SPERRY RAND

UNIVAC

9200/9200 II
9300/9300 II
SYSTEMS

MINIMUM OPERATING SYSTEM

PROGRAMMERS REFERENCE

72/01/11

UP-7547 Rev. 2

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware or software changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

PRINTED IN U.S.A.

CONTENTS

1. INTRODUCTION

1.1. GENERAL

This manual provides a detailed description of the minimum operating system (MOS) designed for use with one of the UNIVAC 9200/9200 II/9300/9300 II Systems comprised of a card reader unit, a card punch unit, and at least two magnetic tape units or a direct access disc subsystem as input/output devices. The information presented concerns the various instructions required to generate, link, and load into main storage the two major programs which comprise the MOS: the supervisor and the job control program. Information concerning the interpretation of displays as well as information concerning operating instructions is also provided.

A knowledge of the *UNIVAC 9200/9200 II/9300/9300 II Systems Card Assembler Programmers Reference, UP-4092* (current version), *UNIVAC 9200 II/9300/9300 II Systems Tape Sort Programmers Reference Manual, UP-4142* (current version), *UNIVAC 9200 II/9300/9300 II Systems Magnetic Tape Input/Output Control System Programmers Reference, UP-4135* (current version), and *UNIVAC 9200/9200 II/9300/9300 II Systems Disc Input/Output Control System Programmers Reference Manual, UP-7639* (current version) is helpful in using this manual.

A knowledge of the *UNIVAC 9200/9200 II/9300/9300 II Systems Card Assembler Programmers Reference,* UP-4092 (current version), *UNIVAC 9200 II/9300/9300 II System Tape Sort Programmers Reference Manual,* UP-4142 (current version), *UNIVAC 9200 II/9300/9300 II Systems Disc Input/Output Control System Programmers Reference Manual, UP-7639* (current version) is helpful in using this manual.

The minimum operating system for the UNIVAC 9200/9200 II/9300/9300 II Systems is comprised of the supervisor and the job control program. Both are stored on cards and are loaded from the card read unit. All programs to be executed must be stored on cards; tape or disc may be used for either input data or output data.

The services provided by the minimum operating system include:

- Handling input/output interrupts as they occur and transferring control to the appropriate subroutine for interrupt analysis, possible error recovery, and subsequent control of the input/output unit;
- Coordination of operator-computer communication through displays and by the keyin of information, