terminal.

4.2. The Use of a Terminal

All kinds of on-line terminal use can be put into one of the following categories:

- interactive use.

The user and the central system exchange (usually) short messages on question-and-answer base. This application is called the *sin/sout* (simple-in/simple-out) use.

- The use of a terminal as reader station or as output station.

Possibly large amounts of data are read or printed, without intervention of the receiving station (user or central system). These applications will be referred to as the *full-duplex use*. In the following discussion we will only consider so-called *full-duplex terminals*. This means that the decision whether a character received from the keyboard will be printed or not (the so-called *echo*) is fully programmable. Full-duplex terminals allow among other things simultaneous input from keyboard/papertape reader and output to printer/papertape punch.

4.2.1. The Interactive Use

During the conversation between the user and the central system, the user can be in one of the following situations:

- a) he is typing a message,
- b) he is waiting for a message,
- c) the message is being printed.

Situation a) changes into situation b) as soon as the user finishes his message by typing a line-feed character. When the central system detects the line-feed, it starts processing the message. By choosing the line-feed as the completer, we define messages to consist of one line of text. This seems to be most convenient to the user.

The UTOR keeps account of these situations mentioned above. This is

necessary because the printer can be at the disposal of only one of the participants in the conversation at the time. The decision to whom the printer is assigned is made by the UTOR. When the user is allowed to type a message he also has the printer at his disposal, so that he can see what he types. As soon as he completes his message the UTOR puts the terminal in the waiting state (b). In the waiting state both participants can request the printer for one complete line of text. It is assigned to the first applicant. The user requests the printer by typing a character. The central system requests the printer by sending the first buffer of the line to the UTOR. Such a request can be made while the other participant is still printing. In that situation the UTOR assures that this request will be granted when the end of line occurs.

If the central system is a correct participant in the conversation it behaves as follows: it will not request the printer before it has a complete message ready. This guarantees the UTOR, that as soon as it assigns the printer to the system, there will be an opportunity to give the printer back to the user, within the amount of time needed to print one line (e.g., the central system cannot delay a user interrupt by printing an incomplete message). If the line consists of more than one buffer, the central system will send only one buffer at the time. As soon as this buffer gets empty the UTOR will make a request for the next one. This is necessary because the satellite computer has only a small buffer capacity. In practice the central system simply sends the first buffer of a message to the UTOR without concern about the turn. Sooner or later it will receive a request for the next buffer, which implies that it has got the disposition of the printer. The user on the other hand should be concerned about who's turn it is. If he does not finish a message properly (with a line-feed), he can hold up the progress of the conversation. We allow this situation because it causes no serious difficulties. The user can see what he is doing; he only frustrates the progress of his own program and, moreover, he might be holding it up deliberately, because he needs to think, or because of paper jam.

When the user types in the first character of his message one of two possible things happens:

- his character is printed, he got the turn,
- the terminal prints a message from the central system.

This printing stops at the end of the line. The user now can restart typing the first character successfully.

The UTOR accentuates the origin of the typed messages by printing them in different colours.

4.2.2. The Full-Duplex Use

In the full-duplex state, the printer is exclusively assigned to the central system. Every character read in or typed by the user is transmitted to the central system without echo. This allows simultaneous input and output. The interval buffer-organisation maintained by the UTOR can be the same as in the interactive case:

- one output buffer per terminal on request,
- transport of every filled input buffer to the central system.

Sofar nothing has been assumed about the capacity of the input channel from the UTOR to the central system. Since the satellite only has a small buffer capacity, each input buffer must be transferred to the central system, before the next buffer is filled. The channel allows transport of one buffer for one peripheral at the time. Therefore, regardless of the capacity of the channel, the UTOR should provide for a queue mechanism. The capacity of the queue must be large enough to contain all buffers that possibly can get filled from all peripherals during the time interval the system is allowed to ignore the channel (= the maximal transporting time for one buffer).

4.2.3. The Off-Line Terminal Use

The UTOR provides another facility for the terminal user. Until now we have assumed that the user was connected, via the UTOR, with the central system. In the off-line situation, there is no connection between the user and the system, but only between the user, the UTOR and possibly other terminals.

Although this type of use is of minor importance (e.g., the preparation of papertape, copy a tape from one terminal to another), it is included because it takes almost no extra effort to implement it. As a consequence of

the possibility to change connections there must be an initial state, the so-called *solo-state*. In this state the user may request the UTOR to establish a connection; the UTOR also may connect the terminal with the central system or another terminal on their request. The solo-state is simulated by the UTOR, by printing every typed character. Therefore, the user can use this state to prepare his papertapes, and at the same time remain connectable with other stations.

The connections between two terminals is similar to the sin/sout-connection between user and central system (cf. 4.3). The rules concerning the turnover are somewhat different, in order to meet the fact that now both partners can behave badly (i.e., not finish their line of text). In this situation the sout-partner (passive) can give a command, which puts him in the sin-status immediately.

4.3. UTOR Commands

The UTOR supports a certain type of terminal use by giving each received character the appropriate treatment. In order to inform the UTOR that a certain connection or treatment is to be established, the user sends a *command* directly to the UTOR. A command is announced by typing the *attention signal*: @. Upon receipt of this character the UTOR puts the terminal into the command status. The user now can type in the specific command, and effectuates it by typing a line-feed.

A necessary condition for proper use of a terminal is, that both partners are permanently informed about the status of their terminal. For the terminal user it is also important to be able to reconstruct the terminal session from the listing on his printer.

A UTOR command is carried out in two stages:

- a) The typing of the command.
- b) The execution of the command by the UTOR.

Stage a) begins when the UTOR encounters the attention sign and ends when the line-feed is received. In stage b) the UTOR first informs the two stations involved, of the new state and next effectuates it. It is clear that now real time problems can occur: Both partners simultaneously can start typing a command, one of the two might continue typing, unaware of the change of state coming forward, etc. Moreover, the UTOR itself now

becomes interested in the terminal printer.

In order to solve this type of real time problems, we first consider commands to change the type of use (e.g., switch from solo to sin/sout, switch from sin/sout to full-duplex). Only the following commands are allowed:

- the terminal is in sin/sout state and wants to change to a different application,
- the terminal is not in sin/sout state and wants to return to the sin/sout status.

It follows that a change of use always takes place via the sin/sout status. This guarantees that before a new type of use and connection is established, there has been an opportunity to carry on the conversation that is needed. Thanks to the sin/sout-connection this conversation is completely listed. The user in situation a) who wants to give a UTOR command needs to be in sin/sout status. As soon as he has the turn, he types a command, beginning with the attention sign. The command will be executed, as soon as he types the line-feed. Instead of a turnover, the new state is established. Incorrect commands cause an error message from the UTOR, followed by the reestablishment of the sin/sout status. The partner cannot disturb during the typing of a command. He remains in sin/sout status (waiting for a turnover), until the command is executed. During the printing of the UTOR message, that announces a new status, both partners are treated as if they are waiting for the turn (sout status).

In situation b) we must decide between the solo-status and the full-duplex status. In the former case the UTOR message consists of the name of the partner (terminal number or central system) the user wants to be connected with. Upon effectuation the UTOR checks, whether the partner is free (terminal must be in solo-status, the central system always is free). During the full-duplex status the user cannot see what he types (only his partner can!). Therefore, commands must be short to avoid errors. The command level also can be used to inform the system about the current input and output (e.g., stop and continue output in case of paperbreak). The latter type of commands are simply passed on to the system in a marked buffer. UTOR commands originating from the central system are received "at once" in one buffer. This solves the real time problem mentioned above. The command can be effectuated if the terminal is (still) in the right status or otherwise refused.

After a command the system waits for a UTOR reply that reports execution or refusal. A refusal can be received under the following circumstances.

- a) The user started a command somewhat earlier. The refusal can be preceded by a report of the execution of a user command.
- b) The terminal is not in the expected status. The refusal might be preceded by a report of a change of status.
- c) In all other cases the command must have been illegal.

4.4. The Implementation

The three stages mentioned in section 4.1 are found back as three program sections:

4.4.1. The Multiplexer Interrupt Routine

The multiplexer routine can be described by the following program scheme:

```

MP: for each terminal do
    begin sample input line;
        if character completed then
            begin if character = attention sign
                then MAKEJOB (attention, prior 1, line)
                else MAKEJOB (investigate, prior 1, line);
            reset line status
        end
    end;

    for each terminal do
        if output then
            begin send next bit;
                if character completed then
                    begin if buffer empty then
                        begin if  $\neg$  echomode then
                            MAKEJOB (report empty, prior 1, line)
                            output := false
                        end
                    end
                end
            end end;
        wait for clock interrupt;
        goto MP;

```

This cycling program is executed on the highest priority level. During the *wait for clock interrupt*, programs of lower priority may run. The communication with other program sections is established exclusively by means of the instruction MAKEJOB. These jobs specify the administrative follow-up, executed in the second stage, the so-called main program section (or main program). The boolean variable *output* is set true by the main program, each time a character or a buffer is handed over for printing.

4.4.2. The PX- and XP interrupt routines

The central system is seen by the UTOR as two peripherals, namely the PX apparatus for data transport from PDP8/I to central site, and the XP apparatus for data transport in the opposite direction.

4.4.2.1. The PX interrupt routine

Each time a PX transport is completed, a PX interrupt occurs. The PX interrupt routine can be described as follows:

```

PX done: if PXqueuelength > 0 then
    begin take next PX buffer from queue;
        send buffer;
        PXqueuelength := PXqueuelength - 1
    end;
    wait for next PX interrupt;
    goto PX done;

```

The execution of *send buffer* takes place partially with the interrupt disabled. The busy-waiting for the ready flag after each buffer word takes less time than a return from interrupt entry interrupt sequence. In this case waiting with interrupt disabled reduces overhead. The instruction *wait for next PX interrupt* allows all other processes to run. The interrupt is caused by the last ready flag of a *send buffer* action. As long as $PXqueuelength \geq 1$, *send buffer* will be executed in the PX interrupt routine. When the PX queue is empty, *send buffer* will be called from a job routine that puts buffers in the PX queue:

```

PX start: if PXqueuelength =  $\emptyset$  then send buffer else
           begin put buffer in PXqueue;
              PXqueuelength := PXqueuelength + 1
           end;

```

4.4.2.2. The XP interrupt routine

Each first word of an XP transport causes an XP interrupt. Then the following sequence of instructions is executed:

```

XP int: Sample terminal number;
        sample data;
        MAKEJOB (XPbuf, prior 1, terminal number);
        wait for next XP interrupt
        goto XP int;

```

For the same reason as in the case of the PX send buffer, the sampling of each word of the transport contains a busy-waiting loop, with the interrupt disabled. Since for each terminal only one unique XP buffer exists (cf. section 4.1) there is no queuing mechanism needed other than the normal MAKEJOB - NEXTJOB sequence. After sampling the terminal number, the XP routine can select the appropriate XP buffer to store the sampled data in.

4.4.3. The Main Program

The main program consists of a set of job routines. Some of these routines we have already encountered in the previous sections. They specify the way the interrupt routines communicate with the main program. We will now first give a short description of each of the job routines and next give a program scheme for the main program, that shows the conditions for creation of a job. This scheme together with the effect of each job on the terminal administration defines the implementation of the communication rules of section 4.1 - 4.2.

4.4.3.1. The Job routines of the Main Program

The variables that are controlled by the job routines constitute the terminal administration. For each terminal there exists such a set. Which

set is currently worked upon is defined by the variable *line*. This variable is used as an index to the following lists. Each list contains one element per terminal.

1. *Status*; the status word indicates the operational state in which the terminal is working (e.g., solo, sin, sout, duplex, etc.) (cf. 4.2)). It also indicates any transition states between two operational states.
2. *Partner*; if defined, tells with whom the terminal is connected (note: in solo-state partner is undefined).
3. *Message*; a buffer, to collect a UTOR message from the terminal.
4. *Lastchar*; specifies the last received character.
5. *XP buffer status*; specifies whether or not there is an XP buffer present, and if so, whether or not it contains a closing symbol. This information is essential for the turnover decision.
6. *XP buffer pointer* and *-counter*.
7. *PX buffer pointer* and *-counter*.

Each job in the main program is of the same priority. For each priority level only one job at the time is in execution. These rules provide the mutual exclusion of jobs that happen to use the same administration. For descriptive reasons the parameter list of the routine MAKEJOB is of variable length. In this way we can specify all information of interest. It is understood that all parameters except the routine address and the priority ought to be stored under the data pointer.

The job routines:

1. *INVESTIGATE (TSNUF)*:

```

if state = solo then MAKEJOB (PRINT, prior 1, line, char) else
  if state = message then
    begin if char = line feed then react on message else
      begin store message part;
        MAKEJOB (PRINT, prior 1, line, char)
      end
    end else
  if state = sin then
    begin MAKEJOB (PRINT, prior 1, line, char);

```

```

    if partner = system then store char in PXbuffer
    else MAKEJOB (PRINT, prior 1, partner, char);
    if char = line feed then check on turnover
end else
if state = sout then check on turnover else
if state = duplex then store char in PXbuffer

```

prior 1 is the priority of the main program.

The routine *store char in PXbuffer* creates a job in case the buffer gets filled, and asks a new buffer from the PX buffer pool. The job in question is created by: MAKEJOB (PX OVER prior 1, line, PXbuffer, PXcounter).

The routines *react on message* and *check on turnover* will be discussed later.

2. ATTENTION:

```

if state = solo then
  begin MAKEJOB (PRINT, prior 1, line, char):
    state := message;
    initialise administration
  end else
if state = message then
  MAKEJOB (INVESTIGATE, prior 1, line, char) else
if state = sin then
  begin if partner = system then
    begin state := message;
      MAKEJOB (PRINT, prior 1, char, line)
    end else
    begin inform (line, "stop", solo);
      inform (partner, "stop", solo)
    end
  if state = sout then
  begin if partner = system then check on turnover
    else force turnover
  end else
if state = duplex then

```

```

begin close PXbuffer;
      MAKEJOB (PX OVER, prior 1, line PXbuffer, PXcounter);
      next PXbuffer; mark PXbuffer
end .

```

The routine *initialise administration* brings the administration in order for a disconnected terminal, e.g.: partner, last char and message buffer are set undefined; there is no PX buffer; the XP buffer is empty.

The routine *inform (line, "message", future state)* selects a buffer that contains the (fixed) message and prints it:

```

MAKEJOB (PRINT, prior 1, line, buffer pointer, buffer counter);
state := future state + transition.

```

The future state is an operational state that will be established when the printing is finished.

The routine *mark PX buffer* sets a flag in the PX buffer indicating that it contains a message instead of data. This should be noticed by the central system on arrival.

3. PRINT:

```

if single char then
  begin extend MPqueue (line, char); echo := true end
else
  begin fill MPqueue (XPbuffer pointer, XPbuffer counter, line);
    echo := false
  end; output := true

```

The boolean *single char* which is passed on to the multiplexer as a parameter, indicates that the terminal is in the process of echoing typed characters. If this is not the case, multiplexer is obliged to report (by creating a job!) that the printing is done (cf. 4.4.1).

4. TAKE XP BUFFER:

```

if message in buffer then
  begin if state = solo then connect (line, system) else
    if partner  $\neq$  system then refuse message else
      if message number =  $\emptyset$  then generate runoff else
        if state = sin  $\vee$  state = sout then react on message else
          if state = duplex then inform (line, "sin/sout", sout)
          else refuse message
        end else
      begin if state = sin  $\vee$  state = sout  $\vee$  state = duplex then
        begin define XP status; convert buffer;
          if XP counter  $\neq \emptyset$  then
            begin if state = sin then check on turnover else
              MAKEJOB (PRINT, prior 1, line, XPbuffer, XPcounter)
            end else
              MAKEJOB (REQUXP, prior 1, line)
            end else MAKEJOB (REQUXP, prior 1, line)
          end .
        end .
      end .
    end .
  end .

```

Obviously an XP buffer can contain a message or data to be printed. If the data buffer is empty (after conversion) or received in a wrong situation, it is skipped by reporting the XP buffer empty again (REQUXP).

In a non sin/sout-state only two messages are accepted: 1. generate runoff, by filling the XP buffer as specified. This buffer is then treated as data. 2. Put the terminal in sin/sout-state.

In the sin/sout-state several messages are possible. The routine *react on message* ends with the following statement:

```

if message correct then
  begin inform (line, "some text", future state);
    affirm message
  end else refuse message.

```

refuse message never changes any state. It just sends a special buffer to the system, that reports the refusal.

affirm message sends in a similar way an affirmative message to the system.

5. *REQUXP: MAKEJOB (PX OVER, prior 1, -line).*

Sending a negative terminal number to the system in a buffer without data, indicates that the XP-buffer is empty again.

6. *PX OVER:*

assure PX active; next PX buffer.

The routine *assure PX active* executes the statements listed in the PX section after the label *PX start* (cf. 4.4.2.1).

7. *REPORT READY:*

lower transition state;

if partner = system then

begin MAKEJOB (REQUXP, prior 1, line);

if XP buffer status = last buffer of line

then check on turnover;

XP buffer status := buffer not present + XP buffer status

end .

The routine *lower transition state* tries to finish the transition state in favor of the future state. The attempt will be successful if there is no terminal partner or the partner has tried previously to enter the future state. This assures that the printing of the message on both terminals has finished.

The seven jobroutines together with some general purpose subroutines constitute the main program.

The interesting two general purpose routines to discuss are:

react on message:

if partner = system then

begin deduce desired state;

if desired state = duplex then

begin inform (line, "duplex", transition + duplex);

affirm message

end else refuse message

end else

```

begin deduce partner;
  if partner = system then
    begin inform (line, "sin", transition + sin);
      affirm message
    end else
    if partner = terminal then
      begin inform (line, "sin", transition + sin);
        if partner  $\neq$  line then
          inform (partner, "hear", transition + sout)
        end else inform (line, "??", transition + solo)
      end .

```

react on message first interprets the message and next composes the reaction. In situations where only one reaction is possible (e.g., solo, $\text{sin} \wedge \text{partner} \neq \text{system}$), *react on message* is not called but the unique reaction is created instead.

```

check on turnover:
  if partner = system then
    begin if XP buffer present then
      begin if  $\neg$  last time deleted then
        begin state := sout;
          MAKEJOB (PRINT, prior 1, line, XP buffer, XP counter)
        end end end else
        if last char of partner = lf then
          begin state := sin, state of partner := sout end

```

The routine *check on turnover* is called each time the active participant in the conversation (e.g., the sin-partner) reaches a situation that allows turnover, and each time a request for the turn is made by the sout-partner. If both conditions meet, the turnover is effectuated.

APPENDIX A

THE UTOR PROGRAM

As noted in the introduction, the UTOR and the PS/8 operating system have been adapted to enable running of PS/8-programs under supervision of the UTOR. Since PS/8 is not designed to run with interrupt enabled it is full of waiting-loops like

```
KSF
```

```
JMP.-1
```

and

```
TSF
```

```
JMP.-1
```

Also many PS/8-programs use PAGE 0 of FIELD 0 which has to be reserved for the UTOR since each interrupt starts the routine starting at 0001. Therefore, two types of changes were necessary:

1. change systematically all references to FIELD 0 into references to FIELD 2 (not only CDF- and CIF-instructions, but also the memory field in disc- and DEctape-transfers).
2. change sequences like:

```
KSF          TLS
JMP.-1      and  TSF
KRB          JMP.-1
```

into:

```
CIF 0          IOF
JMS I (KWAIT  and  CIF 0
KRB          TLS
              JMS I (TWAIT,
```

where KWAIT and TWAIT are routines in the UTOR which save their calling address and field and execute EXIT.

On the first subsequent printer- or keyboard-interrupt this address is retrieved and PS/8 continued. Apart from these there were many incidental difficulties such as:

- the fact that the WORD COUNT- and CURRENT ADDRESS locations of the disc and DECTape are in FIELD 0 and cannot be shifted to FIELD 2,
- the fact that use of DECTape poses some real time requirements which cannot always be met,
- the use of KSF to test for a typed ^C (attention character of PS/8) where we don't want to use JMS KWAIT, or in contexts like:

```

WT, KSF
    SKP
    JMP A
    SRF
    JMP WT

```

A, ... where we wait for a keyboard or a display flag, whichever comes first,

- the fact that there was not always room to replace the (3-instruction) typing sequence by the (5-location) call to the UTOR,
- the fact that PS/8 system programs are not free from bugs, while it is absolutely necessary not to blow up the UTOR.

To solve some of these problems the hardware of the PDP8/I has been extended with keyboard interrupt enable, printer-interrupt enable and display interrupt enable flipflops. This resulted in the much smaller interface half of which can be seen in the listing.

Of course, the only way to handle these things clearly, without having to patch every system program before using it, is to use the time-sharing option, and this is the final solution adapted.

The adaption of the timesharing option has lead to a completely new implementation of the operating system. The description of this operating system falls outside the scope of this report. The design philosophy for this final version however, is totally different from the one described in this report.

In order to make reading of the program listing more easy, we now give an index of the most important sections and routines. The addresses refer to the machine addresses found in the listing.

Field 0,	page 0	global variables
		subroutine addresses
	page 1	general purpose routines for the main program

page 2 take over a character from the Multiplexer
 page 3 start of jobroutine section:
 -TSNUF, -ATTENTION
 page 5 -TKXPBU
 -REQUXP
 page 6 state transition routine
 page 7 PX interrupt
 -PX OVER
 page 10 XP interrupt
 INITIR (meta-instruction)
 INITJB (" ")
 page 11 Multiplexer interrupt
 -PRINT, -REPORT READY
 page 12 PX echo
 page 13 initialise UTOR
 page 14 PXXP-reset interrupt
 buffer pool manager
 page 15 turnover
 send PX buffer
 page 20 Dynamic storage allocation
 page 21 meta-instructions:
 DECIS
 NEXTJB
 MAKEJB
 page 22 REACTIVATE
 primary interrupt routine (skip chain)
 page 23 SAVE
 CANCEL
 page 24 interrupt handlers:
 plotter
 command teletype
 disk
 high speed punch
 high speed reader

	KVØ8 visual display
	KVØ8 interrupt button
page 26	Interface with PS/8
FIELD 3	conversion tables
	buffer space (buddies)

The program listing is followed by a cross reference table.

/ UTOR DD. 12/3/73

PALB 9/17/74 PAGE 1

1
2

/ UTOR DD. 12/3/73
XLIST

/OPDEF

```

3          / UTOR DD. 12/3/73
4
5          /POINTERS IN FIELD 3
6          HDADSP= 0          /!!!/DYNAMIC STORAGE ALLOCATION
7          HDMASK= HDADSP      /IDEM
8          HDFSP= 20          /IDEM
9          TAILS= 30          /JOB CHAIN
10         HDMS= 763          /MESSAGES
11         HDCHTB= 300        /CHARACTER TABLE
12         HDTRB= 760         /SPECIAL SYMBOLS
13         HDAVAIL=1400       /AVAIL LIST; 400 WORDS
14
15         /IF ONE OF THE FOLLOWING TAGS IS CHANGED,
16         / OTHER PROGRAMS MUST BE CHANGED TOO
17         XLINE= 52
18         LISTEN= 54
19         TKOVR2= 400
20         /
21
22         *0
23         0000 0000 ZERO, 0
24         0001 5577 JMP I [INSPEC /SKIP CHAIN
25
26         0002 0000 X8, 0
27         0003 0000 MX8, 0
28
29         *4
30         /SCRATCH REGISTERS
31         0004 0000 SV, 0
32         0005 0000 SV1, 0
33         0006 0000 SV2, 0
34         0007 0000 SV3, 0
35
36         /AUTO INDEX REGISTERS
37         0010 0000 AUTIN1, 0 /USED BY TKXPBU
38         0011 0000 AUTIN2, 0 /IDEM
39         0012 0200 AUTIN3, JOECH /POINTS TO THE JOB-CHAIN
40         0013 0000 AUTIN4, 0 /SPN
41         0014 0000 AUTIN5, 0 /NOT USED
42         0015 0000 AUTIN6, 0 /NOT USED
43         0016 0000 AUTIN7, 0 /NOT USED
44         0017 0000 AUTIN8, 0 /USED IN XP INTERRUPT ROUTINE AND BY INIT
45
46         0020 1400 AVAIL, HDAVAIL /HEAD AVAIL LIST
47         0021 7777 JSW, -1 /JOB STATUS WORD
48         /JSW[0]=0
49         /JSW[6:8]=FIELD OF JOB
50         /JSW[9:11]=PRIORITY JOB
51         /-1 FOR BACKGROUND JOB
52         0022 0000 JSWMJ, 0 /PARAMETER OF MAKE JOB
53         0023 0000 DTPNMJ, 0 /IDEM
54
55         /PARAMETERS OF JOB RCUTINES
56         0024 0000 DATAPN, 0 /DATA POINTER
57         0025 0000 PAR1, 0 /LINE NUMBER

```

```

58 0026 0000 PAR2, 0 /CHARACTER OR BUFFER POINTER
59 0027 0000 PAR3, 0 /- NUMBER OF CHARACTERS
60
61 /DATA
62 *30
63 0030 0000 RR, 0 /POINTERS TO EXTRA OUTPUT REGISTER OF MP
64 0031 0000 R, 0 /POINTER TO OUTPUT REGISTER OF MP
65 0032 0000 XPP, 0 /XP BUFFER POINTERS
66 0033 0000 XPNUMB, 0 /XP BUFFER COUNTER
67 0034 0000 PXP, 0 /PX BUFFER POINTER
68 0035 0000 PXNUMB, 0 /PX BUFFER COUNTER
69 0036 0000 XPMR, 0 /XP MESSAGE STATUS
70 0037 0000 B1, 0 /MESSAGE REGISTER
71 0040 0000 STATE, 0 /TERMINAL STATUS
72 0041 0000 PRTNER, 0 /TERMINAL PARTNER
73 0042 0000 LSTCHR, 0 /LAST RECEIVED CHARACTER
74
75 0043 0000 SP1, 0 /SCRATCH
76 0044 7700 LPCNT, -MAXJOB /INIT FOR CRJBSP; SCRATCH AFTER THAT
77 0045 0000 PART, 0 /CURRENT PARTNER
78 0046 0000 A, 0 /SCRATCH
79 0047 0000 CHAR, 0 /SCRATCH
80 0050 7776 X7FLAG, 7776 /WAIT FOR TWO X8 INITIALIZATION FLAGS
81 0051 0000 LOOK, 0 /DID WE CREATE A JOB DURING THIS INTERRUPT?
82 0052 4000 XLINE, 4000 /TERMINAL CONNECTED WITH DISK: NONE
83 0053 0000 CANCEL, 0 /CANCEL REQUEST
84 0054 0000 LISTEN, 0 /IF ZERO WE IGNORE KEYBOARD INTERRUPTS
85 0055 0100 WARN, 100 /NUMBER OF NODES AVAILABLE
86 /NEGATIVE IF AVAIL EXHAUSTED
87 /STOPS ALL PRIOR 0 JOBS
88
89 /MULTIPLEXER VARIABLES
90 0056 0000 MCHAR, 0 /CHARACTER POINTER
91 0057 0000 MILP, 0 /INITIAL LINE TO SENSE
92 0060 0000 SAVLN, 0 /LAST LINE SERVED
93
94 0061 0000 SVAC, 0 /SAVE AC
95 0062 0000 SVLINK, 0 /SAVE LINK
96
97 /DEFINITIONS
98 IFDEF DECIS <
99 EXIT= JMP I [DECIS
100 EXITS= JMP I [EXITKI
101 REACTIVATE=JMP I [UNSTACK
102 MAKE= JMS I [MAKEJB
103 JOB= 0
104 GETBUF= JMS I [GETBF
105 FREEBUF=JMS I [FRBUF
106 SAVE= JMS I [STACK
107 INITIR= JMS I [INTRAP
108 INITJB= JMS I [INJCB
109 GETNODE=JMS I [GETND
110 ECHO= JMS I [ECHOA
111 >
112

```

113	TRNST1=	10	
114	TRNST2=	20	
115	MAXSZ=	7	
116	MSKPR=	4003	
117	MSKSZ=	7	
118	MAXJOB=	100	
119	PXBUCH=	40	
120	JOBCH=	200	
121	MAXPXB=	220	
122	MAXPR=	3	
123	CHMAX=	216	
124	DUMMY=	377	
125	ACURL=	300	
126	CR=	215	
127	LF=	12	
128	DELETE=	243	
129	NOT=	176	
130	BLCK=	216	/BLACK SHIFT
131	RED=	17	/RED SHIFT
132			
133	STACKP=	AUTIN3	
134	SPN=	AUTIN4	
135		PAGE	

```

136 0200 5601 INIT, JMP I ,+1 /CHANGED INTO "JMP I [RDCOMM]"
137
138 0201 2600 AMA, START /AC:=M[AC]
139 0202 3046 DCA A
140 0203 1446 TAD I A
141 0204 5601 JMP I AMA
142
143 0205 0000 AMAL, 0 /AC:=M[AC+LINE NO]
144 0206 1025 TAD PAR1
145 0207 4201 JMS AMA
146 0210 5605 JMP I AMAL
147
148 0211 0000 MALA, 0 /M[AC+LINE NO]:=CHAR
149 0212 1025 TAD PAR1
150 0213 3046 DCA A
151 0214 1047 TAD CHAR
152 0215 3446 DCA I A
153 0216 5611 JMP I MALA
154
155 0217 0000 GTB1, 0 /CHAR:=AC:=M[B1+LINE NO]
156 0220 1037 TAD B1
157 0221 4205 JMS AMAL
158 0222 3047 DCA CHAR
159 0223 5617 JMP I GTB1
160
161 0224 0000 STB1, 0 /M[B1+LINE NO]:=CHAR
162 0225 3047 DCA CHAR
163 0226 1037 TAD B1
164 0227 4211 JMS MALA
165 0230 5624 JMP I STB1
166
167 0231 0000 GTSTC, 0 /CHAR:=M[AC+STATE]
168 0232 1040 TAD STATE
169 0233 4205 JMS AMAL
170 0234 3047 DCA CHAR
171 0235 5631 JMP I GTSTC
172
173 0236 0000 SETST, 0 /M[STATE+LINE NO]:=AC
174 0237 3047 DCA CHAR
175 0240 1040 TAD STATE
176 0241 4211 JMS MALA
177 0242 5636 JMP I SETST
178
179 0243 0000 GTPRT, 0
180 0244 1041 TAD PRTNER /PART:=M[PARTNER+LINE NO]
181 0245 4205 JMS AMAL
182 0246 3045 DCA PART
183 0247 1045 TAD PART
184 0250 1003 TAD MX8 /((PART=XB/RETURN+1)
185 0251 7650 SNA CLA
186 0252 2243 ISZ GTPRT
187 0253 5643 JMP I GTPRT
188
189 0254 0000 STPRT, 0
190 0255 3045 DCA PART /M[PARTNER+LINE NO]:=PART*AC

```

191	0256	1045	TAD	PART	
192	0257	3047	DCA	CHAR	
193	0260	1041	TAD	PRTNER	
194	0261	4211	JMS	MALA	
195	0262	5654	JMP I	STPRT	
196					
197	0263	0000	STLCH,	0	/CHAR:=M[LAST CHAR+LINE NO]:=AC
198	0264	3047	DCA	CHAR	
199	0265	1042	TAD	LSTCHR	
200	0266	4211	JMS	MALA	
201	0267	5663	JMP I	STLCH	
202					
203	0270	0000	GTLCH,	0	/CHAR:=M[LAST CHAR+LINE NO]
204	0271	1042	TAD	LSTCHR	
205	0272	4205	JMS	AMAL	
206	0273	3047	DCA	CHAR	
207	0274	5670	JMP I	GTLCF	
208					
209	0275	0000	STXPMR,	0	/CHAR:=M[XPMR+LINE NO]:=AC
210	0276	3047	DCA	CHAR	
211	0277	1036	TAD	XPMR	
212	0300	4211	JMS	MALA	
213	0301	5675	JMP I	STXPMR	
214					
215	0302	0000	GTXPMR,	0	/CHAR:=M[XPMR+LINE NO]
216	0303	1036	TAD	XPMR	
217	0304	4205	JMS	AMAL	
218	0305	3047	DCA	CHAR	
219	0306	5702	JMP I	GTXPMR	
220					
221	0307	0000	MSSAGE,	0	/CREATE A JOB TO PRINT A MESSAGE
222	0310	3275	DCA	KEY	/NUMBER OF THE MESSAGE
223	0311	1377	TAD	(MSPT	
224	0312	1275	TAD	KEY	
225	0313	4201	JMS	AMA	
226	0314	3026	DCA	PAR2	/POINTER TO THE MESSAGE BUFFER
227	0315	1376	TAD	(MSNR	
228	0316	1275	TAD	KEY	
229	0317	4201	JMS	AMA	
230	0320	3027	DCA	PAR3	/COUNTS THE NUMBER OF CHARACTERS
231	0321	4564	ECHO		/CREATE JOB
232	0322	5707	JMP I	MSSAGE	
233					
234	0323	0000	SWITCH,	0	/EFFECTUATE TURNOVER
235	0324	3302	DCA	TRANS	
236	0325	1025	TAD	PAR1	
237	0326	1003	TAD	MX8	
238	0327	7650	SNA CLA		
239	0330	5340	JMP	PART2	
240	0331	7240	STA		
241	0332	4263	JMS	STLCH	/INITIALIZE INPUT STATUS
242	0333	7240	STA		/OF BOTH PARTNERS
243	0334	4224	JMS	STB1	
244	0335	7305	CLA CLL	IAC RAL	
245	0336	1302	TAD	TRANS	

```
246 0337 4236 JMS SETST /DEFINE LINE STATUS
247 0340 4775 PART2, JMS CHANGE
248 0341 1302 TAD TRANS
249 0342 7110 CLL RAR
250 0343 2302 ISZ TRANS
251 0344 7630 SZL CLA
252 0345 5723 JMP I SWITCH
253 0346 5325 JMP SWITCH+2
254
255 TRANS= GTXPMR
256 KEY= STXPMR
257
258 0347 0000 STPXP, 0 /LOAD PX BUFFER POINTER
259 0350 1026 TAD PAR2 /AND PX BUFFER COUNTER
260 0351 3047 DCA CHAR
261 0352 1034 TAD PXP /FROM JOB PARAMETERS
262 0353 4211 JMS MALA
263 0354 1027 TAD PAR3
264 0355 3047 DCA CHAR
265 0356 1035 TAD PXNUMB
266 0357 4211 JMS MALA
267 0360 5747 JMP I STPXP
268
269 0361 0000 GTPXP, 0 /GET PX BUFFER POINTER
270 0362 1034 TAD PXP /AND COUNTER
271 0363 4205 JMS AMAL
272 0364 3026 DCA PAR2
273 0365 1035 TAD PXNUMB
274 0366 4205 JMS AMAL
275 0367 3027 DCA PAR3
276 0370 5761 JMP I GTPXP
277 0375 0417
278 0376 1001
279 0377 0763
280 PAGE
```

```

281 0400 0000 TKOVR2, 0
282 0401 6002 IOF /CRITICAL
283 0402 3026 DCA PAR2 /CHAR
284 0403 7001 IAC
285 0404 3021 DCA JSW /WE WILL JUMP TO TSNUF
286 0405 6214 RDF
287 0406 3022 DCA JSWMJ /SAVE DF (NOTE: LEVEL=0)
288 0407 1200 TAD TKOVR2 /MAKE A JOB
289 0410 4573 MAKE JOB /OF THE SUBROUTINE
290 0411 1052 TAD XLINE /TERMINAL NO
291 0412 0163 AND [7
292 0413 3025 DCA PAR1
293 0414 3027 DCA PAR3 /NO BUFFER
294 0415 6001 ION /END OF CRITICAL
295 0416 5777 JMP TSNUF1 /PROCESS CHARACTER
296
297 0417 0000 CHANGE, 0 /EXCHANGE PART AND PAR1
298 0420 1025 TAD PAR1
299 0421 3046 DCA A
300 0422 1045 TAD PART
301 0423 3025 DCA PAR1
302 0424 1046 TAD A
303 0425 3045 DCA PART
304 0426 5617 JMP I CHANGE
305
306 0427 0000 BTWEEN, 0 /SKIP IF 0<=CHAR<=AC
307 0430 7040 CMA
308 0431 1047 TAD CHAR
309 0432 7700 SMA CLA
310 0433 5627 JMP I BTWEEN
311 0434 1047 TAD CHAR
312 0435 7700 SMA CLA
313 0436 2227 ISZ BTWEEN
314 0437 5627 JMP I BTWEEN
315
316 0440 0000 EQUAL, 0 /SKIP IF EQUAL
317 0441 7041 CIA
318 0442 1047 TAD CHAR
319 0443 7650 SNA CLA
320 0444 2240 ISZ EQUAL
321 0445 5640 JMP I EQUAL
322
323 0446 0000 NEQUAL, 0 /SKIP IF NOT EQUAL
324 0447 4240 JMS EQUAL
325 0450 2246 ISZ NEQUAL
326 0451 5646 JMP I NEQUAL
327
328 /ROUTINE TO HANDOVER A RECEIVED CHARACTER, CALLED BY MULTIPLEXER
329
330 0452 0000 TKOVER, 0 /CALLED BY THE MULTIPLEXER INTERRUPT
331 0453 7500 SMA /STOPBIT? ROUTIN
332 0454 5274 JMP TK /NO; SKIP CHARACTER
333 0455 7110 CLL RAR /STRIP OFF EXTRANEIOUS BITS
334 0456 7112 CLL RTR
335 0457 0162 AND [377

```

336	0460	3311	DCA	SVCHAR	/STORE CHARACTER
337	0461	4565	GETNODE		
338	0462	3023	DCA	DTPNMJ	/DATA POINTER
339	0463	1023	TAD	DTPNMJ	
340	0464	3013	DCA	SPN	/AUTO INDEXER
341	0465	6414	TTRL		
342	0466	3413	DCA I	SPN	/LINE NUMBER
343	0467	1311	TAD	SVCHAR	
344	0470	3413	DCA I	SPN	/CHARACTER
345	0471	3413	DCA I	SPN	/NO BUFFER
346	0472	13/6	TAD	(TSNUF	/PROCESS CHARACTER
347	0473	4573	MAKE JOB		/CREATE JOB
348	0474	6461	TK, TTRINC		/SELECT NEXT LINE
349	0475	7346	ACM3		
350	0476	1252	TAD	TKOVER	
351	0477	3311	DCA	SVCHAR	/RETURN ADDRESS=3
352	0500	6201	CDF 00		/I
353	0501	1711	TAD I	SVCHAR	/FETCH CHARACTER
354	0502	0375	AND	(1770	/MASK STOP BITS
355	0503	3711	DCA I	SVCHAR	/RESTORE CHARACTER
356	0504	2311	ISZ	SVCHAR	
357	0505	7132	STL RTR		/AC2000
358	0506	3711	DCA I	SVCHAR	/INITIALIZE CAW
359	0507	6231	CDF 30		
360	0510	5652	JMP I	TKOVER	
361					
362	0511	0000	SVCHAR, 0		
363	0575	1770			
364	0576	0600			
365	0577	0601			
366					

PAGE

```

367 0600 4566 TSNUF, INITJB /JOB TO PROCESS ONE TERMINAL CHARACTER
368 0601 4561 TSNUF1, JMS I [GTSTC /GET STATUS
369 0602 1163 TAD [7
370 0603 4560 JMS I [BTWEEN /TRANSITION STATE?
371 0604 5576 EXIT /THEN IGNORE INPUT
372 0605 1026 TAD PAR2
373 0606 1157 TAD [-ACURL
374 0607 7650 SNA CLA /WAS IT TACURL?
375 0610 1156 TAD [10 /YES
376 0611 1215 TAD JUMPER
377 0612 1047 TAD CHAR
378 0613 3214 DCA .+1
379 0614 0000 TJUMP, 0 /DISPATCH
380 0615 5616 JUMPER, JMP I .+1 /SWITCH ACCORDING TO STATUS
381 0616 0644 SOLO /0 INITIAL STATE
382 0617 1000 TTRUM /1 ASSEMBLE MESSAGE
383 0620 0715 TSIN /2 CONVERSATION ACTIVE
384 0621 1307 TSCUT /3 CONVERSATION PASSIVE
385 0622 1147 DUPLX1 /4 FULL DUPLEX ASCII INPUT
386 0623 3274 DUPLX2 /5 FULL DUPLEX BINARY INPUT
387 0624 3300 SINGLE /6 FULL DUPLEX SINGLE CHARACTER
388 0625 3600 DECIS /7 ILLEGAL, IGNORE
389
390 0626 0636 SOLAC /0 EXIT INITIAL STATUS
391 0627 1074 NOMES /1 ILLEGAL, ERROR
392 0630 0646 SINAC /2 EXIT CONVERSATION
393 0631 0703 SOUTAC /3 CHANGE CONVERSATION
394 0632 1154 DUPLAC /4 ATTENTION IN DUPLEX
395 0633 3274 DUPLX2 /5 NOT SPECIAL
396 0634 3300 SINGLE /6 NOT SPECIAL
397 0635 3600 DECIS /7 ILLEGAL, IGNORE
398
399 0636 7001 SOLAC, IAC /STATUS:=1
400 0637 4555 JMS I [SETST /INITIALIZE MESSAGE ADMINISTRATION
401 0640 7240 STA
402 0641 4554 JMS I [STLCH
403 0642 7240 STA
404 0643 4553 JMS I [STB1
405
406 0644 4564 SOLO, ECHO /PRINT CHARACTER
407 0645 5576 EXIT
408
409 0646 4552 SINAC, JMS I [GTPRT /PARTNER=X8?
410 0647 5267 JMP NOX8 /NO
411 0650 4551 JMS I [ACBUF /YES, SEND DISCONNECTION MESSAGE
412 0651 1150 TAD [ACURL
413 0652 4547 JMS I [PXECHE
414 0653 4546 JMS I [STPRT /NO PARTNER
415 0654 4545 JMS I [GTXPMP /SKIP CURRENT XP BUFFER, IF ANY
416 0655 1047 TAD CHAR
417 0656 7740 SMA SZA CLA
418 0657 5262 JMP SNEX
419 0660 4544 JMS I [STXPMP
420 0661 4543 JMS I [REQUA
421 0662 1156 SNEX, TAD [TRNST1 /TRANSITION TO SOLO

```

422	0663	4555		JMS I	[SETST	
423	0664	1142		TAD	[5	/INFORM TERMINAL
424	0665	4541		JMS I	[MESSAGE	
425	0666	55/6		EXIT		
426						
427	0667	1045	NOX8,	TAD	PART	/INFORM PARTNER, IF ANY
428	0670	7041		CIA		/OF DISCONNECTION
429	0671	1025		TAD	PAR1	
430	0672	7650		SNA CLA		
431	0673	5262		JMP	SNEX	
432	0674	4540		JMS I	[CHANGE	
433	0675	1156		TAD	[TRNST1	
434	0676	4555		JMS I	[SETST	
435	0677	7327		AC6		
436	0700	4541		JMS I	[MESSAGE	
437	0701	4540		JMS I	[CHANGE	
438	0702	5262		JMP	SNEX	
439						
440	0703	4552	SOUTAC,	JMS I	[GTPRT	/PARTNER=X8?
441	0704	7410		SKP		
442	0705	5250		JMP	SINAC+2	/YES, DISCONNECT
443	0706	1137		TAD	[TRNST2	/NO, FORCE TURNOVER
444	0707	4536		JMS I	[SWITCH	
445	0710	7001		IAC		
446	0711	4541		JMS I	[MESSAGE	
447	0712	4540		JMS I	[CHANGE	
448	0713	7326		CLA STL	RTL	
449	0714	5265		JMP	SNEX+3	
450						
451	0715	4564	TSIN,	ECHO		/ECHO CHARACTER
452	0716	1026		TAD	PAR2	
453	0717	1155		TAD	[-CR	
454	0720	7650		SNA CLA		/RETURN?
455	0721	5341		JMP	SINCR	
456						
457	0722	1026	TSXIT,	TAD	PAR2	
458	0723	4554		JMS I	[STLCH	/SAVE LAST CHAR
459	0724	1154		TAD	[LF	
460	0725	4533		JMS I	[NEQUAL	/CHAR=LF?
461	0726	4532		JMS I	[TRTURN	/YES, TRY TURNOVER
462	0727	4552	COMSIN,	JMS I	[GTPRT	
463	0730	7410		SKP		
464	0731	5777		JMP	DUPLX1+2	/SEND TO X8
465	0732	1045		TAD	PART	/OR TO PARTNERS TERMINAL
466	0733	7041		CIA		/IF ANY
467	0734	1025		TAD	PAR1	
468	0735	7650		SNA CLA		
469	0736	55/6		EXIT		
470	0737	4540		JMS I	[CHANGE	
471	0740	5244		JMP	SOLO	
472						
473	0741	4531	SINCR,	JMS I	[GTLCH	/B1:=LAST CHARACTER
474	0742	1047		TAD	CHAR	
475	0743	4553		JMS I	[STB1	
476	0744	5322		JMP	TSXIT	

```
477                                     /LEVEL 0
478 0745 0000 NXTPXB, 0                /NEXT PX-BUFFER
479 0746 7305 AC2                      /SIZE=2
480 0747 6201 CDF 00
481 0750 4572 GETBUF
482 0751 7402 HLT
483 0752 6001 ION
484 0753 3026 DCA PAR2 /BUFFER POINTER
485 0754 1130 TAD [-14
486 0755 3027 DCA PAR3 /BUFFER COUNTER
487 0756 5745 JMP I NXTPXB
488
489                                     /LEVEL 0
490 0757 0000 FREEBF, 0                /RETURN PX-BUFFER TO POOL
491 0760 1027 TAD PAR3
492 0761 1127 TAD [14
493 0762 7041 CIA
494 0763 1026 TAD PAR2 /BUFFER POINTER
495 0764 6201 CDF 00
496 0765 4571 FREEBUF
497 0766 0002 2 /SIZE=2
498 0767 6001 ION
499 0770 5757 JMP I FREEBF
500 0777 1151
501 PAGE
```

```

502
503 /COLLECT INITIAL MESSAGE, SPECIFYING THE PARTNER
504 1000 4564 TTRUM, ECHO
505 1001 1026 TAD PAR2
506 1002 1135 TAD [-CR
507 1003 7450 SNA /RETURN?
508 1004 5576 EXIT /IGNORE CR
509 1005 1126 TAD [CR-LF
510 1006 7640 SZA CLA /LF?
511 1007 5301 JMP NOLFT
512 1010 4525 JMS I [GTB1 /END OF MESSAGE
513 1011 7240 STA
514 1012 4533 JMS I [NEQUAL /TWO DIGIT TERMINAL NO?
515 1013 4531 JMS I [GTLCH /NO, ONE DIGIT OR LETTER
516 1014 1377 TAD (103 /C
517 1015 4533 JMS I [NEQUAL
518 1016 5357 JMP KILL
519
520 1017 1376 TAD (130 /X
521 1020 4524 JMS I [EQUAL
522 1021 5240 JMP NOX
523 1022 1050 TAD X7FLAG /X8 PRESENT
524 1023 7710 SPA CLA
525 1024 5273 JMP PBUSY+1 /NO CONNECTION WITH X8
526
527 1025 1002 TURNON, TAD X8 /START SHAKE HAND SEQUENCE
528 1026 4546 JMS I [STPRT /PARTNER:=X8
529 1027 4523 JMS I [NXTPXB /GET PX BUFFER
530 1030 4522 JMS I [STPXP
531 1031 1150 TAD [ACURL
532 1032 4547 JMS I [PXEQHO /INFORM X8
533 1033 1375 TAD (11
534 1034 4541 JMS I [MESSAGE /INFORM TERMINAL
535 1035 1156 TAD [TRNST1
536 1036 4536 JMS I [SWITCH
537 1037 5576 EXIT
538
539 1040 7240 NOX, STA
540 1041 1002 TAD X8
541 1042 4560 JMS I [BTWEEN /VALID MESSAGE?
542 1043 5274 JMP NOMES /NO
543 1044 1047 TAD CHAR /YES, CONNECT IF PARTNER FREE
544 1045 4546 JMS I [STPRT
545 1046 1025 TAD PAR1
546 1047 4524 JMS I [EQUAL /OWN TERMINAL?
547 1050 5254 JMP .+4 /AVOID CHECK ON PARTNER
548 1051 1134 TAD [12
549 1052 4555 JMS I [SETST
550 1053 5267 JMP MEME
551
552 1054 4540 JMS I [CHANGE
553 1055 4561 JMS I [GTSTC
554 1056 4524 JMS I [EQUAL /PARTNER FREE?
555 1057 5272 JMP PBUSY /NO
556 1060 7307 AC4

```

557	1061	4541	JMS I	[MESSAGE	
558	1062	1045	TAD	PART	/YES, CONNECT
559	1063	4546	JMS I	[STPRT	
560	1064	4540	JMS I	[CHANGE	
561	1065	1137	TAD	[TRNST2	
562	1066	4536	JMS I	[SWITCH	
563	1067	7325	MEME,	AC3	
564	1070	4541	JMS I	[MESSAGE	
565	1071	5576	EXIT		
566					
567	1072	4540	PBUSY,	JMS I	[CHANGE /REPORT PARTNER BUSY
568	1073	7001	IAC		
569					
570	1074	1163	NOMES,	TAD	[7 /REPORT ERROR
571	1075	4541	JMS I	[MESSAGE	
572	1076	1156	TAD	[TRNST1	/TRANSITION TO INITIAL STATE
573	1077	4555	JMS I	[SETST	
574	1100	5576	EXIT		
575					
576	1101	4525	NOLFT,	JMS I	[GTB1 /PROCESS MESSAGE CHARACTER
577	1102	7240	STA		
578	1103	4524	JMS I	[EQUAL	
579	1104	5274	JMP	NOMES	/THREE DIGIT TERMINAL NO, ERROR
580	1105	4531	JMS I	[GTLCH	
581	1106	1047	TAD	CHAR	
582	1107	3045	DCA	PART	
583	1110	1026	TAD	PAR2	
584	1111	0121	AND	[177	
585	1112	1374	TAD	(-72	
586	1113	7100	CLL		
587	1114	1154	TAD	[12	
588	1115	7420	SNL		/DIGIT?
589	1116	5335	JMP	NOCF	/NO
590	1117	0373	AND	(17	/USELESS
591	1120	2045	ISZ	PART	/FIRST ONE
592	1121	5324	JMP	SETB1	/NO
593	1122	4554	TRXIT,	JMS I	[STLCH /YES, SAVE
594	1123	5576	EXIT		
595					
596	1124	3047	SETB1,	DCA	CHAR /ASSEMBLE TERMINAL NUMBER
597	1125	1045	TAD	PART	/MULTIPLY BY TEN
598	1126	7106	CLL RTL		
599	1127	1045	TAD	PART	
600	1130	7004	RAL		
601	1131	1047	TAD	CHAR	/ADD NEXT DIGIT
602	1132	1372	TAD	(-12	
603	1133	4771	JMS	STB1	/STORE
604	1134	5322	JMP	TRXIT	
605					
606	1135	1370	NOCF,	TAD	(60-103 /WE CAME HERE WITH CHAR-60
607	1136	7450	SNA		/C?
608	1137	5344	JMP	TRXIT2	/YES
609	1140	1367	TAD	(103-130	
610	1141	7650	SNA CLA		/X?
611	1142	2045	ISZ	PART	/AND NOT AFTER ANOTHER LETTER?

612	1143	5274	JMP	NOMES	/ELSE ERROR
613	1144	1026	TRXIT2, TAD	PAR2	
614	1145	0121	AND	[177	/CHAR AGAIN
615	1146	5322	JMP	TRXIT	
616					
617	1147	1026	DUPLX1, TAD	PAR2	/PUT CHAR IN PX BUFFER
618	1150	4554	JMS I	[STLCH	
619	1151	1047	TAD	CHAR	
620	1152	4547	JMS I	[PXECHO	
621	1153	5576	EXIT		
622					
623	1154	4766	DUPLAC, JMS I	{ACBUF	/SEND "a" TO X8
624	1155	1150	TAD	{ACURL	
625	1156	5350	JMP	DUPLX1+1	
626					
627	1157	1025	KILL, TAD	PAR1	/SOMEONE TYPED aC
628	1160	7640	SZA CLA		/DID HE TYPE IT ON THE COMMAND TERMINAL?
629	1161	5274	JMP	NOMES	/NO, WE ARE NOT GOING TO LISTEN TO HIM
630	1162	2053	ISZ	CANCEL	/YES, RAISE FLAG TO KILL THE RUNNING PROG.
631	1163	7000	NOP		/USELESS
632	1164	5277	JMP	NOMES+3	/BACK TO INITIAL STATE
633	1166	3035			
634	1167	7753			
635	1170	7755			
636	1171	0224			
637	1172	7766			
638	1173	0017			
639	1174	7706			
640	1175	0011			
641	1176	0130			
642	1177	0103			
643					

```

644      /JOB TO PROCESS XP BUFFER
645      1200  4566      TKXPBU, INITJB
646      1201  1026      TAD      PAR2      /BUFPTR-1
647      1202  1027      TAD      PAR3      /BUFCTR
648      1203  4520      JMS I      [AMA
649      1204  3047      DCA      CHAR      /LAST CHAR OF BUF
650      1205  1117      TAD      [400
651      1206  0047      AND      CHAR
652      1207  3272      DCA      UP      /SAVE MESSAGE TYPE
653      1210  1116      TAD      [600
654      1211  4533      JMS I      [NEQUAL /MESSAGE BUFFER?
655      1212  5777      JMP I      (XACURL /YES
656      1213  4561      JMS I      [GTSTC
657      1214  1047      TAD      CHAR
658      1215  1115      TAD      [-7
659      1216  7100      CLL
660      1217  1142      TAD      [5
661      1220  7620      SNL CLA      /VALID STATE?
662      1221  5346      JMP      REQXP /NO, SKIP BUFFER
663      1222  4273      JMS      TSTMNR /YES, SET BUFFER INDICATION
664
665      1223  1027      TAD      PAR3      /CONVERT BUFFER TO ISO CODE
666      1224  7041      CIA
667      1225  3272      DCA      UP
668      1226  1026      TAD      PAR2
669      1227  3010      DCA      AUTIN1 /READ POINTER
670      1230  1026      TAD      PAR2
671      1231  3011      DCA      AUTIN2 /WRITE POINTER
672
673      1232  1410      LP,      TAD I      AUTIN1
674      1233  0162      AND      [377      /8 BITS ONLY
675      1234  1376      TAD      (-CHMAX
676      1235  7540      SMA SZA      /LEGAL CODE?
677      1236  5242      JMP      STOR-1 /NO, SKIP
678      1237  1375      TAD      (ISOTAB+CHMAX /CONVERT
679      1240  4520      JMS I      [AMA
680      1241  7510      SPA
681      1242  7610      SKP CLA      /LEGAL RESULT?
682      1243  3411      STOR,    DCA I      AUTIN2 /ELSE IGNORE
683      1244  2272      ISZ      UP      /STORE
684      1245  5232      JMP      LP      /NEXT OR CONVERSION DONE
685
686      1246  1011      TAD      AUTIN2
687      1247  1027      TAD      PAR3
688      1250  7041      CIA
689      1251  1010      TAD      AUTIN1
690      1252  7450      SNA
691      1253  5346      JMP      REQXP /CONVERTED BUFFER EMPTY?
692      1254  3027      DCA      PAR3      /YES
693      1255  4561      ISOD1,   JMS I      [GTSTC /NO, SAVE -BUFCT
694      1256  7305      AC2
695      1257  4524      JMS I      [EQUAL /X8 HAS TURN ALREADY?
696      1260  5514      JMP I      [SOLO /YES, CONTINUE PRINTING
697      1261  1027      TAD      PAR3      /NO, STORE BUFFER LENGTH
698      1262  3047      DCA      CHAR

```

```

699 1263 1053 TAD XPNUMB
700 1264 4513 JMS I [MALA
701 1265 4531 JMS I [GTLCH
702 1266 1134 TAD [LF
703 1267 4533 JMS I [NEQUAL
704 1270 4532 JMS I [TRTURN /TRY TO GET TURN
705 1271 5576 EXIT
706
707 1272 0000 UP, 0 /SCRATCH
708
709 1273 0000 TSTMNR, 0
710 1274 7346 ACM3
711 1275 1047 TAD CHAR
712 1276 7440 SZA /CHAR=3
713 1277 7001 IAC
714 1300 7650 SNA CLA /OR CHAR=2?
715 1301 1272 TAD UP
716 1302 7650 SNA CLA /AND UP NONZERO?
717 1303 7001 IAC
718 1304 7001 IAC
719 1305 4544 JMS I [STXPMR
720 1306 5673 JMP I TSTMNR
721
722 1307 4552 TSOUT, JMS I [GTPRT /PARTNER=X8?
723 1310 5320 JMP NOTX8 /NO
724 1311 4545 JMS I [GTXPMP /BUFFER PRESENT?
725 1312 2047 ISZ CHAR /NO, END OF LINE?
726 1313 5335 JMP TSXIT1
727 1314 4337 JMS TURN
728 1315 1026 TAD PAR2
729 1316 4547 JMS I [PXCHO /SEND CHARACTER TO X8
730 1317 5576 EXIT
731
732 1320 4540 NOTX8, JMS I [CHANGE /SELECT PARTNER
733 1321 4531 JMS I [GTLCH
734 1322 1134 TAD [LF
735 1323 4524 JMS I [EQUAL /END OF LINE?
736 1324 5576 EXIT /NO
737 1325 4525 JMS I [GTB1
738 1326 1152 TAD [DELETE
739 1327 4533 JMS I [NEQUAL /AND LINE WAS NOT DELETED?
740 1330 5576 EXIT /NO
741 1331 4540 JMS I [CHANGE /YES, THEN TURNOVER
742 1332 4337 JMS TURN
743 1333 4540 JMS I [CHANGE
744 1334 5514 JMP I [SOLO
745 1335 4554 TSXIT1, JMS I [STLCH
746 1336 5576 EXIT
747
748 1337 0000 TURN, 0 /EFFECTUATE TURNOVER
749 1340 7132 STL RTR /ADD COLOUR SHIFT
750 1341 1026 TAD PAR2
751 1342 3026 DCA PAR2
752 1343 4564 ECHO /PRINT CHAR OR BUFFER
753 1344 4536 JMS I [SWITCH

```

/ UTOR DD. 12/3/73

PAL8 9/17/74 PAGE 7-2

754	1345	5737	JMP I	TURN
755				
756	1346	4543	REQUXP, JMS I	[REQUA /ASK NEXT XP BUFFER
757	1347	5576	EXIT	
758	1375	0716		
759	1376	7562		
760	1377	1400		
761			PAGE	

762	1400	2026	XACURL,	ISZ	PAR2	
763	1401	1426		TAD I	PAR2	/SELECT MESSAGE NUMBER
764	1402	7650		SNA CLA		/=0?
765	1403	5231		JMP	RUNOUT	/RUNOUT
766	1404	4561		JMS I	[GTSTC	
767	1405	4533		JMS I	[NEQUAL	/STATE=0?
768	1406	5777		JMP I	(TURNON	/MAKE CONNECTION
769	1407	1047		TAD	CHAR	
770	1410	1115		TAD	[-7	
771	1411	7100		CLL		
772	1412	1142		TAD	[5	
773	1413	7630		SZL CLA		/TRANSITION STATE?
774	1414	4552		JMS I	[GTPRT	/OR NOT CONNECTED WITH TERMINAL?
775	1415	5267		JMP	TBUSY	/REFUSE MESSAGE
776	1416	1047		TAD	CHAR	
777	1417	1112		TAD	[-4	/STATE=SIN/SOUT?
778	1420	7710		SPA CLA		
779	1421	5240		JMP	SISOUT	/YES
780	1422	1376	CONOUT,	TAD	(3	/AC+3
781	1423	4247		JMS	NEWCCN	/STATE:=SIN/SOUT
782	1424	4523		JMS I	[NXTPIXB	/RESERVE PX-BUFFER
783	1425	4522		JMS I	[STXP	/SAVE POINTERS
784	1426	1150		TAD	[ACURL	/AFFIRM MESSAGE
785	1427	4547		JMS I	[PXCHO	
786	1430	5576		EXIT		
787						
788	1431	7130	RUNOUT,	CLL CML	RAR	/COUNT TO 4000
789	1432	1375		TAD	(-60	/60 BLANKS
790	1433	3027		DCA	PAR3	
791	1434	1032		TAD	XPP	
792	1435	4511		JMS I	[AMAL	
793	1436	3026		DCA	PAR2	/POINTS TO (0
794	1437	5774		JMP I	(ISOD1	
795						
796	1440	1426	SISOUT,	TAD I	PAR2	
797	1441	0376		AND	(3	/NEW STATUS
798	1442	7440		SZA		/=0?
799	1443	5222		JMP	CONOUT	/NO
800	1444	3045		DCA	PART	/YES, DISCONNECT
801	1445	4247		JMS	NEWCCN	
802	1446	5773		JMP	REQUXP	/ASK AN XP-BUFFER
803						
804	1447	0000	NEWCON,	0		
805	1450	3043		DCA	SP1	/SAVE NEW STATUS
806	1451	1043		TAD	SP1	
807	1452	3047		DCA	CHAR	
808	1453	4772		JMS	TSTMNR	/SET BUFFER INDICATION
809	1454	1043		TAD	SP1	
810	1455	1156		TAD	[TRNST1	/ADD TRANSITION BIT
811	1456	4555		JMS I	[SETST	/NEW STATE
812	1457	7344		CLL STA	RAL	
813	1460	1047		TAD	CHAR	/SELECT MESSAGE
814	1461	4541		JMS I	[MSSAGE	
815	1462	1045		TAD	PART	/DEFINE PARTNER
816	1463	4546		JMS I	[STPRT	

817	1464	4510		JMS I	[GTPXP /SKIP PX-BUFFER
818	1465	4507		JMS I	[FREEBF
819	1466	5647		JMP I	NEWCCN
820					
821	1467	4523	TBUSY,	JMS I	[NXTPIXB /NEW PX BUFFER
822	1470	4522		JMS I	[STPXP /STORE POINTERS
823	1471	1371		TAD	(NOT
824	1472	4547		JMS I	[PXEXCH /REFUSE CONNECTION WITH X8
825	1473	1150		TAD	[ACURL
826	1474	4547		JMS I	[PXEXCH /ATTENTION SYMBOL TO X8
827	1475	4507		JMS I	[FREEBF /RETURN BUFFER TO POOL
828	1476	5773		JMP	REQUXP
829					
830					
831	1477	0000	OFFTRS, 0		
832	1500	4561		JMS I	[GTSTC /GET STATE
833	1501	1106		TAD	[-10
834	1502	0047		AND	CHAR
835	1503	7450		SNA	/TRANSITION STATE?
836	1504	5677		JMP I	OFFTRS /NO, RETURN
837	1505	7041		CIA	
838	1506	1047		TAD	CHAR /REMOVE TRANSITION BIT
839	1507	3340		DCA	SVCHR /SAVE FUTURE STATE
840	1510	1047		TAD	CHAR
841	1511	0156		AND	[10
842	1512	7650		SNA CLA	
843	1513	5317		JMP	HITRS /HIGH TRANSITION
844					
845	1514	1340		TAD	SVCHR /FINISH TRANSITION STATE
846	1515	4555		JMS I	[SETST
847	1516	5677		JMP I	OFFTRS
848					
849	1517	4552	HITRS,	JMS I	[GTPRT /PARTNER=X8?
850	1520	7410		SKP	
851	1521	7402		HLT	/YES, IMPOSSIBLE
852	1522	4540		JMS I	[CHANGE
853	1523	4561		JMS I	[GTSTC /STATE OF PARTNER
854	1524	1156		TAD	[10
855	1525	0047		AND	CHAR
856	1526	7640		SZA CLA	/IS IT LOW?
857	1527	5333		JMP	LTRS /YES
858	1530	4540		JMS I	[CHANGE /NO, LOWER OWN TRANSITION
859	1531	1156		TAD	[10
860	1532	5314		JMP	HITRS-3
861	1533	1047	LTRS,	TAD	CHAR /REMOVE ALL TRANSITIONS
862	1534	0163		AND	[7
863	1535	4555		JMS I	[SETST
864	1536	4540		JMS I	[CHANGE
865	1537	5314		JMP	HITRS-3
866	1540	0000	SVCHR, 0		
867					
868	1541	0000	CHBU1, 0		/CHANGE BUFFER
869	1542	3353		DCA	SVPXR /NEW BUFFER
870	1543	6201		CDF 00	
871	1544	1770		TAD I	(PXREPT /CURRENT BUFFER

872	1545	0367	AND	(7700	
873	1546	4571	FREEBUF		/RETURN TO POOL
874	1547	0004	4		
875	1550	6001	ION		
876	1551	1353	TAD	SVPXR	/NEXT BUFFER
877	1552	5741	JMP I	CHBU1	
878					
879	1553	0000	SVPXR,	0	
880	1567	7700			
881	1570	1760			
882	1571	0176			
883	1572	1273			
884	1573	1346			
885	1574	1255			
886	1575	7720			
887	1576	0003			
888	1577	1025			
889					

PAGE

```

890 1600 6332 PX2, CBUP2 /CLEAR PX-FLAG
891 1601 7305 CLA CLL IAC RAL /PRIORITY INTERRUPT ROUTINE
892 1602 4567 INITIR /INITIALIZE INTERRUPT ROUTINE
893 1603 6231 CDF 30
894 1604 7240 CLA CMA
895 1605 3366 DCA PXREADY /PXREADY:=TRUE
896 1606 1361 TAD ATTPX
897 1607 7650 SNA CLA /ANY BUFFERS?
898 1610 5576 EXIT
899 1611 2362 ISZ PXIBC /SPACE BUFFER POINTERS EMPTY?
900 1612 5221 JMP PXS
901 1613 1760 TAD I PXREPT
902 1614 4777 JMS I (CHBU1 /NEXT BUFFER
903 1615 3360 DCA PXREPT /READ POINTER
904 1616 7201 CLA IAC
905 1617 1105 TAD [-PXBCN
906 1620 3362 DCA PXIBC /COUNTER
907 1621 1760 PXS, TAD I PXREPT
908 1622 2360 ISZ PXREPT
909 1623 3367 DCA TERNR /TERMINAL NUMBER
910 1624 1760 TAD I PXREPT
911 1625 2390 ISZ PXREPT
912 1626 3370 DCA BUFPTR /BUFFER
913 1627 7240 PXSEND, CLA CMA
914 1630 1361 TAD ATTPX
915 1631 3361 DCA ATTPX /DECREASE COUNT
916 1632 1367 TAD TERNR
917 1633 6342 APX1 /TERMINAL NUMBER TO X8
918 1634 7710 SPA CLA
919 1635 7040 CMA /NEGATIVE TERMINAL NUMBER MEANS:
920 1636 6344 APX2 /ASK NEXT BUFFER
921 1637 6334 APX3
922 1640 7325 AC3
923 1641 6352 CKP
924 1642 1115 TAD [-7 /-4 IN AC
925 1643 3365 DCA SAVDIR
926 1644 4776 PXLOOP, JMS I (PXWAIT /SEND BUFFER TO X8
927 1645 4342 JMS PXCHAR
928 1646 6334 APX3
929 1647 4342 JMS PXCHAR
930 1650 6344 APX2
931 1651 4342 JMS PXCHAR
932 1652 6342 APX1
933 1653 2365 ISZ SAVDIR
934 1654 5244 JMP PXLOOP
935 1655 7201 PxDONE, CLA IAC
936 1656 6356 OKP CAP /INFORM X8
937 1657 1367 TAD TERNR
938 1660 7750 SPA SNA CLA
939 1661 5267 JMP .+6
940
941 1662 1130 TAD [-14 /RETURN BUFFER TO POOL
942 1663 1370 TAD BUFPTR
943 1664 6201 CDF 00
944 1665 4571 FREEBUF

```

```

945 1666 0002 2
946
947 1667 6001 ION
948 1670 3366 DCA PXREADY /PXREADY:=FALSE
949 1671 5576 EXIT
950
951 /JOB ROUTINE
952 /INSERT PX-BUFFER IN QUEUE FOR X8
953 1672 6002 PXOVER, IOF
954 1673 6231 CDF 30
955 1674 1024 TAD DATAPN
956 1675 3013 DCA SPN
957 1676 1413 TAD I SPN
958 1677 3367 DCA TERNR /TERMINAL NUMBER
959 1700 1413 TAD I SPN
960 1701 3370 DCA BUFPTR /BUFFER
961 1702 1024 TAD DATAPN
962 1703 4504 JMS I [FREEND
963 1704 6001 ION
964 1705 2361 ISZ ATTPX
965 1706 1361 TAD ATTPX
966 1707 1366 TAD PXREADY
967 1710 7650 SNA CLA /IF ATTPX=1 AND PXREADY THEN PXSEND
968 1711 5227 JMP PXSEND
969 1712 1361 TAD ATTPX
970 1713 13/5 TAD (-MAXPXB
971 1714 7700 SMA CLA
972 1715 7402 HLT /TOO MANY PX-BUFFERS
973 1716 2363 ISZ PXOBC
974 1717 5333 JMP PX1
975 1720 7307 AC4 /GET SPACE FOR POINTERS
976 1721 6201 CDF 00
977 1722 4572 GETBUF
978 1723 7402 HLT
979 1724 6001 ION
980 1725 3764 DCA I PXWAPT /POINTER TO NEXT BUFFER
981 1726 1764 TAD I PXWAPT
982 1727 3364 DCA PXWAPT
983 1730 7201 CLA IAC
984 1731 1105 TAD [-PXBUCL
985 1732 3363 DCA PXOBC /COUNTS THE NUMBER OF POINTERS
986 1733 1367 PX1, TAD TERNR /TERMINAL NO
987 1734 3764 DCA I PXWAPT
988 1735 2364 ISZ PXWAPT
989 1736 1370 TAD BUFPTR /BUFFER POINTER
990 1737 3764 DCA I PXWAPT
991 1740 2364 ISZ PXWAPT
992 1741 5576 EXIT
993
994 1742 0000 PXCHAR, 0 /TAKE NEXT CHARACTER FROM BUFFER
995 1743 7200 CLA
996 1744 2370 ISZ BUFPTR
997 1745 1770 TAD I BUFPTR
998 1746 5742 JMP I PXCHAR
999

```

1000	1747	0000	SETIF, 0	/INITIALIZE POINTER BUFFER
1001	1750	3360	DCA	PXREPT /READ POINTER
1002	1751	1360	TAD	PXREPT
1003	1752	3364	DCA	PXWAPT /ALSO WRITE POINTER
1004	1753	1105	TAD	[-PXBUEN/EMPTY BUFFER
1005	1754	3362	DCA	PXIBC
1006	1755	1105	TAD	[-PXBUEN
1007	1756	3363	DCA	PXOBC
1008	1757	5747	JMP I	SETIF
1009				
1010	1760	0000	PXREPT, 0	/PX READ POINTER
1011	1761	0000	ATTPX, 0	/ATTENTION PX
1012	1762	0000	PXIBC, 0	/PX INPUT BUFFER COUNTER
1013	1763	0000	PXOBC, 0	/PX BUFFER OUTPUT COUNTER
1014	1764	0000	PXWAPT, 0	/PX WRITE POINTER
1015	1765	0000	SAVDIR, 0	/
1016	1766	7777	PXREADY, 7777	/PX READY
1017	1767	0000	TERNR, 0	/TERMINAL NUMBER
1018	1770	0000	BUFPTR, 0	/BUFFER POINTER
1019	1775	7560		
1020	1776	2111		
1021	1777	1541		
1022				

PAGE

```

1023 2000 6302 XP2, CBUX2 /CLEAR XP-FLAG
1024 2001 7305 AC2 /PRIORITY IR
1025 2002 4567 INITIR /INITIALIZE IR
1026 2003 6231 CDF 30
1027 2004 7305 AC2
1028 2005 6322 OKX
1029 2006 4300 JMS XPWAIT /TURNS INTERRUPT ON
1030 2007 6312 XPA1 /READ TERMINAL NUMBER
1031 2010 3255 DCA XPTNR
1032 2011 1255 TAD XPTNR
1033 2012 1032 TAD XPP
1034 2013 3260 DCA SCHARX
1035 2014 1660 TAD I SCHARX
1036 2015 3256 DCA XPBPTR /XP-BUFFER[TERMINAL NUMBER]
1037 2016 1660 TAD I SCHARX
1038 2017 3017 DCA AUTIN8
1039 2020 3261 DCA CNTCHR
1040 2021 1112 TAD [-4]
1041 2022 3257 DCA XPLPCN
1042 2023 4300 XPLOOP, JMS XPWAIT /RECEIVE BUFFER FROM X8
1043 2024 6304 XPA3
1044 2025 4262 JMS XPSTOR
1045 2026 6314 XPA2
1046 2027 4262 JMS XPSTCR
1047 2030 6312 XPA1
1048 2031 4262 JMS XPSTCR
1049 2032 2257 ISZ XPLPCN
1050 2033 5223 JMP XPLOOP
1051 2034 4565 XPDONE, GETNODE
1052 2035 3023 DCA DTPNMJ
1053 2036 1023 TAD DTPNMJ
1054 2037 3013 DCA SPN
1055 2040 1255 TAD XPTNR /TERMINAL NUMBER
1056 2041 3413 DCA I SPN
1057 2042 1256 TAD XPBPTR /BUFFER
1058 2043 3413 DCA I SPN
1059 2044 1261 TAD CNTCHR /COUNT
1060 2045 3413 DCA I SPN
1061 2046 7201 CLA IAC
1062 2047 3022 DCA JSWMJ /SET JSW
1063 2050 1103 TAD [TKXPBU /TRANSLATE XP-BUFFER
1064 2051 4573 MAKE JOB
1065 2052 7201 CLA IAC
1066 2053 6326 OKX CAX
1067 2054 5576 EXIT
1068
1069 2055 0000 XPTNR, 0 /XP TERMINAL NUMBER
1070 2056 0000 XPBPTR, 0 /XP BUFFER COUNTER
1071 2057 0000 XPLPCN, 0 /XP LOOP COUNTER
1072 2060 0000 SCHARX, 0 /SAVE CHAR
1073 2061 0000 CNTCHR, 0 /COUNT CHAR
1074
1075 2062 0000 XPSTOR, 0 /PUT CHARACTER IN XP BUFFER
1076 2063 3260 DCA SCHARX
1077 2064 1260 TAD SCHARX

```

```

1078 2065 3417      DCA I   AUTINB
1079 2066 2261      ISZ     CNTCHR
1080 2067 1260      TAD     SCHARX
1081 2070 0117      AND     [400    /END MARKER
1082 2071 7650      SNA CLA
1083 2072 5662      JMP I    XPSTOR
1084 2073 2257      XPSKIP, ISZ     XPLPCN /YES, SKIP REST OF BUFFER
1085 2074 7410      SKP
1086 2075 5234      JMP     XPDONE
1087 2076 4300      JMS     XPWAIT
1088 2077 5273      JMP     XPSKIP
1089
1090 2100 0000      XPWAIT, 0      /BUSY WAITING LOOP
1091 2101 6002      IOF          /FOR ONE CHARACTER FROM XP
1092 2102 6324      CAX
1093 2103 6301      BUX2
1094 2104 5303      JMP     , -1
1095 2105 6302      CBUX2
1096 2106 7200      CLA
1097 2107 6001      ION
1098 2110 5700      JMP I    XPWAIT
1099
1100
1101 2111 0000      PXWAIT, 0      /BUSY WAITING LOOP
1102 2112 6002      IOF          /FOR ONE CHARACTER TO PX
1103 2113 6354      CAP
1104 2114 6351      BUP2
1105 2115 5314      JMP     , -1
1106 2116 6332      CBUP2
1107 2117 6001      ION
1108 2120 5711      JMP I    PXWAIT
1109
1110
1111 2121 0000      INITIR= JMS I   [.
1112 2122 7510      INTRAP, 0      /INITIALIZE IR
1113 2123 7402      SPA
1114 2124 3022      HLT
1115 2125 1022      DCA     JSWMJ
1116 2126 0377      TAD     JSWMJ
1117 2127 3004      AND     (MSKPR
1118 2130 1021      DCA     SV      /PRIORITY INTERRUPT ROUTINE
1119 2131 0377      TAD     JSW
1120 2132 7040      AND     (MSKPR /PRIORITY CURRENT JOB
1121 2133 1004      CMA
1122 2134 7700      TAD     SV
1123 2135 5341      SMA CLA /PRIORITY IR>PRIORITY CURRENT JOB?
1124 2136 1321      JMP     EXCIR /YES
1125 2137 4573      TAD     INTRAP
1126 2140 5502      MAKE JOB
1127 2141 4570      JMP I    [EXKI
1128 2142 1022      EXCIR, SAVE /SAVE STATUS
1129 2143 3021      TAD     JSWMJ
1130 2144 6001      DCA     JSW      /JOB STATUS WORD INTERRUPT ROUTINE
1131 2145 5721      ION
1132      JMP I    INTRAP /EXECUTE INTERRUPT ROUTINE

```

1133			INITJB=	JMS	I	[.	
1134	2146	0000	INJOB,	0		/INITIALIZE JOB	
1135	2147	6002		IOF			
1136	2150	6231		CDF	30		
1137	2151	1024		TAD		DATAPN	/DATA AREA
1138	2152	3013		DCA		SPN	/AUTO INDEXER
1139	2153	1413		TAD	I	SPN	
1140	2154	3025		DCA		PAR1	/PAR1
1141	2155	1413		TAD	I	SPN	
1142	2156	3026		DCA		PAR2	/PAR2
1143	2157	1413		TAD	I	SPN	
1144	2160	3027		DCA		PAR3	/PAR3
1145	2161	1024		TAD		DATAPN	
1146	2162	2024		ISZ		DATAPN	/DETECT ERRORS
1147	2163	4504		JMS	I	[FREEND	/RETURN NODE TO AVAIL LIST
1148	2164	6001		ION			
1149	2165	5746		JMP	I	INJOB	
1150	2177	4003					
1151				PAGE			

```

1152          /MULTIPLEXER SECTION
1153 2200 6424 T1S, T1CN          /CLEAR AND START CLOCK
1154 2201 7201 CLA IAC
1155 2202 3022 DCA JSWMJ
1156 2203 1057 TAD MILP          /1/5TH OF THE LINES
1157 2204 6473 TTCR TTLR        /LOAD LINE COUNT REGISTER
1158 2205 7200 CLA
1159 2206 6231 CDF 30
1160 2207 5777 JMP I (HDTT) /EXECUTE ALL TTI'S
1161
1162 2210 1060 OUTPUT, TAD SAVLN   /LAST OUTPUT LINE+1
1163 2211 6413 TTCL TTL        /LOAD LINE REGISTER
1164 2212 7200 CLA
1165 2213 1057 TAD MILP
1166 2214 3351 DCA MNCNT        /SET UP TTD-COUNT
1167
1168 2215 1456 TTOGO, TAD I MCHAR /CURRENT CHARACTER
1169 2216 7440 SZL              /OUTPUT COMPLETED?
1170 2217 5223 JMP .+4          /NO
1171 2220 4250 JMS GETMCH       /FETCH NEW CHARACTER
1172 2221 1456 TAD I MCHAR     /NEW CHARACTER
1173 2222 7440 SZL
1174 2223 6404 TTC             /OUTPUT NEXT BIT
1175 2224 3456 DCA I MCHAR     /RESTORE REMAINDER
1176
1177 2225 6401 TTDONE, TTINCR    /SELECT NEXT LINE
1178 2226 2056 ISZ MCHAR        /SELECT NEXT CURRENT CHARACTER
1179 2227 2351 ISZ MNCNT        /1/5TH DONE?
1180 2230 5215 JMP TTOGO        /NO, CONTINUE
1181 2231 6414 TTRL            /READ AND
1182 2232 3060 DCA SAVLN       /SAVE LAST LINE NUMBER
1183 2233 2352 ISZ MJRCNT      /ALL LINES DONE?
1184 2234 5236 JMP .+2
1185 2235 4342 JMS INIMP       /YES, RESET POINTERS
1186
1187          EXITS: JMP I [.
1188 2236 7200 EXITKI, CLA
1189 2237 1051 TAD LOOK
1190 2240 7650 SNA CLA          /ANY JOB CREATED?
1191 2241 5502 JMP I [EXKI      /NO, TAKE THE "SMALL" EXIT
1192 2242 4570 SAVE             /THE INTERRUPTED PROGRAM ON THE STACK
1193 2243 5776 JMP I (DEC1     /AND PERFORM A DECISION
1194
1195 2244 0000 P2, 0
1196 2245 0000 P3, 0
1197 2246 0000 EPTR, 0
1198 2247 0000 SVPT, 0
1199
1200 2250 0000 GETMCH, 0
1201 2251 1060 TAD SAVLN       /LINE NUMBER
1202 2252 7104 CLL RAL         /TIMES TWO
1203 2253 1030 TAD RR
1204 2254 3247 DCA SVPT        /POINTS TO THE PARAMETERS
1205 2255 1647 TAD I SVPT
1206 2256 3244 DCA P2         /SAVE POINTER

```

1207	2257	1247		TAD	SVPT	
1208	2260	7001		IAC		
1209	2261	3245		DCA	P3	/COUNTER
1210	2262	1645		TAD	I P3	
1211	2263	7640		SZA	CLA	/BUFFER?
1212	2264	5302		JMP	BF	/YES
1213						
1214	2265	1244		TAD	P2	/SINGLE CHARACTER
1215	2266	7450		SNA		/ANYTHING TO DO?
1216	2267	5276		JMP	EXGM	/NO
1217	2270	0101		AND	[7400	
1218	2271	7440		SZA		/ATTENTION?
1219	2272	5277		JMP	AT1	/YES
1220	2273	1244		TAD	P2	/CHARACTER
1221	2274	4500		JMS	I [FILLR	/ADD STOPBIT AND PUT THE CHARACTER INTO
1222						/MULTIPLEXER BUFFER THE
1223	2275	3647	EXGMM2,	DCA	I SVPT	/CLEAR PARAMETER
1224	2276	5650	EXGM,	JMP	I GETMCH	
1225						
1226	2277	4775	AT1,	JMS	FIRST	/PUT THE ATTENTION CHARACTER INTO THE
1227	2300	0244		AND	P2	/REMOVE ATTENTION BIT BUFFER
1228	2301	5275		JMP	EXGMM2	/STORE CHARACTER BACK
1229						
1230	2302	2244	BF,	ISZ	P2	/INCREASE POINTER
1231	2303	1644		TAD	I P2	/CURRENT CHARACTER
1232	2304	0101		AND	[7400	
1233	2305	7440		SZA		/ATTENTION?
1234	2306	5334		JMP	DUB	/YES
1235	2307	2645		ISZ	I P3	/INCREASE CHARACTER COUNTER
1236	2310	5322		JMP	NOTLST	
1237	2311	1644	BFLST,	TAD	I P2	/LAST CHARACTER
1238	2312	4774		JMS	FILLRB	
1239	2313	7201		CLA	IAC	/PRIORITY ONE
1240	2314	3022		DCA	JSWMJ	
1241	2315	1060		TAD	SAVLN	/LINE NO
1242	2316	3023		DCA	DTPNMJ	
1243	2317	1077		TAD	[CLBUF	/CLOSE BUFFER
1244	2320	4573		MAKE	JOB	
1245	2321	5275		JMP	EXGMM2	
1246						
1247	2322	1645	NOTLST,	TAD	I P3	
1248	2323	7104		CLL	RAL	
1249	2324	7650		SNA	CLA	/0 OR 4000?
1250	2325	5311		JMP	BFLST	
1251	2326	1644		TAD	I P2	
1252	2327	4774		JMS	FILLRB	
1253	2330	1645		TAD	I P3	
1254	2331	7710		SPA	CLA	
1255	2332	5340		JMP	DUB+4	/STORE POINTER
1256	2333	5337		JMP	DUB+3	/RESET AND STORE POINTER
1257						
1258	2334	4775	DUB,	JMS	FIRST	
1259	2335	0644		AND	I P2	
1260	2336	3644		DCA	I P2	
1261	2337	7240		STA		

1262	2340	1244	TAD	P2	
1263	2341	5275	JMP	EXGMM2	
1264					
1265	2342	0000	INIMP,	0	/INITIALIZE FOR NEXT LINES SCAN
1266	2343	1076	TAD	[-5	
1267	2344	3352	DCA	MJRCNT	
1268	2345	3060	DCA	SAVLN	
1269	2346	1031	TAD	R	
1270	2347	3056	DCA	MCHAR	
1271	2350	5742	JMP I	INIMP	
1272	2351	0000	MNRCNT,	0	
1273	2352	0000	MJRCNT,	0	
1274					
1275			ECHO=	JMS I	[-
1276	2353	0000	ECHOA,	0	
1277	2354	6002		IOF	
1278	2355	1025	TAD	PAR1	/LINE NO
1279	2356	7104	CLL	RAL	
1280	2357	1030	TAD	RR	
1281	2360	3246	DCA	EPTR	/POINTER TO BUFFER
1282	2361	1026	TAD	PAR2	/CHARACTER OR BUFFER POINTER
1283	2362	3646	DCA I	EPTR	/STORE
1284	2363	2246	ISZ	EPTR	/NEXT
1285	2364	1027	TAD	PAR3	/COUNTER
1286	2365	3646	DCA I	EPTR	/STORE
1287			/ISZ	EPTR	
1288	2366	6001	ION		
1289	2367	5753	JMP I	ECHOA	
1290	2374	2563			
1291	2375	3046			
1292	2376	3601			
1293	2377	2610			
1294					

PAGE

1295	2400	1024	CLBUF,	TAD	DATAPN	
1296	2401	3025		DCA	PAR1	
1297	2402	6231		CDF	30	
1298	2403	4777		JMS	I	(OFFTRS
1299	2404	4552		JMS	I	[GTPRT
1300	2405	55/6		EXIT		
1301	2406	4543		JMS	I	[REQUA
1302	2407	4545		JMS	I	[GTXPMR
1303	2410	7344		CLA	CLL	CMA RAL
1304	2411	1047		TAD		CHAR
1305	2412	4544		JMS	I	[STXPMR
1306	2413	2047		ISZ		CHAR
1307	2414	55/6		EXIT		
1308	2415	4531		JMS	I	[GTLCH /TRY TURN
1309	2416	4533		JMS	I	[NEQUAL
1310	2417	55/6		EXIT		
1311	2420	4536		JMS	I	[SWITCH
1312	2421	1376		TAD		(BLCK
1313	2422	3026		DCA		PAR2
1314	2423	3027		DCA		PAR3
1315	2424	4564		ECHO		
1316	2425	55/6		EXIT		
1317						
1318	2426	0000	PXECHO,	0		/PUT CHAR IN PX BUFFER
1319	2427	0121		AND	[177	/7 BITS ONLY
1320	2430	1150		TAD	[INTREP	/CONVERT TO X8 CODE
1321	2431	4520		JMS	I	[AMA
1322	2432	7510		SPA		/VALID CHARACTER
1323	2433	5246		JMP	PXECHX	/NO, SKIP
1324	2434	3043		DCA	SP1	/SAVE CHARACTER
1325	2435	4510		JMS	I	[GTPXP /GET POINTERS
1326	2436	1043		TAD	SP1	
1327	2437	0375		AND	(777	
1328	2440	2026		ISZ	PAR2	
1329	2441	3426		DCA	I	PAR2 /STORE IN BUFFER
1330	2442	2027		ISZ	PAR3	/BUFFER FULL
1331	2443	5250		JMP	TSTBIT	/NO
1332	2444	4475	CHOXIT,	JMS	I	[SEND /YES, SEND AWAY
1333	2445	4522		JMS	I	[STPXP /RESTORE POINTERS
1334	2446	7200	PXECHX,	CLA		
1335	2447	5626		JMP	I	PXECHO
1336						
1337	2450	1043	TSTBIT,	TAD	SP1	/SPECIAL CHARACTERS
1338	2451	0117		AND	[400	
1339	2452	7650		SNA	CLA	
1340	2453	5245		JMP	CHOXIT+1	/NO
1341	2454	1026		TAD	PAR2	/YES, FILL
1342	2455	3317		DCA	SP2	/REST OF LINE
1343	2456	7346		CLA	CLL	CMA RTL
1344	2457	3044		DCA		LPCNT
1345	2460	2026	NILFIL,	ISZ	PAR2	
1346	2461	1162		TAD	[DUMMY	/WITH DUMMIES
1347	2462	3426		DCA	I	PAR2
1348	2463	2027		ISZ		PAR3
1349	2464	7410		SKP		

```

1350 2465 52/1      JMP      BCOUNT
1351 2466 2044      ISZ      LPCNT
1352 2467 5260      JMP      NILFIL
1353 2470 5256      JMP      NILFIL-2
1354 2471 1043      BCOUNT, TAD      SP1      /ACCOUNT PROPER FORMAT
1355 2472 3426      DCA I      PAR2      /FOR NEXT BUFFER
1356 2473 2044      ISZ      LPCNT
1357 2474 7410      SKP
1358 2475 5244      JMP      CHOXIT
1359 2476 1043      TAD      SP1
1360 2477 1374      TAD      (-600
1361 2500 7650      SNA CLA
1362 2501 5244      JMP      CHOXIT
1363 2502 4475      JMS I      [SEND
1364 2503 1317      TAD      SP2
1365 2504 1044      TAD      LPCNT
1366 2505 3317      DCA      SP2
1367 2506 2317      ISZ      SP2
1368 2507 1717      TAD I      SP2
1369 2510 0162      AND      [377
1370 2511 2026      ISZ      PAR2
1371 2512 3426      DCA I      PAR2
1372 2513 2027      ISZ      PAR3
1373 2514 2044      ISZ      LPCNT
1374 2515 5306      JMP      ,-7
1375 2516 5245      JMP      CHOXIT+1
1376
1377 2517 0000      SP2,      0
1378
1379 2520 0000      SETTI,    0      /GENERATE TTI INSTRUCTIONS
1380 2521 1106      TAD      [-10      /DURING MULTIPLEXER INITIALIZATION
1381 2522 3047      DCA      CHAR
1382 2523 1003      TAD      MX8
1383 2524 3044      DCA      LPCNT
1384 2525 13/3      TAD      (HDTTI-1
1385 2526 3017      DCA      AUTIN8
1386 2527 6201      CDF 00
1387 2530 13/2      NXTTI,    TAD      (TTI      /CREATE TTI INSTRUCTIONS
1388 2531 3417      DCA I      AUTIN8      /TTI
1389 2532 1156      TAD      [10      /LSW
1390 2533 1047      TAD      CHAR
1391 2534 3047      DCA      CHAR
1392 2535 1047      TAD      CHAR
1393 2536 3417      DCA I      AUTIN8
1394 2537 7332      AC2000      /CAW
1395 2540 3417      DCA I      AUTIN8
1396 2541 1371      TAD      (JMS I [TKOVER
1397 2542 3417      DCA I      AUTIN8      /ROUTINE CALL
1398 2543 2044      ISZ      LPCNT
1399 2544 5330      JMP      NXTTI
1400 2545 7352      CLA CLL CMA RTR /JMP I (END OF PAGE)
1401 2546 3417      DCA I      AUTIN8
1402 2547 5720      JMP I      SETTI
1403
1404 2550 0000      FILLR,    0      /STORE CHARACTER TO BE PRINTED

```

1405	2551	1117	TAD	[400	/ADD STOPBIT
1406	2552	7104	CLL RAL		/AND STARTBIT
1407	2553	3456	DCA I	MCHAR	/STORE
1408	2554	1031	TAD	R	
1409	2555	1052	TAD	XLINE	/DIT SOMEONE SET XLINE
1410	2556	7041	CIA		/TO THE NUMBER OF THE CURRENT TERMINAL?
1411	2557	1056	TAD	MCHAR	
1412	2560	7650	SNA CLA		
1413	2561	4770	JMS	TODISK	/THEN IT GOES TO THE DISK
1414	2562	5750	JMP I	FILLR	
1415					
1416	2563	0000	FILLRB, 0		
1417	2564	1073	TAD	[1000	/EXTRA STOPBIT
1418	2565	4350	JMS	FILLR	
1419	2566	5763	JMP I	FILLRB	
1420	2570	4231			
1421	2571	4474			
1422	2572	6402			
1423	2573	2607			
1424	2574	7200			
1425	2575	0777			
1426	2576	0216			
1427	2577	1477			
1428					

PAGE

```

1429 2600 2777      (OUTPUT      /CREATE LINK
1430                *.-1
1431 2600 7000  START,  NOP          /HLT
1432 2601 7325        AC3          /LAS
1433 2602 0163        AND          [7
1434 2603 7450        SNA
1435 2604 5200        JMP          START
1436 2605 3002        DCA          X8      /NO OF TERMINALS
1437 2606 1002        TAD          X8
1438 2607 7041        CIA
1439 2610 3003        DCA          MX8
1440
1441                /CREATE JOB SPACE
1442 2611 6231        CDF 30
1443                /TAD          (-MAXJOB
1444                /DCA          LPCNT
1445 2612 1020        TAD          AVAIL
1446 2613 7410        SKP
1447 2614 1404  LPCRJS, TAD I      SV      /CREATE A LIST OF NODES
1448 2615 3004        DCA          SV
1449 2616 7307        AC4
1450 2617 1004        TAD          SV
1451 2620 3404        DCA I      SV      /LINK TO NEXT NODE
1452 2621 2044        ISZ          LPCNT  /FINISHED?
1453 2622 5214        JMP          LPCRJS
1454 2623 3404        DCA I      SV      /ZERO ENDS THE LIST
1455
1456                /COMPUTE SIZE OF BUFFER
1457 2624 3057        DCA          MILP
1458 2625 1003        TAD          MX8
1459 2626 1142        TAD          [5
1460 2627 2057        ISZ          MILP
1461 2630 7510        SPA
1462 2631 5226        JMP          , -3
1463 2632 1002        TAD          X8
1464 2633 3365        DCA          DR      /DR=5*[X8+4 / 5]
1465                /MILP=[X8+4 /5]
1466 2634 1365        TAD          DR
1467 2635 7104        CLL RAL
1468 2636 1365        TAD          DR
1469 2637 3047        DCA          SP      /3*DR : R, RR
1470 2640 1002        TAD          X8
1471 2641 7106        CLL RTL
1472 2642 7004        RAL
1473 2643 1002        TAD          X8      /X8*9
1474 2644 1047        TAD          SP
1475 2645 3047        DCA          SP      /TOTAL SPACE NEEDED
1476 2646 3004        DCA          SCNT
1477 2647 1047        TAD          SP
1478 2650 7104  LPSP,  CLL RAL
1479 2651 2004        ISZ          SCNT
1480 2652 7420        SNL
1481 2653 5250        JMP          LPSP
1482 2654 7650        SNA CLA
1483 2655 2004        ISZ          SCNT

```

1484	2656	1004	TAD	SCNT	
1485	2657	1376	TAD	(-13	
1486	2660	7041	CIA		/(LN(SP)+1)
1487					
1488	2661	6201	CDF	00	
1489	2662	4572	GETBUF		/DF:=3
1490	2663	7402	HLT		
1491	2664	3046	DCA	A	
1492					
1493	2665	7240	CLA	CMA	/ZERO BUFFER
1494	2666	1046	TAD	A	
1495	2667	3017	DCA	AUTIN8	
1496	2670	1047	TAD	SP	
1497	2671	7041	CIA		
1498	2672	3364	DCA	INICNT	
1499	2673	3417	DCA	I	AUTIN8
1500	2674	2364	ISZ	INICNT	
1501	2675	5273	JMP	, -2	
1502					
1503	2676	6201	CDF	00	
1504	2677	1375	TAD	(-11	
1505	2700	3364	DCA	INICNT	
1506	2701	1374	TAD	(XPP	
1507	2702	3047	DCA	SPTR	
1508	2703	1046	TAD	A	
1509	2704	3447	ST1, DCA	I	SPTR /ALLOCATE BUFFERS
1510	2705	1002	TAD	X8	
1511	2706	1447	TAD	I	SPTR
1512	2707	2047	ISZ	SPTR	
1513	2710	2364	ISZ	INICNT	
1514	2711	5304	JMP	ST1	
1515	2712	3051	DCA	R	/MULTIPLEXER BUFFER
1516	2713	1365	TAD	DR	
1517	2714	1051	TAD	R	
1518	2715	3030	DCA	RR	/PARAMETERS MULTIPLEXER
1519	2716	1003	TAD	MX8	
1520	2717	3364	DCA	INICNT	
1521	2720	7240	CLA	CMA	
1522	2721	1032	TAD	XPP	
1523	2722	3017	DCA	AUTIN8	
1524	2723	7305	ST2, CLA	CLL	IAC RAL /CREATE XP-BUFFER
1525	2724	6201	CDF	00	
1526	2725	4572	GETBUF		
1527	2726	7402	HLT		
1528	2727	3417	DCA	I	AUTIN8 /INITIALIZE XPP[0:X8]
1529	2730	2364	ISZ	INICNT	
1530	2731	5323	JMP	ST2	
1531	2732	7307	CLA	CLL	IAC RTL /INITIALIZE QUEUE FOR PX-BUFFER POINTERS
1532	2733	6201	CDF	00	
1533	2734	4572	GETBUF		
1534	2735	7402	HLT		
1535	2736	4773	JMS	SETIF	
1536					
1537			/INITIALIZE MULTIPLEXER		
1538	2737	1057	TAD	MILP	

```
1539 2740 7041 CIA
1540 2741 3057 DCA MILP
1541 2742 4772 JMS SETTI
1542 2743 4771 JMS INIMP
1543 2744 6424 T1CN /START CLOCK
1544
1545 /START INTERFACE X8
1546 2745 6322 OKX
1547 /ACM2
1548 /DCA X7FLAG
1549
1550 2746 4770 JMS INUSER /MAKE CONNECTION PS-8 AND UTOR
1551 /CLA CMA
1552 /DCA JSW /LEVEL BACKGROUND PROGRAM
1553 2747 6001 ION
1554 /BACKGROUND PROGRAM, ROTATES AC LIGHTS
1555 /TRY TO GIVE HIM A NICE PLACE
1556 2750 0000 IFZERO 2760-.,^4000 <ZBLOCK 2760-., >
1557 2760 7024 CML RAL
1558 2761 2024 ISZ DATAPN
1559 2762 5361 JMP .-1
1560 2763 5360 JMP .-3
1561
1562 2764 0000 INICNT, 0
1563 2765 0000 DR, 0
1564
1565 SCNT= SV
1566 SP= CHAR
1567 SPTR= SP
1568
1569 HDTTI= START+10 /WRITE TTI INSTRUCTIONS OVER START
1570 2770 4600
1571 2771 2342
1572 2772 2520
1573 2773 1747
1574 2774 0032
1575 2775 7767
1576 2776 7765
1577 2777 2210
1578
```

```

1579 3000 7201 X7S, CLA IAC
1580 3001 6327 CBUX7 OKX CAX
1581 3002 7200 OUT7S, CLA
1582 3003 2050 ISZ X7FLAG
1583 3004 5502 JMP I [EXKI
1584 /TAD (HLT
1585 /DCA OUT7S /HALT ON LS
1586 3005 7305 AC2 /PRICRITY INTERRUPT ROUTINE
1587 3006 4567 INITIR /INITIALIZE IR
1588 3007 1003 TAD MX8
1589 3010 3234 DCA X7P1
1590 3011 6231 AGPR, CDF 30 /ASK XP-BUFFER
1591 3012 4565 GETNODE
1592 3013 3023 DCA DTPNMJ
1593 3014 1023 TAD DTPNMJ /DATA POINTER
1594 3015 3013 DCA SPN
1595 3016 1234 TAD X7P1 /-(TERMINAL NO + 1)
1596 3017 3413 DCA I SPN
1597 3020 3413 DCA I SPN /NILL
1598 3021 7305 AC2
1599 3022 3022 DCA JSWMJ /PRIORITY
1600 3023 10/2 TAD [PXOVER
1601 3024 4573 MAKE JOB /HANDLE TO X8
1602 3025 6001 ION
1603 3026 2234 ISZ X7P1
1604 3027 5211 JMP AGPR
1605 3030 5576 EXIT
1606
1607 3031 7201 P7S, CLA IAC
1608 3032 6353 CBUP7 OKP
1609 3033 5202 JMP OUT7S
1610 3034 0000 X7P1, 0
1611
1612 3035 0000 ACBUF, 0
1613 3036 4510 JMS I [GTPXP /GET POINTERS
1614 3037 4507 JMS I [FREEBF /RETURN TO POOL
1615 3040 4523 JMS I [NXTPIXB /NEXT PX BUFFER
1616 3041 4522 JMS I [STXPX /STORE POINTERS
1617 3042 4531 JMS I [GTLCH /LAST CHARACTER
1618 3043 1047 TAD CHAR
1619 3044 4547 JMS I [PX Echo /TO X8
1620 3045 5635 JMP I ACBUF
1621
1622 3046 0000 FIRST, 0
1623 3047 7104 CLL RAL
1624 3050 7510 SPA
1625 3051 5262 JMP COLOUR
1626 3052 7106 CLL RTL
1627 3053 7006 RTL
1628 3054 13/7 TAD (HDTRB-1
1629 3055 32/7 DCA SVF
1630 3056 16/7 TAD I SVF
1631 3057 47/6 JMS FILLRB
1632 3060 1162 TAD [DUMMY
1633 3061 5646 JMP I FIRST

```

```

1634
1635      3062  7200  COLOUR, CLA
1636      3063  1040      TAD      STATE
1637      3064  1060      TAD      SAVLN
1638      3065  3277      DCA      SVF
1639      3066  7344      CLA CLL CMA RAL
1640      3067  1677      TAD I    SVF
1641      3070  7640      SZA CLA
1642      3071  1375      TAD      (RED-BLCK
1643      3072  1374      TAD      (BLCK
1644      3073  4776      JMS      FILLRB
1645      3074  1373      TAD      (1777
1646      3075  5646      JMP I    FIRST
1647
1648      3076  0000  MCH,      0
1649      3077  0000  SVF,      0
1650
1651      3100  0000  REQUA,      0      /ASK XP-BUFFER
1652      3101  4565      GETNODE
1653      3102  3023      DCA      DTPNMJ
1654      3103  1023      TAD      DTPNMJ
1655      3104  3013      DCA      SPN
1656      3105  2021      ISZ      JSW      /INCREASE LEVEL OF CURRENT JOB
1657      3106  7305      CLA CLL IAC RAL
1658      3107  3022      DCA      JSWMJ
1659      3110  1025      TAD      PAR1
1660      3111  7040      CMA      /NEGATIVE TERMINAL NUMBER
1661      3112  3413      DCA I    SPN
1662      3113  3413      DCA I    SPN      /NILL
1663      3114  1072      TAD      [PXOVER
1664      3115  4573      MAKE JOB
1665      3116  6001      ION
1666      3117  5700      JMP I    REQUA
1667
1668      GETNODE=JMS I    [.      /FROM THE AVAIL LIST
1669      3120  0000  GETND,      0
1670      3121  6002      IOF
1671      3122  7240      CLA CMA
1672      3123  1055      TAD      WARN
1673      3124  3055      DCA      WARN
1674      3125  1020      TAD      AVAIL
1675      3126  7450      SNA
1676      3127  5335      JMP      NOROOM /AVAIL LIST EXHAUSTED
1677      3130  3004      DCA      SV
1678      3131  1420      TAD I    AVAIL
1679      3132  3020      DCA      AVAIL
1680      3133  1004      TAD      SV
1681      3134  5720      JMP I    GETND
1682      3135  6201  NOROOM, CDF 0
1683      3136  4572      GETBUF
1684      3137  7402      HLT      /EMPTY TOO
1685      3140  5720      JMP I    GETND
1686
1687      3141  0000  FREEND,      0      /TO THE AVAIL LIST
1688      3142  6002      IOF

```

1689 3143 3004
1690 3144 1004
1691 3145 7112
1692 3146 7770
1693 3147 7402
1694 3150 1020
1695 3151 3404
1696 3152 1004
1697 3153 3020
1698 3154 2055
1699 3155 7000
1700 3156 5741
1701 3173 1777
1702 3174 0216
1703 3175 7601
1704 3176 2563
1705 3177 0757
1706

DCA SV /BUFFER POINTER
TAD SV
CLL RTR
SZL SPA SNA CLA
HLT
TAD AVAIL
DCA I SV /SUCCESSOR OF SV
TAD SV
DCA AVAIL /NEW FIRST FREE NODE
ISZ WARN
NOP
JMP I FREEND

PAGE

```

1707 3200 0000 TRTURN, 0 /TEST TURN OVER CONDITIONS
1708 3201 4552 JMS I [GTPRT /PARTNER=X8?
1709 3202 5233 JMP TURXIT /NO, KEEP TURN
1710 3203 4525 JMS I [GTB1 /LAST LINE WAS DELETED
1711 3204 1152 TAD [DELETE
1712 3205 4533 JMS I [NEQUAL
1713 3206 5233 JMP TURXIT /YES, KEEP TURN
1714 3207 4545 JMS I [GTXPMPR /XP BUFFER PRESENT?
1715 3210 7344 CLA CLL CMA RAL
1716 3211 1047 TAD CHAR
1717 3212 7710 SPA CLA
1718 3213 5233 JMP TURXIT /NO, KEEP TURN
1719 3214 4540 JMS I [CHANGE /TURN OVER
1720 3215 4536 JMS I [SWITCH
1721 3216 4540 JMS I [CHANGE
1722 3217 1032 TAD XPP /XP POINTERS
1723 3220 4511 JMS I [AMAL
1724 3221 3026 DCA PAR2
1725 3222 7132 STL RTR
1726 3223 1426 TAD I PAR2
1727 3224 3426 DCA I PAR2
1728 3225 1033 TAD XPNUMB
1729 3226 4511 JMS I [AMAL
1730 3227 7041 CIA /DECREMENT AC
1731 3230 7040 CMA
1732 3231 3027 DCA PAR3
1733 3232 4564 ECHO
1734 3233 1134 TURXIT, TAD [LF
1735 3234 3047 DCA CHAR
1736 3235 5600 JMP I TRTURN
1737
1738 3236 0000 SEND, 0
1739 3237 1130 TAD [-14
1740 3240 1026 TAD PAR2
1741 3241 3026 DCA PAR2
1742 3242 4565 GETNODE
1743 3243 3023 DCA DTPNMJ /DATA AREA
1744 3244 2021 ISZ JSW /INCREASE LEVEL OF CURRENT JOB
1745 3245 7305 CLA CLL IAC RAL
1746 3246 3022 DCA JSWMJ /JOB STATUS WORD
1747 3247 1023 TAD DTPNMJ
1748 3250 3013 DCA SPN /POINTS TO DATA AREA
1749 3251 1025 TAD PAR1 /STORE
1750 3252 3413 DCA I SPN /PAR1
1751 3253 1026 TAD PAR2 /PAR2
1752 3254 3413 DCA I SPN
1753 3255 1072 TAD [PXOVER
1754 3256 4513 MAKE JOB /HANDLE BUFFER TO X8
1755 3257 6001 ION
1756 3260 4523 JMS I [NXTPXB /NEXT PX BUFFER
1757 3261 5636 JMP I SEND
1758
1759
1760 3262 0000 STAMP, 0 /ADD CHARACTER TO PX BUFFER
1761 3263 1026 TAD PAR2

```

1762	3264	3043	DCA	SP1	/SAVE CHARACTER
1763	3265	4510	JMS I	[GTPXP	/GET PX POINTERS
1764	3266	2026	ISZ	PAR2	
1765	3267	1043	TAD	SP1	
1766	3270	3426	DCA I	PAR2	
1767	3271	2027	ISZ	PAR3	/BUFFER FULL?
1768	3272	2262	ISZ	STAMP	/NO, RETURN+1
1769	3273	5662	JMP I	STAMP	
1770					
1771	3274	4262	DUPLX2, JMS	STAMP	/STORE CHARACTER
1772	3275	4236	JMS	SEND	/SEND BUFFER, IF ANY
1773	3276	4522	JMS I	[STPXP	/RESTORE POINTERS
1774	3277	5576	EXIT		
1775					
1776	3300	4262	SINGLE, JMS	STAMP	/STORE SINGLE CHARACTER
1777	3301	1377	TAD	(13	/SEND BUFFER AWAY
1778	3302	5275	JMP	SINGLE-3	

```

1779      /DYNAMIC STORAGE ALLOCATION CORE
1780      /BUDDY SYSTEM AS DESCRIBED IN
1781      /KNUTH, THE ART OF COMPUTER PROGRAMMING (VOL 1)
1782
1783      FLD=    30
1784      CK=     SV3
1785
1786      3303  0000  GETBIT, 0
1787      3304  7112      CLL RTR
1788      3305  3004      DCA      SV
1789      3306  1004      TAD      SV
1790      3307  7110      CLL RAR
1791      3310  7110      CLL RAR
1792      3311  7110      CLL RAR
1793      /TAD      (HDADSP
1794      3312  3005      DCA      WA      /WORD ADDRESS
1795      3313  1004      TAD      SV
1796      3314  0163      AND      [7
1797      /TAD      (HDMASK
1798      3315  3006      DCA      BA      /BIT ADDRESS
1799      3316  5703      JMP I    GETBIT
1800
1801      3317  0000  INSERT, 0      /SV AFTER SV2 IN A DOUBLY LINKED LIST
1802      3320  3004      DCA      SV      /POINTER TO THE NODE TO BE INSERTED
1803      3321  1406      TAD I    SV2
1804      3322  7450      SNA      /EMPTY LIST?
1805      3323  5332      JMP      ,+7      /YES
1806      3324  7001      IAC
1807      3325  3005      DCA      SV1      /PREDECESSOR FIELD OF THE SUCCESSOR
1808      3326  1004      TAD      SV
1809      3327  3405      DCA I    SV1      /NEW PREDECESSOR      OF SV
1810      3330  7240      CLA CMA
1811      3331  1005      TAD      SV1
1812      3332  3404      DCA I    SV      /NEW SUCCESSOR OF SV
1813      3333  1004      TAD      SV
1814      3334  3406      DCA I    SV2      /SUCCESSOR OF SV2
1815      3335  2004      ISZ      SV
1816      3336  1006      TAD      SV2
1817      3337  3404      DCA I    SV      /PREDECESSOR OF SV
1818      3340  2004      ISZ      SV
1819      3341  1007      TAD      CK      /CURRENT SIZE
1820      3342  3404      DCA I    SV      /STORE
1821      3343  5717      JMP I    INSERT
1822
1823
1824      3344  0000  DELET, 0      /DELETE SV FROM A DOUBLY LINKED LIST
1825      3345  3004      DCA      SV
1826      3346  1404      TAD I    SV
1827      3347  3005      DCA      SV1      /SUCCESSOR
1828      3350  2004      ISZ      SV
1829      3351  1404      TAD I    SV
1830      3352  3006      DCA      SV2      /PREDECESSOR
1831      3353  1005      TAD      SV1
1832      3354  3406      DCA I    SV2      /NEW SUCCESSOR
1833      3355  1005      TAD      SV1

```

```

1834 3356 7650      SNA CLA      /EMPTY LIST?
1835 3357 5744      JMP I   DELET
1836 3360 1006      TAD      SV2
1837 3361 2005      ISZ      SV1
1838 3362 3405      DCA I   SV1      /NEW PREDECESSOR
1839 3363 5744      JMP I   DELET
1840
1841 3377 0013
1842
1843                PAGE
1843                /CALLING SEQUENCE
1844                /
1844                CDF CUR
1845                /
1845                TAD      BUFFER
1846                /
1846                FREEBUF
1847                /
1847                SIZECODE
1848                /
1849                FREEBUF=JMS I   [.
1850                FRBUF, 0
1851 3400 0000      IOF
1851 3401 6002      DCA      LOC      /ADDRESS BUFFER
1852 3402 3364      TAD I   FRBUF
1853 3403 1600      ISZ      FRBUF
1854 3404 2200      AND      (MSKSZ
1855 3405 0377      DCA      CK      /BUFFER SIZE
1856 3406 3007      TAD      [CIF
1857 3407 1071      RDF
1858 3410 6214      DCA      OUTFBF
1859 3411 3274      CDF FLD
1860 3412 6251      TAD      LOC
1861 3413 1364      CLL RTR
1862 3414 7112      SPA SNA SZL CLA
1863 3415 7770      HLT      /ERROR
1864 3416 7402      TAD      LOC
1865 3417 1364      JMS      GETBIT
1866 3420 4776      TAD I   BA
1867 3421 1406      CMA
1868 3422 7040      AND I   WA
1869 3423 0405      DCA I   WA      /OCCUPIED[LOC]=FALSE
1870 3424 3405      TAD      CK      /TRY TO MERGE
1871 3425 1007      /TAD      (HDMASK
1872                DCA      MPNT
1873 3426 3367      TAD I   MPNT
1874 3427 1767      AND      (777
1875 3430 0375      SNA      /MAXIMUM SIZE??
1876 3431 7450      JMP      INS      /YES
1877 3432 5267      AND      LOC      /EXOR(LOC, HDMASK[CK])
1878 3433 0364      CIA
1879 3434 7041      CLL RAL
1880 3435 7104      TAD      LOC
1881 3436 1364      TAD I   MPNT
1882 3437 1767      DCA      LOCBDY      /ADDRESS BUDDY
1883 3440 3365      TAD      LOCBDY
1884 3441 1365      JMS      GETBIT
1885 3442 4776      TAD I   BA
1886 3443 1406      AND I   WA
1887 3444 0405      SZA CLA      /BUDDY OCCUPIED?
1888 3445 7640

```

```

1889 3446 5267      JMP      INS      /YES.
1890 3447 7305      CLA CLL IAC RAL
1891 3450 1365      TAD      LOCB DY
1892 3451 3004      DCA      SV
1893 3452 1404      TAD I      SV
1894 3453 7041      CIA
1895 3454 1007      TAD      CK
1896 3455 7640      SZA CLA      /SIZE BUFFER=SIZE BUDDY
1897 3456 5267      JMP      INS      /NO
1898 3457 1365      TAD      LOCB DY
1899 3460 4774      JMS      DELET      /DELETE BUDDY FROM FREE LIST
1900 3461 1767      TAD I      MPNT
1901 3462 7040      CMA
1902 3463 0365      AND      LOCB DY
1903 3464 3364      DCA      LOC      /LOC:=MINIMUM(LOC, LOCB DY)
1904 3465 2007      ISZ      CK
1905 3466 5225      JMP      AGMREE
1906 3467 1373      INS,      TAD      (HDFSP
1907 3470 1007      TAD      CK
1908 3471 3006      DCA      SV2
1909 3472 1364      TAD      LOC
1910 3473 4772      JMS      INSERT /INSERT BUFFER IN FREE LIST
1911 3474 7000      OUTFBF, NOP
1912 3475 5600      JMP I      FRBUF
1913
1914      /CALLLING SEQUENCE:
1915      /      CDF CUR
1916      /      TAD      SIZECODE
1917      /      GETBUF
1918      /      ERROR
1919      /      NORMAL RETURN
1920      /
1921      /AC      SIZE
1922      /0      4
1923      /1      10
1924      /2      20
1925      /3      40
1926      /4      100
1927      /5      200
1928      /6      400
1929      /7      1000
1930      /
1931      GETBUF= JMS I      (.
1932 3476 0000      GETBF, 0
1933 3477 6002      IOF
1934 3500 3366      DCA      SIZE
1935 3501 1366      TAD      SIZE
1936 3502 0371      AND      (7777=MAXSZ
1937 3503 7440      SZA
1938 3504 7402      HLT      /OVERFLOW SIZE
1939 3505 1071      TAD      [CIF
1940 3506 6214      RDF
1941 3507 3362      DCA      OUTGB /PREPARE A PROPER RETURN
1942 3510 6251      CDF FLD
1943 3511 1366      TAD      SIZE

```

1944	3512	3007		DCA	CK	/CURRENT SIZE
1945	3513	1373	SRCHBF,	TAD	(HDFSP	
1946	3514	1007		TAD	CK	
1947	3515	3367		DCA	MPNT	
1948	3516	1767		TAD	MPNT	
1949	3517	7440		SZA		/ANY SPACE?
1950	3520	5327		JMP	GET	/YES
1951	3521	1007		TAD	CK	
1952	3522	1370		TAD	(-MAXSZ	
1953	3523	7700		SMA	CLA	/MAXIMUM SIZE
1954	3524	5362		JMP	OUTGB	/NO SPACE
1955	3525	2007		ISZ	CK	
1956	3526	5313		JMP	SRCHBF	/TRY AGAIN
1957	3527	3364	GET,	DCA	LOC	/ADDRESS BUFFER
1958	3530	1364		TAD	LOC	
1959	3531	4774		JMS	DELET	/DELETE BUFFER FROM FREE LIST
1960	3532	1007	SPLIT,	TAD	CK	
1961	3533	7041		CIA		
1962	3534	1366		TAD	SIZE	
1963	3535	7650		SNA	CLA	/CORRECT SIZE?
1964	3536	5353		JMP	FIN	/YES
1965	3537	7240		CLA	CMA	/DECREMENT SIZE
1966	3540	1007		TAD	CK	
1967	3541	3007		DCA	CK	
1968				/TAD	(HDMASK	
1969	3542	1007		TAD	CK	
1970	3543	3367		DCA	MPNT	/MASK
1971	3544	1373		TAD	(HDFSP	
1972	3545	1007		TAD	CK	
1973	3546	3006		DCA	SV2	/SELECT LIST
1974	3547	1364		TAD	LOC	
1975	3550	1767		TAD	MPNT	/SPLIT BUFFER AND
1976	3551	4772		JMS	INSERT	/INSERT ONE HALF IN FREE LIST
1977	3552	5332		JMP	SPLIT	
1978						
1979	3553	1364	FIN,	TAD	LOC	
1980	3554	4776		JMS	GETBIT	
1981	3555	1406		TAD	BA	
1982	3556	1405		TAD	WA	
1983	3557	3405		DCA	WA	/OCCUPIED[LOC]:=TRUE
1984	3560	2276		ISZ	GETBF	
1985	3561	1364		TAD	LOC	
1986	3562	7000	OUTGB,	NOP		
1987	3563	5676		JMP	GETBF	
1988						
1989	3564	0000	LOC,	0		
1990	3565	0000	LOCBDY,	0		
1991	3566	0000	SIZE,	0		
1992	3567	0000	MPNT,	0		
1993						
1994			WA=	SV1		
1995			BA=	SV2		
1996	3570	7771				
1997	3571	7770				
1998	3572	3317				

/ UTOR DD. 12/3/73

PALB 9/17/74 PAGE 16-4

1999	3573	0020
2000	3574	3344
2001	3575	0777
2002	3576	3303
2003	3577	0007
2004		

PAGE

```

2005      /JOB SECTION
2006
2007      /THE FOLLOWING CONDITIONS SHOULD HOLD:
2008      / * PRIORITY JOB>PRIORITY INTERRUPTED JOBS
2009      /** PRIORITY JOB >= PRIORITY JOB CHAIN
2010
2011      EXIT=    JMP I    [.
2012      3600    6001      DECIS,  ION                      /GIVE CPU THE OPPORTUNITY TO LISTEN
2013      3601    6231      DEC1,   CDF 30                  /BUT NOT ALWAYS
2014      3602    7300      CLA CLL
2015      3603    6002      IOF
2016      3604    3051      DCA      LOOK
2017      3605    1777      TAD I    (JOBCH
2018      3606    7450      SNA                      /ANY JOBS?
2019      3607    5574      REACTIVATE                  /NO
2020      3610    3013      DCA      SPN
2021      3611    1413      TAD I    SPN
2022      3612    0376      AND      (MSKPR
2023      3613    3300      DCA      PCP                /PRIORITY JOB CHAIN
2024      3614    1012      TAD      STACKP
2025      3615    3004      DCA      SV
2026      3616    1404      TAD I    SV
2027      3617    0376      AND      (MSKPR                /PRIORITY INTERRUPT STACK
2028      3620    7040      CMA
2029      3621    1300      TAD      PCP
2030      3622    7710      SPA CLA                      /PRIOR. JOB CHAIN>PRIOR. INTERRUPT
2031      3623    5574      REACTIVATE                  /NO
2032      3624    7344      NEXTJB, CLA CLL CMA RAL      /AC:=-2
2033      3625    1013      TAD      SPN
2034      3626    3013      DCA      SPN
2035      3627    1413      TAD I    SPN
2036      3630    3777      DCA I    (JOBCH
2037      3631    1300      TAD      PCP
2038      3632    1375      TAD      (TAILS
2039      3633    3355      DCA      LTAIL
2040      3634    1013      TAD      SPN
2041      3635    7041      CIA
2042      3636    1755      TAD I    LTAIL
2043      3637    7640      SZA CLA                      /RESET TAIL[PCP]?
2044      3640    5246      JMP      ,+6                /NO
2045      3641    1300      TAD      PCP
2046      3642    1374      TAD      (-MAXPR
2047      3643    7650      SNA CLA
2048      3644    1377      TAD      (JOBCH
2049      3645    3755      DCA I    LTAIL
2050
2051      /IF AANTAL JOBS(PCP) =0 THEN
2052      /      (IF PCP=MAX PRIORITY THEN TAIL[PCP]:=LOC(HEADL)
2053      /      ELSE      TAIL[PCP]:=0
2054      /      )
2055
2056      3646    1413      TAD I    SPN
2057      3647    7510      SPA                      /ERROR JSW
2058      3650    7402      HLT                      /YES
2059      3651    3021      DCA      JSW                /JOB STATUS WORD

```

2060	3652	1413	TAD I	SPN	
2061	3653	3356	DCA	SAJOB	/START ADDRESS
2062	3654	1413	TAD I	SPN	
2063	3655	3024	DCA	DATAPN	/DATA POINTER
2064	3656	7346	CLA CLL	CMA RTL	
2065	3657	1013	TAD	SPN	
2066	3660	4504	JMS I	[FREEND	/RETURN TO FREE LIST
2067	3661	6201	CDF 0		/FOR CANCEL
2068	3662	7325	CLA CLL	CML IAC RAL	/3
2069	3663	0021	AND	JSW	
2070	3664	7650	SNA CLA		
2071	3665	1053	TAD	CANCEL	
2072	3666	7640	SZA CLA		
2073	3667	4773	JMS	ER	
2074	3670	1021	TAD	JSW	
2075	3671	0070	AND	[70	/FIELD OF JOB
2076	3672	1067	TAD	[CIF CDF	
2077	3673	3274	DCA	.*1	
2078	3674	6203	CDF CIF		
2079	3675	7300	CLA CLL		
2080	3676	6001	ION		
2081	3677	5756	JMP I	SAJOB	
2082					
2083	3700	0000	PCP,	0	
2084					
2085			MAKE=JMS I [.		
2086			JOB=0		
2087	3701	0000	MAKEJB,	0	
2088	3702	6231	CDF 30		
2089	3703	4306	JMS	CRNODE	/CREATE A NODE
2090	3704	4323	JMS	INNODE	/INSERT THE NODE IN THE JOB CHAIN
2091	3705	5701	JMP I	MAKEJB	
2092					
2093	3706	0000	CRNODE,	0	
2094	3707	3356	DCA	SAJOB	
2095	3710	4565	GETNODE		
2096	3711	3013	DCA	SPN	
2097	3712	1022	TAD	JSWMJ	
2098	3713	3413	DCA I	SPN	/JOB STATUS WORD
2099	3714	1356	TAD	SAJOB	
2100	3715	3413	DCA I	SPN	/ADDRESS JOB
2101	3716	1023	TAD	DTPNMJ	
2102	3717	3413	DCA I	SPN	/DATA POINTER
2103	3720	7346	CLA CMA	CLL RTL	
2104	3721	1013	TAD	SPN	/ADDRESS OF THE NODE
2105	3722	5706	JMP I	CRNODE	
2106					
2107	3723	0000	INNODE,	0	/INSERT A NODE
2108	3724	2051	ISZ	LOOK	
2109	3725	3004	DCA	SV	/SAVE ADDRESS OF THE NODE
2110	3726	1022	TAD	JSWMJ	
2111	3727	7510	SPA		
2112	3730	7402	HLT		/ERROR IN JSW
2113	3731	0376	AND	(MSKPR	/PRIORITY JOB
2114	3732	1375	TAD	(TAILS	

2115	3733	3355		DCA	LTAIL	
2116	3734	1755		TAD I	LTAIL	/POINTER DEFINED?
2117	3735	7440		SZA		
2118	3736	5344		JMP	GETTL	/YES
2119	3737	1355		TAD	LTAIL	/NO, SEARCH
2120	3740	3013		DCA	SPN	
2121	3741	1413		TAD I	SPN	/AUTO INDEXER
2122	3742	7450		SNA		
2123	3743	5341		JMP	.-2	
2124	3744	3354	GETTL,	DCA	POETS	/ENTRY POINT JOB CHAIN
2125	3745	1754		TAD I	POETS	/INSERT NODE IN JOB CHAIN
2126	3746	3404		DCA I	SV	/POINTS TO NEXT JOB
2127	3747	1004		TAD	SV	
2128	3750	3754		DCA I	POETS	
2129	3751	1004		TAD	SV	
2130	3752	3755		DCA I	LTAIL	/RESET ENTRYPOINT JOB CHAIN[PRIORITY JOB]
2131	3753	5723		JMP I	INNODE	
2132						
2133	3754	0000	POETS,	0		
2134	3755	0000	LTAIL,	0		
2135	3756	0000	SAJOB,	0		
2136	3773	5263				
2137	3774	7775				
2138	3775	0030				
2139	3776	4003				
2140	3777	0200				
2141						

PAGE

```

2142          REACTIVATE=JMP I      [.
2143 4000 7300 UNSTACK,CLA CLL      /RESTORE STATUS AFTER INTERRUPT
2144 4001 6231          CDF 30
2145 4002 1076          TAD      [-5
2146 4003 1012          TAD      STACKP
2147 4004 3012          DCA      STACKP
2148 4005 1412          TAD I      STACKP
2149 4006 3062          DCA      SVLINK /LINK, IF, DF
2150 4007 1412          TAD I      STACKP
2151 4010 3061          DCA      SVAC /SAVE AC
2152 4011 1412          TAD I      STACKP
2153 4012 3000          DCA      0 /PC
2154 4013 1412          TAD I      STACKP
2155 4014 3024          DCA      DATAPN /DATA POINTER
2156 4015 1412          TAD I      STACKP
2157 4016 3021          DCA      JSW /JOB STATUS WORD
2158 4017 1076          TAD      [-5
2159 4020 1012          TAD      STACKP
2160 4021 3012          DCA      STACKP /DECREASE STACK POINTER
2161 4022 4777          JMS      TSTCAN /TEST CANCEL FLAG
2162 4023 1062          TAD      SVLINK
2163 4024 0070          AND      [70
2164 4025 1071          TAD      [CIF
2165 4026 3227          DCA      .+1
2166 4027 6202          CIF      /RESTORE INSTRUCTION FIELD
2167 4030 1062          TAD      SVLINK
2168 4031 7006          RTL
2169 4032 7004          RAL
2170 4033 0070          AND      [70
2171 4034 1066          TAD      [CDF
2172 4035 3236          DCA      .+1
2173 4036 6201          CDF      /RESTORE DATA FIELD
2174 4037 1062          TAD      SVLINK
2175 4040 7104          CLL RAL /RESTORE LINK
2176 4041 7200          CLA
2177 4042 1061          TAD      SVAC /AND AC
2178 4043 6001          ION
2179 4044 5400          JMP I 0
2180
2181 4045 6001 EXKI, ION
2182 4046 6272          CIF 70 /PREVENT INTERRUPTS
2183 4047 7300          CLA CLL
2184 4050 6244          RMF      /RESTORE IF, DF
2185 4051 1062          TAD      SVLINK
2186 4052 7004          RAL      /RESTORE LINK
2187 4053 1061          TAD      SVAC
2188 4054 5400          JMP I 0
2189
2190 4055 3061 INSPEC, DCA      SVAC /SAVE AC
2191 4056 7010          RAR
2192 4057 3062          DCA      SVLINK /AND LINK
2193 4060 6234          RIB      /READ INTERRUPT BUFFER
2194 4061 0070          AND      [70
2195 4062 1066          TAD      [CDF
2196 4063 3264          DCA      .+1

```

2197	4064	6201	CDF		
2198	4065	1400	TAD I	0	/NEXT INSTRUCTION TO BE EXECUTED IOF?
2199	4066	1376	TAD	(-IOF	
2200	4067	7650	SNA	CLA	
2201	4070	5246	JMP	EXKI+1	/DO NOT TURN ON IE
2202	4071	3022	DCA	JSWMJ	/NO
2203	4072	6421	T1SKP		
2204	4073	7410	SKP		
2205	4074	5775	JMP I	(T1S	/MULTIPLEXER
2206	4075	6301	BUX2		
2207	4076	7410	SKP		
2208	4077	5774	JMP I	(XP2	/XP2 FLAG
2209	4100	6331	BUP2		
2210	4101	7410	SKP		
2211	4102	5773	JMP I	(PX2	/PX2 FLAG
2212	4103	6311	BUX7		
2213	4104	7410	SKP		
2214	4105	5772	JMP I	(X7S	/XP7 FLAG
2215	4106	6341	BUP7		
2216	4107	7410	SKP		
2217	4110	5771	JMP I	(P7S	/PX7 FLAG
2218	4111	6623	6623		
2219	4112	7410	SKP		
2220	4113	4770	JMS	RF08	/DISK
2221	4114	6031	KSF		
2222	4115	7410	SKP		
2223	4116	4767	JMS I	(KB	/KEYBOARD
2224	4117	6041	TSF		
2225	4120	7410	SKP		
2226	4121	4766	JMS I	(TTY	/TELETYPE
2227	4122	6771	DTSF		
2228	4123	7410	SKP		
2229	4124	4765	JMS I	(DTA	/DECTAPE
2230	4125	6071	SRF		
2231	4126	7410	SKP		
2232	4127	4764	JMS	SCOPE	/SCOPE
2233	4130	6501	PLSF		
2234	4131	7410	SKP		
2235	4132	5763	JMP	PLOTR	/PLOTTER
2236	4133	6011	RSF		
2237	4134	7410	SKP		
2238	4135	5762	JMP	HSR	/HIGH SPEED READER
2239	4136	6021	PSF		
2240	4137	7410	SKP		
2241	4140	5761	JMP	HSP	/HIGH SPEED PUNCH
2242	4141	6254	SINT		/USER INTERRUPT?
2243	4142	7410	SKP		
2244	4143	5355	JMP	CLEARU	/ALMOST IMPOSSIBLE
2245	4144	6051	SNC		
2246	4145	5353	JMP	CLEAR	/CURSOR
2247	4146	6374	RIS		/READ INTERRUPT STATUS
2248	4147	7510	SPA		/INTERRUPT PENDING?
2249	4150	7412	HLT SKP		/YES, STRANGE DEVICE
2250	4151	4760	JMS	ER	/NO, SOFTWARE ERROR
2251	4152	5245	JMP	EXKI	

2252						
2253	4153	6052	CLEAR,	CCF		/CLEAR CURSOR FLAG
2254	4154	5245		JMP	EXKI	
2255						
2256	4155	6204	CLEARU,	CINT		
2257	4156	4567		INITIR		
2258	4157	4760		JMS	ER	
2259	4160	5263				
2260	4161	4473				
2261	4162	4532				
2262	4163	4430				
2263	4164	4535				
2264	4165	4544				
2265	4166	4444				
2266	4167	4711				
2267	4170	4453				
2268	4171	3031				
2269	4172	3000				
2270	4173	1600				
2271	4174	2000				
2272	4175	2200				
2273	4176	1776				
2274	4177	4217				
2275						

PAGE

```

2276          SAVE=   JMS I   [.
2277 4200 0000  STACK, 0          /SAVE STATUS AFTER INTERRUPT
2278 4201 6231          CDF 30
2279 4202 7300          CLA CLL
2280 4203 1062          TAD      SVLINK
2281 4204 6234          RIB
2282 4205 3412          DCA I   STACKP /LINK, INSTRUCTION FIELD, DATA FIELD
2283 4206 1061          TAD      SVAC
2284 4207 3412          DCA I   STACKP /AC
2285 4210 1000          TAD      0
2286 4211 3412          DCA I   STACKP /PROGRAM COUNTER
2287 4212 1024          TAD      DATAPN
2288 4213 3412          DCA I   STACKP /DATA POINTER
2289 4214 1021          TAD      JSW
2290 4215 3412          DCA I   STACKP /JOB STATUS WORD
2291 4216 5600          JMP I   STACK
2292
2293 4217 0000  TSTCAN, 0
2294 4220 7325          AC3
2295 4221 0021          AND      JSW    /MASK PRIORITY
2296 4222 7640          SZA CLA
2297 4223 5617          JMP I   TSTCAN /DONT CANCEL ANY JOB
2298 4224 1053          TAD      CANCEL /EXCEPT PRIORITY 0
2299 4225 7650          SNA CLA
2300 4226 5617          JMP I   TSTCAN
2301 4227 3053          DCA      CANCEL /CLEAR CANCEL FLAG
2302 4230 4777          JMS      ER     /REPORT
2303
2304 4231 0000  TODISK, 0          /CALLED BY FILLR IF XLINE IS SET
2305 4232 7305          AC2
2306 4233 3022          DCA      JSWMJ
2307 4234 1456          TAD I   MCHAR  /GET CHAR
2308 4235 3023          DCA      DTPNMJ /SAVE
2309 4236 3456          DCA I   MCHAR  /CLEAR, IT DOESNT GO TO THE TERMINAL
2310 4237 1376          TAD      (TDSK  /BUT TO THE DISK
2311 4240 4573          MAKE JOB
2312 4241 5631          JMP I   TODISK
2313
2314 4242 1024  TDSK,  TAD      DATAPN /GET CHAR
2315 4243 6201          CDF 0         /ENABLE PROPER RETURN
2316 4244 6212          CIF 10        /START UP THE PROGRAM THAT ASKED FOR IT
2317 4245 4775          JMS I   (ROCHAR /BY CHANGING XLINE
2318 4246 5576          EXIT          /READY
2319          ROCHAR= 3200
2320 4375 3200
2321 4376 4242
2322 4377 5263
2323

```

```

2324      IFNZRO .-4400 <THIS IS AN ASSEMBLY ERROR>
2325      4400 0000 PLOUT, 0 /CALLED BY ANY PROGRAM
2326      4401 6002 IOF /CRITICAL
2327      4402 3244 DCA PLTEM /AC CONTAINS PLOTINSTRUCTIONS
2328      4403 1234 TAD PLINS
2329      4404 7640 SZA CLA
2330      4405 4777 JMS ER /SOMEONE ELSE IS PLOTTING
2331      4406 1200 TAD PLOUT
2332      4407 3243 DCA PLEX+2 /SAVE RETURN ADDRESS
2333      4410 1244 TAD PLTEM
2334      4411 7112 CLL RTR
2335      4412 7012 RTR
2336      4413 7012 RTR
2337      4414 0376 AND (77
2338      4415 1375 TAD (6500
2339      4416 3235 DCA PLINS+1 /SECOND INSTRUCTION
2340      4417 1244 TAD PLTEM
2341      4420 0376 AND (77
2342      4421 7440 SZA /ZERO IF END OF PLOTTING
2343      4422 1375 TAD (6500
2344      4423 3234 DCA PLINS /FIRST INSTRUCTION
2345      4424 6214 RDF
2346      4425 1067 TAD [CIF CDF
2347      4426 3241 DCA PLEX /RETURN
2348      4427 5576 EXIT
2349
2350      4430 6502 PLOTR, PLCF
2351      4431 1234 TAD PLINS
2352      4432 7650 SNA CLA /SOMETHING TO PLOT?
2353      4433 5575 EXITS /KILL THE PLOTTING PROG
2354                                     /OR IGNORE INTERRUPT
2355      4434 0000 PLINS, 0;0
2356      4435 0000
2357      4436 3234 DCA PLINS
2358      4437 7001 IAC /THIS IS ALLOWABLE ONLY IF THE PLOTTING
2359                                     /PROGRAM TAKES VIRTUALLY NO TIME TO
2360                                     /COMPUTE ITS NEXT INSTRUCTION
2361      4440 4567 INITIR
2362      4441 6203 PLEX, CDF CIF
2363      4442 5643 JMP I .+1
2364      4443 0000 0
2365
2366      PLTEM=
2367      4444 0000 TTY, 0
2368      4445 6374 RIS /READ INTERRUPT STATUS
2369      4446 0073 AND [1000
2370      4447 7650 SNA CLA /WAS FLAG ENABLED TO INTERRUPT BUS
2371      4450 5644 JMP I TTY /NO, SOMEONE ELSE CAUSED THE INTERRUPT
2372      4451 6042 TCF /YES, CLEAR FLAG
2373      4452 5575 EXITS
2374
2375      4453 0000 RF08, 0
2376      4454 6616 /READ STATUS
2377      4455 0374 AND (700
2378      4456 7650 SNA CLA /ANY INTERRUPT ENABLED?

```

```

2379 4457 5653 JMP I RF08 /NO, THIS INTERRUPT WAS NOT CAUSED
2380 4460 6621 6621
2381 4461 7410 SKP BY THE D 3K
2382 4462 5266 JMP RF08ER
2383 4463 6611 6611
2384 4464 6601 6601 /CLEAR ALL FLAGS
2385 4465 5575 EXITSI
2386 4466 6611 RF08ER, 6611
2387 4467 6601 6601
2388 4470 7200 CLA
2389 4471 4567 INITIR
2390 4472 4777 JMS ER
2391
2392 4473 6022 HSP, PCF
2393 4474 4567 INITIR /EXITSI
2394 4475 6203 PUEX, CDF CIF 0
2395 4476 5700 JMP I PUNCH
2396
2397 IFZERO ,-4500-4000 <ASSEMBLY ERROR!!>
2398 *4500
2399 4500 2236 PUNCH, EXITKI /THE FLAG MIGHT HAVE BEEN UP WHEN WE
2400 4501 6002 IOF STARTED
2401 4502 6026 PLS
2402 4503 7200 CLA
2403 4504 6214 RDF
2404 4505 1067 TAD [CIF CDF
2405 4506 3275 DCA PUEX
2406 4507 5576 EXIT
2407
2408 IFNZRO ,-4510 <ASSEMBLY ERROR!!>
2409 4510 2236 READ, EXITKI
2410 4511 7240 CLA CMA
2411 4512 3331 DCA RCHAR
2412 4513 3330 DCA RTIM
2413 4514 6014 RFC /IF THE CHAR IS PRESENT ALREADY
2414 /THIS CAUSES AN INTERRUPT IMMEDIATELY
2415 4515 7200 RDLP, CLA
2416 4516 6214 RDF
2417 4517 1067 TAD [CIF CDF
2418 4520 3326 DCA REEX
2419 4521 1331 TAD RCHAR
2420 4522 7500 SMA /STILL NEGATIVE?
2421 4523 5326 JMP REEX /NO, WEVE GOT IT
2422 4524 2330 ISZ RTIM
2423 4525 5315 JMP RDLP /CONTINUE WAITING
2424 4526 6203 REEX, CDF CIF 0
2425 4527 5710 JMP I READ
2426
2427 4530 0000 RTIM, 0
2428 4531 0000 RCHAR, 0
2429
2430 4532 6012 HSR, RRB
2431 4533 3331 DCA RCHAR
2432 4534 5575 EXITSI
2433

```

2434	4535	0000	SCOPE,	0
2435	4536	6374	RIS	
2436	4537	0117	AND	[400
2437	4540	7650	SNA CLA	
2438	4541	5735	JMP I	SCOPE
2439	4542	6072	CRF	
2440	4543	5575	EXITS	
2441				
2442	4544	0000	DTA,	0
2443	4545	6761	DTRA	
2444	4546	0373	AND	(4
2445	4547	7650	SNA CLA	
2446	4550	5744	JMP I	DTA
2447	4551	1163	TAD	[7
2448	4552	6764	DTXA	
2449	4553	5575	EXITS	
2450	4573	0004		
2451	4574	0700		
2452	4575	6500		
2453	4576	0077		
2454	4577	5263		
2455				

PAGE

2456	4600	0000	INUSER,	0		
2457	4601	7201		CLA	IAC	
2458	4602	6366		KIE	TIE	
2459	4603	6201		CDF	00	
2460	4604	6212		CIF	10	
2461	4605	4777		JMS	I (7700	
2462	4606	0002		2		/LOOKUP
2463	4607	5326	US0,	USER		/OWN VERSION OF USER SERVICE ROUTINE
2464	4610	0000		0		
2465	4611	4305		JMS	EXUS	/NOT FOUND
2466	4612	7201		CLA	IAC	
2467	4613	6212		CIF	10	
2468	4614	4777		JMS	I (7700	
2469	4615	0002		2		/LOOKUP
2470	4616	5322	US1,	CDEC		/OWN VERSION OF COMMAND DECODER
2471	4617	0000		0		
2472	4620	4305		JMS	EXUS	
2473	4621	1207		TAD	US0	
2474	4622	7001		IAC		
2475	4623	6211		CDF	10	
2476	4624	3776		DCA	I (7725	/TELL THE SYSTEM WHERE TO FIND THIS USR
2477	4625	6201		CDF	00	
2478	4626	7327		CLA	CLL CML IAC RTL	
2479	4627	4572		GETBUF		
2480	4630	7402		HLT		
2481	4631	3240		DCA	US2	
2482	4632	1207		TAD	US0	
2483	4633	7001		IAC		
2484	4634	3241		DCA	US2+1	
2485	4635	6203		CIF	CDF 00	
2486	4636	4775		JMS	I (7607	/SWAP USR IN
2487	4637	0230		0230		
2488	4640	0000	US2,	0		
2489	4641	0000		0		
2490	4642	4305		JMS	EXUS	/INPUT ERROR
2491	4643	6231		CDF	30	
2492	4644	1240		TAD	US2	
2493	4645	1374		TAD	(267	
2494	4646	3327		DCA	RDPTR	
2495	4647	1216		TAD	US1	
2496	4650	7001		IAC		
2497	4651	3727		DCA	I RDPTR	/NOTE PLACE OF COMMAND DECODER
2498	4652	1240		TAD	US2	
2499	4653	3261		DCA	US3	
2500	4654	1241		TAD	US2+1	
2501	4655	3262		DCA	US3+1	
2502	4656	6203		CIF	CDF 00	
2503	4657	4775		JMS	I (7607	/SWAP USR BACK
2504	4660	4230		4230		
2505	4661	0000	US3,	0		
2506	4662	0000		0		
2507	4663	4305		JMS	EXUS	/OUTPUT ERROR
2508	4664	1240		TAD	US2	
2509	4665	4571		FREEBUF		
2510	4666	0006		6		

2511	4667	1373	TAD	(7604	/NEW SYS HANDLER
2512	4670	3011	DCA	11	
2513	4671	1372	TAD	(PATCH=1	
2514	4672	3010	DCA	10	
2515	4673	6201	CDF	00	
2516	4674	1410	TAD I	10	
2517	4675	3411	DCA I	11	
2518	4676	2330	ISZ	SCNTR	
2519	4677	5274	JMP	.-3	
2520	4700	3771	DCA I	(RDCOMM	
2521	4701	1370	TAD	(JMP I	(RDCOMM/PROTECTION AGAINST STARTING A
2522	4702	3767	DCA I	(INIT	/AT 200
2523	4703	3325	DCA	MODKB	PROGRAM
2524	4704	7410	SKP		
2525	4705	0000	EXUS,	0	
2526	4706	2054	ISZ	LISTEN	
2527	4707	6362	KIE		
2528	4710	5600	JMP I	INUSER	
2529					
2530	4711	0000	KB,	0	
2531	4712	6374	RIS		
2532	4713	0366	AND	(2000	
2533	4714	7650	SNA CLA		
2534	4715	5711	JMP I	KB	
2535	4716	6036	KRB		
2536	4717	1365	TAD	(-203	
2537	4720	7650	SNA CLA		
2538	4721	1054	TAD	LISTEN	
2539	4722	7650	SNA CLA		
2540	4723	5575	EXITS		
2541	4724	4567	INITIR		
2542	4725	4764	MODKB,	JMS	ER
2543	4726	5771	JMP	RDCOMM	
2544					
2545	4727	0000	RDPTR,	0	
2546	4730	7643	SCNTR,	PATCH-SPEND	
2547	4764	5263			
2548	4765	7575			
2549	4766	2000			
2550	4767	0200			
2551	4770	5465			
2552	4771	5000			
2553	4772	5377			
2554	4773	7604			
2555	4774	0267			
2556	4775	7607			
2557	4776	7725			
2558	4777	7700			
2559					

2560	5000	5576	RDCOMM,	EXIT	
2561	5001	6361		KTID	
2562	5002	3054		DCA	LISTEN
2563	5003	6201		CDF 0	
2564	5004	1777		TAD 1	(PLINS /ARE WE PLOTTING?
2565	5005	7650		SNA CLA	
2566	5006	6371		SCID	/NO, DISABLE SCOPE, CURSOR AND PLOTTER
2567					/FROM INTERRUPT BUS
2568	5007	3021		DCA	JSW
2569	5010	6212		CIF 10	
2570	5011	4776		JMS 1	(7700
2571	5012	0010		10	
2572	5013	6212		CIF 10	
2573	5014	4776		JMS 1	(7700 /TRICK
2574	5015	0010		10	
2575	5016	4775	RDN,	JMS	CRL
2576	5017	1374		TAD	(300
2577	5020	4773		JMS	PRINT
2578	5021	4306		JMS	RDNNAME
2579	5022	3227		DCA	BLI
2580	5023	6212		CIF 10	
2581	5024	7201		CLA IAC	
2582	5025	4464		JMS 1	(200
2583	5026	0002		2	/LOOKUP
2584	5027	0000	BLI,	0	
2585	5030	0000		0	
2586	5031	4772		JMS	ER
2587	5032	1227		TAD	BLI
2588	5033	3237		DCA	BLO
2589	5034	6212		CIF 10	
2590	5035	4464		JMS 1	(200
2591	5036	0006		6	/CHAIN
2592	5037	0000	BLO,	0	
2593					
2594	5040	2054	EXA,	ISZ	LISTEN
2595	5041	6362		KIE	
2596	5042	5576		EXIT	/RETURN TO THE BACKGROUND PROGRAM
2597					
2598	5043	0000	NM0,	0	
2599	5044	0000	NM1,	0	
2600	5045	0000	NM2,	0	
2601	5046	2524	NM3,	2524	/UT EXTENSION
2602					
2603	5047	0000	GETCH,	0	
2604	5050	4771		JMS	RESYM
2605	5051	1370		TAD	(-203
2606	5052	7450		SNA	
2607	5053	5240		JMP	EXA
2608	5054	1367		TAD	(-340+203
2609	5055	7100		CLL	
2610	5056	1366		TAD	(100
2611	5057	7420		SNL	
2612	5060	5264		JMP	GET1
2613	5061	1365		TAD	(-40
2614	5062	0364		AND	(77

2615	5063	5647		JMP I	GETCH	
2616	5064	1363	GET1,	TAD	(240-215	
2617	5065	7650		SNA	CLA	
2618	5066	5303		JMP	RD2EX	
2619	5067	1362		TAD	("?	
2620	5070	4773		JMS	PRINT	
2621	5071	5216		JMP	RDN	
2622						
2623	5072	0000	RD2,	0		
2624	5073	3305		DCA	NM	
2625	5074	4247		JMS	GETCH	
2626	5075	7106		CLL	RTL	
2627	5076	7006		RTL		
2628	5077	7006		RTL		
2629	5100	3305		DCA	NM	
2630	5101	4247		JMS	GETCH	
2631	5102	7100		CLL		
2632	5103	1305	RD2EX,	TAD	NM	
2633	5104	5672		JMP I	RD2	
2634						
2635	5105	0000	NM,	0		
2636						
2637	5106	0000	RDNAME,	0		
2638	5107	3243		DCA	NM0	
2639	5110	3244		DCA	NM1	
2640	5111	3245		DCA	NM2	
2641	5112	4272		JMS	RD2	
2642	5113	3243		DCA	NM0	
2643	5114	7430		SZL		
2644	5115	5331		JMP	EONAME	
2645	5116	4272		JMS	RD2	
2646	5117	3244		DCA	NM1	
2647	5120	7430		SZL		
2648	5121	5331		JMP	EONAME	
2649	5122	4272		JMS	RD2	
2650	5123	3245		DCA	NM2	
2651	5124	7430		SZL		
2652	5125	5331		JMP	EONAME	
2653	5126	4272		JMS	RD2	
2654	5127	7620		SNL	CLA	
2655	5130	5326		JMP	, -2	/WAIT FOR CR
2656	5131	4775	EONAME,	JMS	CRL	
2657	5132	1361		TAD	(NM0	
2658	5133	5706		JMP I	RDNAME	
2659	5161	5043				
2660	5162	0277				
2661	5163	0023				
2662	5164	0077				
2663	5165	7740				
2664	5166	0100				
2665	5167	7643				
2666	5170	7575				
2667	5171	5276				
2668	5172	5263				
2669	5173	5200				

/ UTOR DD. 12/3/73

PAL8 9/17/74 PAGE 22-2

2670	5174	0300
2671	5175	5206
2672	5176	7700
2673	5177	4434
2674		

PAGE

2675	5200	0000	PRINT,	0	
2676	5201	6046		TLS	
2677	5202	6041		TSF	
2678	5203	5202		JMP	.-1
2679	5204	7200		CLA	
2680	5205	5600		JMP I	PRINT
2681					
2682	5206	0000	CRL,	0	
2683	5207	1377		TAD	(215
2684	5210	6201		CDF 00	
2685	5211	4200		JMS	PRINT
2686	5212	1376		TAD	(212
2687	5213	4200		JMS	PRINT
2688	5214	5606		JMP I	CRL
2689					
2690	5215	0000	PROK,	0	
2691	5216	3236		DCA	NUMB
2692	5217	1375		TAD	(-4
2693	5220	3237		DCA	PCTR
2694	5221	1236	PRO,	TAD	NUMB
2695	5222	7004		RAL	
2696	5223	7006		RTL	
2697	5224	3236		DCA	NUMB
2698	5225	1236		TAD	NUMB
2699	5226	7004		RAL	
2700	5227	0374		AND	(7
2701	5230	1373		TAD	(260
2702	5231	6201		CDF 00	
2703	5232	4200		JMS	PRINT
2704	5233	2237		ISZ	PCTR
2705	5234	5221		JMP	PRO
2706	5235	5615		JMP I	PROK
2707					
2708	5236	0000	NUMB,	0	
2709	5237	0000	PCTR,	0	
2710					
2711	5240	6361	ERROR,	KTID	
2712	5241	4206		JMS	CRL
2713	5242	1372		TAD	("E
2714	5243	6201		CDF 00	
2715	5244	4200		JMS	PRINT
2716	5245	1371		TAD	("R
2717	5246	6201		CDF 00	
2718	5247	4200		JMS	PRINT
2719	5250	1370		TAD	(240
2720	5251	6201		CDF 00	
2721	5252	4200		JMS	PRINT
2722	5253	1024		TAD	DATAPN
2723	5254	4215		JMS	PROK
2724	5255	6002		IOF	
2725	5256	3022		DCA	JSWMJ
2726	5257	6201		CDF 00	
2727	5260	1367		TAD	(RDCOMM
2728	5261	4573		MAKE JOB	
2729	5262	5576		EXIT	

```
2730
2731 5263 0000 ER, 0
2732 5264 7200 CLA
2733 5265 6201 CDF 0
2734 5266 3053 DCA CANCEL
2735 5267 6002 IOF
2736 5270 1263 TAD ER
2737 5271 3023 DCA DTPNMJ
2738 5272 3022 DCA JSWMJ
2739 5273 1366 TAD (ERROR
2740 5274 4573 MAKE JOB
2741 5275 5576 EXIT
2742
2743 5276 0000 RESYM, 0
2744 5277 6001 ION
2745 5300 3320 DCA ECHOFG
2746 5301 6214 RDF
2747 5302 1365 TAD (CIF CDF
2748 5303 3315 DCA RESEX
2749 5304 6031 KSF
2750 5305 5304 JMP .-1
2751 5306 6036 KRB
2752 5307 3321 DCA RECHAR
2753 5310 1321 TAD RECHAR
2754 5311 2320 ISZ ECHOFG
2755 5312 4200 JMS PRINT
2756 5313 7200 CLA
2757 5314 1321 TAD RECHAR
2758 5315 6203 RESEX, CDF CIF
2759 5316 6001 ION
2760 5317 5676 JMP I RESYM
2761 5320 0000 ECHOFG, 0
2762 5321 0000 RECHAR, 0
2763
2764 5322 0304 CDEC, FILENA CDECLT,SU
2765 5323 0503
2766 5324 1424
2767 5325 2325
2768 5326 2523 USER, FILENA USER,SU
2769 5327 0522
2770 5330 0000
2771 5331 2325
2772
2773 5365 6203
2774 5366 5240
2775 5367 5000
2776 5370 0240
2777 5371 0322
2778 5372 0305
2779 5373 0260
2780 5374 0007
2781 5375 7774
2782 5376 0212
2783 5377 0215
2784
```

PAGE

```

2785      /OWN VERSION OF PS/8 SYSTEM HANDLER
2786      PATCH=
2787          NOPUNC
2788          *7605
2789          ENPUNC
2790
2791      SHNDLR= 7607
2792      SOFSET= 7747
2793      SWC=    SHNDLR-7+150
2794      SCA=    SHNDLR-7+151
2795      7605    5606    MON,    JMP I    ,+1
2796      7606    5000          RDCOMM
2797      7607    0000    SHNDLR, 0
2798      7610    7346          CLA CLL CMA RTL
2799      7611    3330          DCA    SYSCNT
2800      7612    1607    STRY,   TAD I    SHNDLR
2801      7613    7004          RAL
2802      7614    7206          CLA RTL
2803      7615    1312          TAD    S6603
2804      7616    3273          DCA    SFUN
2805      7617    1607          TAD I    SHNDLR
2806      7620    0315          AND    S70
2807      7621    3306          DCA    SFIELD
2808      7622    1607          TAD I    SHNDLR
2809      7623    7004          RAL
2810      7624    0256          AND    S7600
2811      7625    7041          CIA
2812      7626    3350          DCA    SWC
2813      7627    7240          CLA CMA
2814      7630    2207          ISZ    SHNDLR
2815      7631    1607          TAD I    SHNDLR
2816      7632    3351          DCA    SCA
2817      7633    1273          TAD    SFUN
2818      7634    7012          RTR
2819      7635    7620          SNL CLA
2820      7636    5253          JMP    SGOED
2821      7637    1306          TAD    SFIELD
2822      7640    7640          SZA CLA
2823      7641    5253          JMP    SGOED
2824      7642    1314          TAD    S2201
2825      7643    1351          TAD    SCA
2826      7644    7630          SZL CLA
2827      7645    4713          JMS I    SERR
2828      7646    1350          TAD    SWC
2829      7647    7041          CIA
2830      7650    1351          TAD    SCA
2831      7651    7630          SZL CLA
2832      7652    4713          JMS I    SERR
2833      7653    2207    SGOED,   ISZ    SHNDLR
2834      7654    1306          TAD    SFIELD
2835      7655    6615          6615
2836      7656    7600    S7600,   7600
2837      7657    1607          TAD I    SHNDLR
2838      7660    1347          TAD    SOFSET
2839      7661    7012          RTR

```

2840	7662	7012	RTR	
2841	7663	0316	AND	S377
2842	7664	6643	6643	
2843	7665	1607	TAD I	SHNDLR
2844	7666	1347	TAD	SOFSET
2845	7667	7012	RTR	
2846	7670	7012	RTR	
2847	7671	7010	RAR	
2848	7672	0331	AND	S7400
2849	7673	6603	SFUN,	6603
2850	7674	6214	RDF	
2851	7675	1311	TAD	SCDIF
2852	7676	3306	DCA	SFIELD
2853	7677	6623	6623	
2854	7700	5277	JMP	.-1
2855	7701	2207	ISZ	SHNDLR
2856	7702	6621	6621	
2857	7703	7410	SKP	
2858	7704	5332	JMP	SERROR
2859	7705	2207	ISZ	SHNDLR
2860	7706	6203	SFIELD,	CIF CDF
2861	7707	6601	6601	
2862	7710	5607	JMP I	SHNDLR
2863	7711	6203	SCDIF,	CDF CIF
2864	7712	6603	S6603,	6603
2865	7713	5263	SERR,	ER
2866	7714	2201	S2201,	2201
2867	7715	0070	S70,	70
2868	7716	0377	S377,	377
2869	7717	0500	S500,	500
2870				
2871			BOOT=	7720
2872	7720	1256	BOOT,	TAD S7600
2873	7721	3350	DCA	SHNDLR-7+150
2874	7722	1312	TAD	S6603
2875	7723	3351	DCA	SHNDLR-7+151
2876	7724	6603	6603	
2877	7725	6622	6622	
2878	7726	5325	JMP	.-1
2879	7727	5725	JMP I	.-2
2880				
2881	7730	7775	SYSNT,	=3
2882	7731	7400	S7400,	7400
2883				
2884			SERROR,	
2885	7732	7330	CLA CLL CML RAR	
2886	7733	2330	ISZ	SYSNT
2887	7734	7610	SKP CLA	
2888	7735	5306	JMP	SFIELD
2889	7736	7346	CLA CLL CMA RTL	
2890	7737	1207	TAD	SHNDLR
2891	7740	3207	DCA	SHNDLR
2892	7741	5212	JMP	STRY
2893			SPEND=	.,+PATCH-MON

2894		
2895	0064	0200
2896	0065	5000
2897	0066	6201
2898	0067	6203
2899	0070	0070
2900	0071	6202
2901	0072	16/2
2902	0073	1000
2903	0074	0452
2904	0075	3236
2905	0076	7773
2906	0077	2400
2907	0100	2550
2908	0101	7400
2909	0102	4045
2910	0103	1200
2911	0104	3141
2912	0105	7740
2913	0106	7770
2914	0107	0757
2915	0110	0361
2916	0111	0205
2917	0112	7774
2918	0113	0211
2919	0114	0644
2920	0115	7771
2921	0116	0600
2922	0117	0400
2923	0120	0201
2924	0121	0177
2925	0122	0347
2926	0123	0745
2927	0124	0440
2928	0125	0217
2929	0126	0203
2930	0127	0014
2931	0130	7764
2932	0131	0270
2933	0132	3200
2934	0133	0446
2935	0134	0012
2936	0135	7563
2937	0136	0323
2938	0137	0020
2939	0140	0417
2940	0141	0307
2941	0142	0005
2942	0143	3100
2943	0144	0275
2944	0145	0302
2945	0146	0254
2946	0147	2426
2947	0150	0300
2948	0151	3035

2949	0152	0243
2950	0153	0224
2951	0154	0263
2952	0155	0236
2953	0156	0010
2954	0157	7500
2955	0160	0427
2956	0161	0231
2957	0162	0377
2958	0163	0007
2959	0164	2353
2960	0165	3120
2961	0166	2146
2962	0167	2121
2963	0170	4200
2964	0171	3450
2965	0172	3476
2966	0173	3701
2967	0174	4000
2968	0175	2236
2969	0176	3600
2970	0177	4055

FIELD 3

/DATA FIELD 3

2974		
2975	0000	0000

*HDADSP
ZBLOCK 200

2976		
2977		

/POINTERS TO FREE SPACE

2978		
2979	0020	0000
2980	0021	0000
2981	0022	0000
2982	0023	0000
2983	0024	0000
2984	0025	0000
2985	0026	0000
2986	0027	2000

*HDFSP

0

0

0

0

0

0

0

2000

2987		
2988		

*HDMASK

2989	0000	0004
2990	0001	0010
2991	0002	0020
2992	0003	0040
2993	0004	0100
2994	0005	0200
2995	0006	0400
2996	0007	1000

0004

0010

0020

0040

0100

0200

0400

1000

2997		
2998		

/LIST OF FREE AREAS

2999		
3000		
3001	2000	3000
3002	2001	0027
3003	2002	0007

SIZE=7

*2000

3000

HDFSP+7

7

3004			*3000	
3005	3000	4000	4000	
3006	3001	2000	2000	
3007	3002	0007	7	
3008			*4000	
3009	4000	5000	5000	
3010	4001	3000	3000	
3011	4002	0007	7	
3012			*5000	
3013	5000	0000	0000	
3014	5001	4000	4000	
3015	5002	0007	7	
3016			/UP FROM 6000 CORE IS USED BY THE PLOTTING ROUTINES	
3017				
3018				
3019			/ENTRY POINTS JOB CHAIN[0:MAXPR]	
3020			*TAILS	
3021	0030	0000	0	
3022	0031	0000	0	
3023	0032	0000	0	
3024	0033	0200	JOECH	
3025				
3026			*JOBCH	
3027	0200	0000	0	/POINTS TO FIRST WAITING JOB
3028				
3029			/SPECIAL CHARACTERS	
3030			*HDTRB	
3031	0760	0137	0137	/UL
3032	0761	0174	0174	/BAR
3033	0762	0215	215	/CR
3034				
3035			*HDMS	
3036	0763	1016	BEL-1	
3037	0764	1106	ZOM-1	
3038	0765	1076	ROM-1	
3039	0766	1023	OK-1	
3040	0767	1035	HOCR-1	
3041	0770	1064	ZSCLO-1	
3042	0771	1051	RSCLO-1	
3043	0772	1136	GUEST-1	
3044	0773	1126	NOPT-1	
3045	0774	1116	ACX8-1	
3046	0775	1144	DUPL-1	
3047	0776	1162	BIPL-1	
3048		1200	SINGL-1	
3049	1000	1021	CRLF-1	
3050	1001	7775	MSNR, 3	
3051	1002	7770	10	
3052	1003	7770	10	
3053	1004	7766	12	
3054	1005	7764	14	
3055	1006	7766	12	
3056	1007	7765	13	
3057	1010	7772	6	
3058	1011	7770	10	

3059	1012	7770		-10
3060	1013	7762		-16
3061	1014	7762		-16
3062	1015	7762		-16
3063	1016	7776		-2
3064				
3065	1017	0207	BEL,	207
3066	1020	0207		207
3067	1021	0207		207
3068	1022	0215	CRLF,	215
3069	1023	0012		012
3070	1024	0024	OK,	024
3071	1025	0223		223
3072	1026	0215		215
3073	1027	0012		012
3074	1030	0216		216
3075	1031	0300		300
3076	1032	0317		317
3077	1033	0113		113
3078	1034	0215		215
3079	1035	0012		012
3080	1036	0024	HOOR,	024
3081	1037	0223		223
3082	1040	0215		215
3083	1041	0012		012
3084	1042	0017		017
3085	1043	0300		300
3086	1044	0110		110
3087	1045	0317		317
3088	1046	0317		317
3089	1047	0322		322
3090	1050	0215		215
3091	1051	0012		012
3092	1052	0215	RSOLO,	215
3093	1053	0012		012
3094	1054	0017		017
3095	1055	0300		300
3096	1056	0123		123
3097	1057	0317		317
3098	1060	0314		314
3099	1061	0317		317
3100	1062	0215		215
3101	1063	0012		012
3102	1064	0216		216
3103	1065	0215	ZSOLO,	215
3104	1066	0012		012
3105	1067	0216		216
3106	1070	0300		300
3107	1071	0123		123
3108	1072	0317		317
3109	1073	0314		314
3110	1074	0317		317
3111	1075	0215		215
3112	1076	0012		012
3113	1077	0215	ROM,	215

3114	1100	0012		012
3115	1101	0017		017
3116	1102	0300		300
3117	1103	0317		317
3118	1104	0115		115
3119	1105	0215		215
3120	1106	0012		012
3121	1107	0215	ZOM,	215
3122	1110	0012		012
3123	1111	0216		216
3124	1112	0300		300
3125	1113	0317		317
3126	1114	0115		115
3127	1115	0215		215
3128	1116	0012		012
3129	1117	0024	ACX8,	024
3130	1120	0223		223
3131	1121	0215		215
3132	1122	0012		012
3133	1123	0017		017
3134	1124	0300		300
3135	1125	0330		330
3136	1126	0270		270
3137	1127	0215	NOPT,	215
3138	1130	0012		12
3139	1131	0300		300
3140	1132	0116		116
3141	1133	0317		317
3142	1134	0120		120
3143	1135	0215		215
3144	1136	0012		012
3145	1137	0215	QUEST,	215
3146	1140	0012		012
3147	1141	0300		300
3148	1142	0077		077
3149	1143	0215		215
3150	1144	0012		012
3151	1145	0024	DUPL,	024
3152	1146	0223		223
3153	1147	0215		215
3154	1150	0012		012
3155	1151	0017		017
3156	1152	0300		300
3157	1153	0104		104
3158	1154	0125		125
3159	1155	0120		120
3160	1156	0314		314
3161	1157	0305		305
3162	1160	0330		330
3163	1161	0215		215
3164	1162	0012		012
3165	1163	0024	BIPL,	024
3166	1164	0223		223
3167	1165	0215		215
3168	1166	0012		012

3169	1167	0017	017
3170	1170	0300	300
3171	1171	0102	102
3172	1172	0311	311
3173	1173	0120	120
3174	1174	0314	314
3175	1175	0305	305
3176	1176	0330	330
3177	1177	0215	215
3178	1200	0012	012
3179	1201	0024	SINGL, 024
3180	1202	0223	223
3181	1203	0215	215
3182	1204	0012	012
3183	1205	0017	017
3184	1206	0300	300
3185	1207	0123	123
3186	1210	0311	311
3187	1211	0116	116
3188	1212	0107	107
3189	1213	0314	314
3190	1214	0305	305
3191	1215	0215	215
3192	1216	0012	012
3193			
3194	1217	0000	WORKSP, 0

			#HDCMTB
3195			
3196	0300	7777	INTREP, -1
3197	0301	7777	-1
3198	0302	7777	-1
3199	0303	7777	-1
3200	0304	7777	-1
3201	0305	7777	-1
3202	0306	7777	-1
3203	0307	7777	-1
3204	0310	7777	-1
3205	0311	0166	166
3206	0312	0567	567
3207	0313	7777	-1
3208	0314	7777	-1
3209	0315	7777	-1
3210	0316	7777	-1
3211	0317	7777	-1
3212	0320	7777	-1
3213	0321	0211	211
3214	0322	0212	212
3215	0323	0213	213
3216	0324	0214	214
3217	0325	7777	-1
3218	0326	7777	-1
3219	0327	7777	-1
3220	0330	7777	-1
3221	0331	7777	-1
3222	0332	7777	-1
3223	0333	7777	-1
3224	0334	7777	-1
3225	0335	7777	-1
3226	0336	7777	-1
3227	0337	7777	-1
3228	0340	0135	135
3229	0341	0201	201
3230	0342	0171	171
3231	0343	0575	575
3232	0344	0205	205
3233	0345	0204	204
3234	0346	0173	173
3235	0347	0170	170
3236	0350	0142	142
3237	0351	0143	143
3238	0352	0102	102
3239	0353	0100	100
3240	0354	0127	127
3241	0355	0101	101
3242	0356	0130	130
3243	0357	0103	103
3244	0360	0000	0
3245	0361	0001	1
3246	0362	0002	2
3247	0363	0003	3
3248	0364	0004	4
3249	0365	0005	5

3250	0366	0006	6
3251	0367	0007	7
3252	0370	0010	10
3253	0371	0011	11
3254	0372	0132	132
3255	0373	0133	133
3256	0374	0110	110
3257	0375	0106	106
3258	0376	0112	112
3259	0377	0172	172
3260	0400	0600	600
3261	0401	0045	45
3262	0402	0046	46
3263	0403	0047	47
3264	0404	0050	50
3265	0405	0051	51
3266	0406	0052	52
3267	0407	0053	53
3268	0410	0054	54
3269	0411	0055	55
3270	0412	0056	56
3271	0413	0057	57
3272	0414	0060	60
3273	0415	0061	61
3274	0416	0062	62
3275	0417	0063	63
3276	0420	0064	64
3277	0421	0065	65
3278	0422	0066	66
3279	0423	0067	67
3280	0424	0070	70
3281	0425	0071	71
3282	0426	0072	72
3283	0427	0073	73
3284	0430	0074	74
3285	0431	0075	75
3286	0432	0076	76
3287	0433	0144	144
3288	0434	0131	131
3289	0435	0145	145
3290	0436	0105	105
3291	0437	0176	176
3292	0440	0116	116
3293	0441	0012	12
3294	0442	0013	13
3295	0443	0014	14
3296	0444	0015	15
3297	0445	0016	16
3298	0446	0017	17
3299	0447	0020	20
3300	0450	0021	21
3301	0451	0022	22
3302	0452	0023	23
3303	0453	0024	24
3304	0454	0025	25

3305	0455	0026	26
3306	0456	0027	27
3307	0457	0030	30
3308	0460	0031	31
3309	0461	0032	32
3310	0462	0033	33
3311	0463	0034	34
3312	0464	0035	35
3313	0465	0036	36
3314	0466	0037	37
3315	0467	0040	40
3316	0470	0041	41
3317	0471	0042	42
3318	0472	0043	43
3319	0473	0120	120
3320	0474	0177	177
3321	0475	0117	117
3322	0476	0114	114
3323	0477	7777	=1
3324			
3325			*HDCHTB+200
3326	0500	0060	ISOTAB, 60
3327	0501	0261	261
3328	0502	0262	262
3329	0503	0063	63
3330	0504	0264	264
3331	0505	0065	65
3332	0506	0066	66
3333	0507	0267	267
3334	0510	0270	270
3335	0511	0071	71
3336	0512	0341	341
3337	0513	0342	342
3338	0514	0143	143
3339	0515	0344	344
3340	0516	0145	145
3341	0517	0146	146
3342	0520	0347	347
3343	0521	0350	350
3344	0522	0151	151
3345	0523	0152	152
3346	0524	0353	353
3347	0525	0154	154
3348	0526	0355	355
3349	0527	0356	356
3350	0530	0157	157
3351	0531	0360	360
3352	0532	0161	161
3353	0533	0162	162
3354	0534	0363	363
3355	0535	0164	164
3356	0536	0365	365
3357	0537	0366	366
3358	0540	0167	167
3359	0541	0170	170

3360	0542	0371	371
3361	0543	0372	372
3362	0544	0077	77
3363	0545	0101	101
3364	0546	0102	102
3365	0547	0303	303
3366	0550	0104	104
3367	0551	0305	305
3368	0552	0306	306
3369	0553	0107	107
3370	0554	0110	110
3371	0555	0311	311
3372	0556	0312	312
3373	0557	0113	113
3374	0560	0314	314
3375	0561	0115	115
3376	0562	0116	116
3377	0563	0317	317
3378	0564	0120	120
3379	0565	0321	321
3380	0566	0322	322
3381	0567	0123	123
3382	0570	0324	324
3383	0571	0125	125
3384	0572	0126	126
3385	0573	0327	327
3386	0574	0330	330
3387	0575	0131	131
3388	0576	0132	132
3389	0577	0077	77
3390	0600	0053	053
3391	0601	0055	055
3392	0602	0252	252
3393	0603	0257	257
3394	0604	0472	472
3395	0605	0336	336
3396	0606	0275	275
3397	0607	1275	1275
3398	0610	0074	074
3399	0611	0474	474
3400	0612	0276	276
3401	0613	0676	676
3402	0614	0176	176
3403	0615	0675	675
3404	0616	0140	140
3405	0617	0175	175
3406	0620	0173	173
3407	0621	0022	22
3408	0622	0024	24
3409	0623	0021	21
3410	0624	0223	223
3411	0625	0017	17
3412	0626	0216	216
3413	0627	0254	254
3414	0630	0056	56

3415	0631	0134	134
3416	0632	0072	072
3417	0633	0273	273
3418	0634	0077	77
3419	0635	0240	240
3420	0636	0077	77
3421	0637	0077	77
3422	0640	0077	77
3423	0641	0077	77
3424	0642	0050	050
3425	0643	0251	251
3426	0644	0333	333
3427	0645	0335	335
3428	0646	1074	1074
3429	0647	1276	1276
3430	0650	0077	77
3431	0651	0077	77
3432	0652	0077	77
3433	0653	0077	77
3434	0654	0077	77
3435	0655	0077	77
3436	0656	0077	77
3437	0657	0077	77
3438	0660	0077	77
3439	0661	0077	77
3440	0662	0077	77
3441	0663	0077	77
3442	0664	0077	77
3443	0665	0077	77
3444	0666	0011	011
3445	0667	1412	1412
3446	0670	0047	47
3447	0671	0042	42
3448	0672	0077	77
3449	0673	0246	246
3450	0674	0077	77
3451	0675	0243	243
3452	0676	0137	137
3453	0677	0374	374
3454	0700	0300	300
3455	0701	0041	41
3456	0702	0077	77
3457	0703	0077	77
3458	0704	0245	245
3459	0705	0044	44
3460	0706	0215	215
3461	0707	0012	12
3462	0710	0017	17
3463	0711	0216	216
3464	0712	0021	21
3465	0713	0223	223
3466	0714	0022	22
3467	0715	0024	24
3468	0716	0207	207
3469			

ENDCTB= .

/ UTOR DD. 12/3/73

PAL8 9/17/74 PAGE 26-5

3470

\$\$

3471

A	0046	ACBUF	3035	ACURL	0300	ACX8	1117	AGMRGE	3425
AGPR	3011	AMA	0201	AMAL	0205	ATTPX	1761	AT1	2277
AUTIN1	0010	AUTIN2	0011	AUTIN3	0012	AUTIN4	0013	AUTIN5	0014
AUTIN6	0015	AUTIN7	0016	AUTIN8	0017	AVAIL	0020	BA	0006
BCOUNT	2471	BEL	1017	BF	2302	BFLST	2311	BIPL	1163
BLCK	0216	BLI	5027	BLO	5037	BOOT	7720	BTWEEN	0427
BUFPTR	1770	BOOT	7720	B1	0037	CANCEL	0053	CDEC	5322
CHANGE	0417	CHAR	0047	CHBU1	1541	CHMAX	0216	CHOXIT	2444
CK	0007	CLBUF	2400	CLEAR	4153	CLEARU	4155	CNTCHR	2061
COLOUR	3062	COMSIN	0727	CONOUT	1422	CR	0215	CRL	5206
CRLF	1022	CRNODE	3706	DATAPN	0024	DECIS	3600	DEC1	3601
DELET	3344	DELETE	0243	DR	2765	DTA	4544	DTPNMJ	0023
DUB	2334	DUMMY	0377	DUPL	1145	DUPLAC	1154	DUPLX1	1147
DUPLX2	3274	ECHO	4564	ECHOA	2353	ECHOFG	5320	ENDCTB	0717
EONAME	5131	EPTR	2246	EQUAL	0440	ER	5263	ERROR	5240
EXA	5040	EXCIR	2141	EXGM	2276	EXGMM2	2275	EXIT	5576
EXITKI	2236	EXITSI	5575	EXKI	4045	EXUS	4705	FILLR	2550
FILLRB	2563	FIN	3553	FIRST	3046	FLD	0030	FRBUF	3400
FREEBF	0757	FREEBU	4571	FREEND	3141	GET	3527	GETBF	3476
GETBIT	3303	GETBUF	4572	GETCH	5047	GETMCH	2250	GETND	3120
GETNOD	4565	GETTL	3744	GET1	5064	GTB1	0217	GTLCH	0270
GTPRT	0243	GTPXP	0361	GTSTC	0231	GTXPMR	0302	HDADSP	0000
HDAVAI	1400	HDCHTB	0300	HDFSP	0020	HDMASK	0000	HDMS	0763
HDTRB	0760	HDTTI	2610	HITRS	1517	HOOR	1036	HSP	4473
HSR	4532	INICNT	2764	INIMP	2342	INIT	0200	INITIR	4567
INITJB	4566	INJOB	2146	INNOD	3723	INS	3467	INSERT	3317
INSPEC	4055	INTRAP	2121	INTREP	0300	INUSER	4600	ISOD1	1255
ISOTAB	0500	JOB	0000	JOBCH	0200	JSW	0021	JSWMJ	0022
JUMPER	0615	KB	4711	KEY	0275	KILL	1157	LF	0012
LISTEN	0054	LOC	3564	LOCBDY	3565	LOCK	0051	LP	1232
LPCNT	0044	LPCRJS	2614	LPSP	2650	LSTCHR	0042	LTAIL	3755
LTRS	1533	MAKE	4573	MAKEJB	3701	MALA	0211	MAXJOB	0100
MAXPR	0003	MAXPX8	0220	MAXSZ	0007	MCH	3076	MCHAR	0056
MEME	1067	MILP	0057	MJRCNT	2352	MNRCNT	2351	MODKB	4725
MON	7605	MPNT	3567	MSKPR	4003	MSKSZ	0007	MSNR	1001
MSPT	0763	MSAGE	0307	MX8	0003	NEQUAL	0446	NEWCON	1447
NEXTJB	3624	NILFIL	2460	NM	5105	NM0	5043	NM1	5044
NM2	5045	NM3	5046	NOCF	1135	NOLFT	1101	NOMES	1074
NOPT	1127	NOROOM	3135	NOT	0176	NOTLST	2322	NOTX8	1320
NOX	1040	NOX8	0667	NUMB	5236	NXTPXB	0745	NXTTI	2530
OFFTRS	1477	OK	1024	OUTFBF	3474	OUTGB	3562	OUTPUT	2210
OUT7S	3002	PART	0045	PART2	0340	PAR1	0025	PAR2	0026
PAR3	0027	PATCH	5400	PEUSY	1072	PCP	3700	PCTR	5237
PLEX	4441	PLINS	4434	PLOTR	4430	PLCUT	4400	PLTEM	4444
POETS	3754	PRINT	5200	PROK	5215	PRTNER	0041	PRO	5221
PUEX	4475	PUNCH	4500	PXBUCN	0040	PXCHAR	1742	PXDONE	1655
PXECHO	2426	PXECHX	2446	PXIBC	1762	PXLOOP	1644	PXNUMB	0035
PXOBC	1763	PXOVER	1672	PXP	0034	PXREAD	1766	PXREPT	1760
PXS	1621	PXSEND	1627	PXWAIT	2111	PXWAPT	1764	PX1	1733
PX2	1600	P2	2244	P3	2245	P7S	3031	QUEST	1137
R	0031	RCHAR	4531	RDCOMM	5000	RDLP	4515	RDN	5016
RDNAME	5106	RDPTR	4727	RD2	5072	RD2EX	5103	REACTI	5574
READ	4510	RECHAR	5321	RED	0017	REEX	4526	REQUA	3100
REQUXP	1346	RESEX	5315	RESYM	5276	RF08	4453	RF08ER	4466

ROCHAR	3200	ROM	1077	RR	0030	RSOLO	1052	RTIM	4530
RUNOUT	1431	SAJOB	3756	SAVDIR	1765	SAVE	4570	SAVLN	0060
SCA	7751	SCDIF	7711	SCHARX	2060	SCNT	0004	SCNTR	4730
SCOPE	4535	SEND	3236	SERR	7713	SERROR	7732	SETB1	1124
SETIF	1747	SETST	0236	SETTI	2520	SFIELD	7706	SFUN	7673
SGOED	7653	SHNDLR	7607	SINAC	0646	SINCR	0741	SINGL	1201
SINGLE	3300	SISOUT	1440	SIZE	3566	SNEX	0662	SOFSET	7747
SOLAC	0636	SOLO	0644	SOUTAC	0703	SP	0047	SPEND	5535
SPLIT	3532	SPN	0013	SPTR	0047	SP1	0043	SP2	2517
SRCHBF	3513	STACK	4200	STACKP	0012	STAMP	3262	START	2600
STATE	0040	STB1	0224	STLCH	0263	STOR	1243	STPRT	0254
STPXP	0347	STRY	7612	STXPMR	0275	ST1	2704	ST2	2723
SV	0004	SVAC	0061	SVCHAR	0511	SVCHR	1540	SVF	3077
SVLINK	0062	SVPT	2247	SVPXR	1553	SV1	0005	SV2	0006
SV3	0007	SWC	7750	SWITCH	0323	SYSCNT	7730	S2201	7714
S377	7716	S500	7717	S6603	7712	S70	7715	S7400	7731
S7600	7656	TAILS	0030	TEBUSY	1467	TDSK	4242	TERNR	1767
TJUMP	0614	TK	0474	TKOVER	0452	TKOVR2	0400	TKXPBU	1200
TODISK	4231	TRANS	0302	TRNST1	0010	TRNST2	0020	TRTURN	3200
TRXIT	1122	TRXIT2	1144	TSIN	0715	TSNUF	0600	TSNUF1	0601
TSOUT	1307	TSTBIT	2450	TSTCAN	4217	TSTMNR	1273	TSXIT	0722
TSXIT1	1335	TTDONE	2225	TTOGO	2215	TTRUM	1000	TTY	4444
TURN	1337	TURNON	1025	TURXIT	3233	T1S	2200	UNSTAC	4000
UP	1272	USER	5326	US0	4607	US1	4616	US2	4640
US3	4661	WA	0005	WARN	0055	WORKSP	1217	XACURL	1400
XLIN	0052	XPBPTR	2056	XPDONE	2034	XPLOOP	2023	XLPCN	2057
XPMR	0036	XPNUMB	0033	XPP	0032	XPSKIP	2073	XPSTOR	2062
XPTNR	2055	XPWAIT	2100	XP2	2000	X7FLAG	0050	X7P1	3034
X7S	3000	X8	0002	ZERO	0000	ZOM	1107	ZSOLO	1065

CBUX2	1023	1095								
CBUX7	1580									
CCF	2253									
CDEC	2470	2764#								
CDECLT	2764									
CHANGE	247	297#	304	432	437	447	470	552	560	567
	732	741	743	852	858	864	1719	1721		
CHAR	79#	151	158	162	170	174	192	198	206	210
	218	260	264	308	311	318	377	416	474	543
	581	596	601	619	649	651	657	698	711	725
	769	776	807	813	834	838	840	855	861	1304
	1306	1381	1390	1391	1392	1566	1618	1716	1735	
CHBU1	868#	877	902							
CHMAX	123#	675	678							
CHOXIT	1332#	1340	1358	1362	1375					
CINT	2256									
CK	1784#	1819	1856	1871	1895	1904	1907	1944	1946	1951
	1955	1960	1966	1967	1969	1972				
CLBUF	1243	1295#								
CLEAR	2246	2253#								
CLEARU	2244	2256#								
CNTCHR	1039	1059	1073#	1079						
COLOUR	1625	1635#								
COMSIN	462#									
CONOUT	780#	799								
CR	126#	453	506	509						
CRF	2439									
CRL	2575	2656	2682#	2688	2712					
CRLF	3049	3068#								
CRNODE	2089	2093#	2105							
DATAPN	56#	955	961	1137	1145	1146	1295	1558	2063	2155
	2287	2314	2722							
DECIS	98	99	388	397	2012#					
DEC1	1193	2013#								
DELET	1824#	1835	1839	1899	1959					
DELETE	128#	738	1711							
DR	1464	1466	1468	1516	1563#					
DTA	2229	2442#	2446							
DTPNMJ	53#	338	339	1052	1053	1242	1592	1593	1653	1654
	1743	1747	2101	2308	2737					
DUB	1234	1255	1256	1258#						
DUMMY	124#	1346	1632							
DUPL	3046	3151#								
DUPLAC	394	623#								
DUPLX1	385	464	617#	625						
DUPLX2	386	395	1771#							
ECHO	110	231	406	451	504	752	1275#	1315	1733	
ECHOA	110	1276#	1289							
ECHOFG	2745	2754	2761#							
ENDCTB	3469#									
EONAME	2644	2648	2652	2656#						
EPTR	1197#	1281	1283	1284	1286					
EQUAL	316#	320	321	324	521	546	554	578	695	735
ER	2073	2250	2258	2302	2330	2390	2542	2586	2731#	2736
	2865									
ERROR	2324	2397	2408	2711#	2739					

EXA	2594#	2607								
EXCIR	1123	1127#								
EXGM	1216	1224#								
EXGMM2	1223#	1228	1245	1263						
EXIT	99	371	407	425	469	508	537	565	574	594
	621	705	730	736	740	746	757	786	898	949
	992	1067	1300	1307	1310	1316	1605	1774	2011#	2318
	2348	2406	2560	2596	2729	2741				
EXITKI	100	1188#	2399	2409						
EXITSI	100	1187#	2353	2373	2385	2432	2440	2449	2540	
EXKI	1126	1191	1583	2181#	2201	2251	2254			
EXUS	2465	2472	2490	2507	2525#					
FILLR	1221	1404#	1414	1418						
FILLRB	1238	1252	1416#	1419	1631	1644				
FIN	1964	1979#								
FIRST	1226	1258	1622#	1633	1646					
FLD	1/83#	1860	1942							
FRBUF	105	1850#	1853	1854	1912					
FREEBF	490#	499	818	827	1614					
FREEBU	105	496	873	944	1849#	2509				
FREEND	962	1147	1687#	1700	2066					
GET	1950	1957#								
GETBF	104	1932#	1984	1987						
GETBIT	1786#	1799	1866	1885	1980					
GETBUF	104	481	977	1489	1526	1533	1683	1931#	2479	
GETCH	2603#	2615	2625	2630						
GETMCH	1171	1200#	1224							
GETND	109	1669#	1681	1685						
GETNOD	109	337	1051	1591	1652	1668#	1742	2095		
GETTL	2118	2124#								
GETI	2612	2616#								
GTB1	155#	159	512	576	737	1710				
GTLCH	203#	207	473	515	580	701	733	1308	1617	
GTPRT	179#	186	187	409	440	462	722	774	849	1299
	1708									
GTPXP	269#	276	817	1325	1613	1763				
GTSTC	167#	171	368	553	656	693	766	832	853	
GTXPMP	215#	219	255	415	724	1302	1714			
HDA DSP	6#	7	2974							
HDAVAI	13#	46								
HDCHTB	11#	3195	3325							
HDFSP	8#	1906	1945	1971	2978	3002				
HDMASK	7#	2988								
HDMS	10#	3035								
HDTRB	12#	1628	3030							
HDTTI	1160	1384	1569#							
HITRS	843	849#	860	865						
HOOR	3040	3080#								
HSP	2241	2392#								
HSR	2238	2430#								
INICNT	1498	1500	1505	1513	1520	1529	1562#			
INIMP	1185	1265#	1271	1542						
INIT	136#	2522								
INITIR	107	892	1025	1110#	1587	2257	2361	2389	2393	2541
INITJB	108	367	645	1133#						

INNOD	2090	2107#	2131								
INS	1877	1889	1897	1906#							
INSERT	1801#	1821	1910	1976							
INSPEC	24	2190#									
INTRAP	107	1111#	1124	1131							
INTREP	1320	3196#									
INUSER	1550	2456#	2528								
IS	2324										
ISOD1	693#	794									
ISOTAB	678	3326#									
JOB	103	289	347	1064	1125	1244	1601	1664	1754	2086#	
	2311	2728	2740								
JOBCH	39	120#	2017	2036	2048	3024	3026				
JSW	47#	285	1118	1129	1656	1744	2059	2069	2074	2157	
	2289	2295	2568								
JSWMJ	52#	287	1062	1114	1115	1128	1155	1240	1599	1658	
	1746	2097	2110	2202	2306	2725	2738				
JUMPER	376	380#									
KB	2223	2530#	2534								
KEY	222	224	228	256#							
KIE	2458	2527	2595								
KILL	518	627#									
KTID	2561	2711									
LF	127#	459	509	702	734	1734					
LISTEN	18	84#	2526	2538	2562	2594					
LOC	1852	1861	1865	1878	1881	1903	1909	1957	1958	1974	
	1979	1985	1989#								
LOCBDY	1883	1884	1891	1898	1902	1990#					
LOOK	81#	1189	2016	2108							
LP	673#	684									
LPCNT	76#	1344	1351	1356	1365	1373	1383	1398	1452		
LPCRJS	1447#	1453									
LPSP	1478#	1481									
LSTCHR	73#	199	204								
LTAIL	2039	2042	2049	2115	2116	2119	2130	2134#			
LTRS	857	861#									
MAKE	102	289	347	1064	1125	1244	1601	1664	1754	2085#	
	2311	2728	2740								
MAKEJB	102	2087#	2091								
MALA	148#	153	164	176	194	200	212	262	266	700	
MAXJOB	76	118#									
MAXPR	122#	2046									
MAXPXB	121#	970									
MAXSZ	115#	1936	1952								
MCH	1648#										
MCHAR	90#	1168	1172	1175	1178	1270	1407	1411	2307	2309	
MEME	550	563#									
MILP	91#	1156	1165	1457	1460	1538	1540				
MJRCNT	1183	1267	1273#								
MNRCNT	1166	1179	1272#								
MODKB	2523	2542#									
MON	2/95#	2893									
MPNT	1873	1874	1882	1900	1947	1948	1970	1975	1992#		
MSKPR	116#	1116	1119	2022	2027	2113					

MSPT	223#	3036#								
MSSAGE	221#	232	424	436	446	534	557	564	571	814
MX8	27#	184	237	1382	1439	1458	1519	1588		
NEQUAL	323#	325	326	460	514	517	654	703	739	767
	1309	1712								
NEWCON	781	801	804#	819						
NEXTJB	2032#									
NILFIL	1345#	1352	1353							
NM	2624	2629	2632	2635#						
NM0	2598#	2638	2642	2657						
NM1	2599#	2639	2646							
NM2	2600#	2640	2650							
NM3	2601#									
NOCF	589	606#								
NOLFT	511	576#								
NOMES	391	542	570#	579	612	629	632			
NOPT	3044	3137#								
NOROOM	1676	1682#								
NOT	129#	823								
NOTLST	1236	1247#								
NOTX8	723	732#								
NOX	522	539#								
NOX8	410	427#								
NUMB	2691	2694	2697	2698	2708#					
NXTPXB	478#	487	529	782	821	1615	1756			
NXTTI	1387#	1399								
OFFTRS	831#	836	847	1298						
OK	3039	3070#								
OKP	923	936	1608							
OKX	1028	1066	1546	1580						
OUTFBF	1859	1911#								
OUTGB	1941	1954	1986#							
OUTPUT	1162#	1429								
OUT7S	1581#	1609								
PART	77#	182	183	190	191	300	303	427	465	558
	582	591	597	599	611	800	815			
PART2	239	247#								
PAR1	57#	144	149	236	292	298	301	429	467	545
	627	1140	1278	1296	1659	1749				
PAR2	58#	226	259	272	283	372	452	457	484	494
	505	583	613	617	646	668	670	728	750	751
	762	763	793	796	1142	1282	1313	1328	1329	1341
	1345	1347	1355	1370	1371	1724	1726	1727	1740	1741
	1751	1761	1764	1766						
PAR3	59#	230	263	275	293	486	491	647	665	687
	692	697	790	1144	1285	1314	1330	1348	1372	1732
	1767									
PATCH	2513	2546	2786#	2893						
PBUSY	525	555	567#							
PCP	2023	2029	2037	2045	2083#					
PCTR	2693	2704	2709#							
PLCF	2350									
PLEX	2332	2347	2362#							
PLINS	2328	2339	2344	2351	2355#	2357	2564			
PLOTR	2235	2350#								
PLOUT	2325#	2331								

PLSF	2233
PLTEM	2327 2333 2340 2366#
POETS	2124 2125 2128 2133#
PRINT	2577 2620 2675# 2680 2685 2687 2703 2715 2718 2721 2155
PROK	2690# 2706 2723
PRTNER	72# 180 193
PR0	2694# 2705
PUEX	2394# 2405
PUNCH	2395 2399#
PXBUCN	119# 905 984 1004 1006
PXCHAR	927 929 931 994# 998
PXDONE	935#
PXECHO	413 532 620 729 785 824 826 1318# 1335 1619
PXECHX	1323 1334#
PXIBC	899 906 1005 1012#
PXLOOP	926# 934
PXNUMB	68# 265 273
PXOBC	973 985 1007 1013#
PXOVER	953# 1600 1663 1753
PXP	67# 261 270
PXREAD	895 948 966 1016#
PXREPT	871 901 903 907 908 910 911 1001 1002 1010#
PXS	900 907#
PXSEND	913# 968
PXWAIT	926 1101# 1108
PXWAPT	980 981 982 987 988 990 991 1003 1014#
PX1	974 986#
PX2	890# 2211
P2	1195# 1206 1214 1220 1227 1230 1231 1237 1251 1259 1260 1262
P3	1196# 1209 1210 1235 1247 1253
P7S	1607# 2217
QUEST	3043 3145#
R	64# 1269 1408 1515 1517
RCHAR	2411 2419 2428# 2431
RDCOMM	2520 2521 2543 2560# 2727 2796
RDLP	2415# 2423
RDN	2575# 2621
RDNAME	2578 2637# 2658
RDPTR	2494 2497 2545#
RD2	2623# 2633 2641 2645 2649 2653
RD2EX	2618 2632#
REACTION	101 2019 2031 2142#
READ	2409# 2425
RECHAR	2752 2753 2757 2762#
RED	131# 1642
REEX	2418 2421 2424#
REQUA	420 756 1301 1651# 1666
REQUXP	662 691 756# 802 828
RESEX	2748 2758#
RESYM	2604 2743# 2760
RF08	2220 2375# 2379
RF08ER	2382 2386#
RIS	2247 2368 2435 2531
ROCHAR	2317 2319#

[illegible]

STACK	106	2277#	2291							
STACKP	133#	2024	2146	2147	2148	2150	2152	2154	2156	2159
	2160	2282	2284	2286	2288	2290				
STAMP	1760#	1768	1769	1771	1776					
START	138	1431#	1435	1569						
STATE	71#	168	175	1636						
STB1	161#	165	243	404	475	603				
STLCH	197#	201	241	402	458	593	618	745		
STOR	677	682#								
STPRT	189#	195	414	528	544	559	816			
STPXP	258#	267	530	783	822	1333	1616	1773		
STRY	2800#	2892								
STXPMR	209#	213	256	419	719	1305				
ST1	1509#	1514								
ST2	1524#	1530								
SU	2764	2768								
SV	31#	1117	1121	1447	1448	1450	1451	1454	1565	1677
	1680	1689	1690	1695	1696	1788	1789	1795	1802	1808
	1812	1813	1815	1817	1818	1820	1825	1826	1828	1829
	1892	1893	2025	2026	2109	2126	2127	2129		
SVAC	94#	2151	2177	2187	2190	2283				
SVCHAR	336	343	351	353	355	356	358	362#		
SVCHR	839	845	866#							
SVF	1629	1630	1638	1640	1649#					
SVLINK	95#	2149	2162	2167	2174	2185	2192	2280		
SVPT	1198#	1204	1205	1207	1223					
SVPXR	869	876	879#							
SV1	32#	1807	1809	1811	1827	1831	1833	1837	1838	1994
SV2	33#	1803	1814	1816	1830	1832	1836	1908	1973	1995
SV3	34#	1784								
SWC	2793#	2812	2828							
SWITCH	234#	252	253	444	536	562	753	1311	1720	
SYSCNT	2799	2881#	2886							
S2201	2824	2866#								
S377	2841	2868#								
S500	2869#									
S6603	2803	2864#	2874							
S70	2806	2867#								
S7400	2848	2882#								
S7600	2810	2836#	2872							
TAILS	9#	2038	2114	3020						
TBUSY	175	821#								
TDSK	2310	2314#								
TERNR	909	916	937	958	986	1017#				
THIS	2324									
TIE	2458									
TJUMP	379#									
TK	332	348#								
TKOVER	330#	350	360	1396						
TKOVR2	19	281#	288							
TKXPBU	645#	1063								
TODISK	1413	2304#	2312							
TRANS	235	245	248	250	255#					
TRNST1	113#	421	433	535	572	810				
TRNST2	114#	443	561							
TRTURN	461	704	1707#	1736						

TRXIT	593#	604	615				
TRXIT2	608	613#					
TSIN	383	451#					
TSNUF	346	367#					
TSNUF1	295	368#					
TSOUT	384	722#					
TSTBIT	1331	1337#					
TSTCAN	2161	2293#	2297	2300			
TSTMNR	663	709#	720	808			
TSXIT	457#	476					
TSXIT1	726	745#					
TTCL	1163						
TTCR	1157						
TTDONE	1177#						
TTI	1387						
TTINCR	1177						
TTLL	1163						
TTLR	1157						
TTO	1174						
TTOGO	1168#	1180					
TTRINC	348						
TTRL	341	1181					
TTRUM	382	504#					
TTY	2226	2367#	2371				
TURN	727	742	748#	754			
TURNON	527#	768					
TURXIT	1709	1713	1718	1734#			
T1ON	1153	1543					
T1S	1153#	2205					
T1SKP	2203						
UNSTAC	101	2143#					
UP	652	667	683	707#	715		
USER	2463	2768#	2768#				
US0	2463#	2473	2482				
US1	2470#	2495					
US2	2481	2484	2488#	2492	2498	2500	2508
US3	2499	2501	2505#				
WA	1/94	1869	1870	1887	1982	1983	1994#
WARN	85#	1672	1673	1698			
WORKSP	3194#						
XACURL	655	762#					
XLIN	17	82#	290	1409			
XPA1	1030	1047					
XPA2	1045						
XPA3	1043						
XPBPTR	1036	1057	1070#				
XPDONE	1051#	1086					
XPLOOP	1042#	1050					
XPLPCN	1041	1049	1071#	1084			
XPMR	69#	211	216				
XPNUMB	66#	699	1728				
XPP	65#	791	1033	1506	1522	1722	
XPSKIP	1084#	1088					
XPSTOR	1044	1046	1048	1075#	1083		
XPTNR	1031	1032	1055	1069#			
XPWAIT	1029	1042	1087	1090#	1098		

XP2	1023#	2208										
X7FLAG	80#	523	1582									
X7P1	1589	1595	1603	1610#								
X7S	1579#	2214										
X8	26#	527	540	1436	1437	1463	1470	1473	1510			
ZERO	23#											
ZOM	3037	3121#										
ZSOLO	3041	3103#										
4L00PP	99	100	101	102	104	105	106	107	108	109		
	110	1110	1133	1187	1275	1668	1849	1931	2011	2085		
	2142	2276										
4L0064	2582	2590										
4L0066	2171	2195										
4L0067	2076	2346	2404	2417								
4L0070	2075	2163	2170	2194								
4L0071	1857	1939	2164									
4L0072	1600	1663	1753									
4L0073	1417	2369										
4L0075	1332	1363										
4L0076	1266	2145	2158									
4L0077	1243											
4L0100	1221											
4L0101	1217	1232										
4L0102	1126	1191	1583									
4L0103	1063											
4L0104	962	1147	2066									
4L0105	905	984	1004	1006								
4L0106	833	1380										
4L0107	818	827	1614									
4L0110	817	1325	1613	1763								
4L0111	792	1723	1729									
4L0112	777	1040										
4L0113	700											
4L0114	696	744										
4L0115	658	770	924									
4L0116	653											
4L0117	650	1081	1338	1405	2436							
4L0120	648	679	1321									
4L0121	584	614	1319									
4L0122	530	783	822	1333	1616	1773						
4L0123	529	782	821	1615	1756							
4L0124	521	546	554	578	695	735						
4L0125	512	576	737	1710								
4L0126	509											
4L0127	492											
4L0130	485	941	1739									
4L0131	473	515	580	701	733	1308	1617					
4L0132	461	704										
4L0133	460	514	517	654	703	739	767	1309	1712			
4L0134	459	548	587	702	734	1734						
4L0135	453	506										
4L0136	444	536	562	753	1311	1720						
4L0137	443	561										
4L0140	432	437	447	470	552	560	567	732	741	743		
	852	858	864	1719	1721							
4L0141	424	436	446	534	557	564	571	814				

[illegible]

4L2577	1298		
4L2774	1506		
4L2775	1504		
4L2776	1485		
4L2777	1429		
4L3173	1645		
4L3174	1643		
4L3175	1642		
4L3177	1628		
4L3377	1777		
4L3570	1952		
4L3571	1936		
4L3573	1906	1945	1971
4L3575	1875		
4L3577	1855		
4L3774	2046		
4L3775	2038	2114	
4L3776	2022	2027	2113
4L3777	2017	2036	2048
4L4165	2229		
4L4166	2226		
4L4167	2223		
4L4171	2217		
4L4172	2214		
4L4173	2211		
4L4174	2208		
4L4175	2205		
4L4176	2199		
4L4375	2317		
4L4376	2310		
4L4573	2444		
4L4574	2377		
4L4575	2338	2343	
4L4576	2337	2341	
4L4765	2536		
4L4766	2532		
4L4767	2522		
4L4770	2521		
4L4771	2520		
4L4772	2513		
4L4773	2511		
4L4774	2493		
4L4775	2486	2503	
4L4776	2476		
4L4777	2461	2468	
4L5161	2657		
4L5162	2619		
4L5163	2616		
4L5164	2614		
4L5165	2613		
4L5166	2610		
4L5167	2608		
4L5170	2605		
4L5174	2576		
4L5176	2570	2573	
4L5177	2564		

4L5365	2747
4L5366	2739
4L5367	2727
4L5370	2719
4L5371	2716
4L5372	2713
4L5373	2701
4L5374	2700
4L5375	2692
4L5376	2686
4L5377	2683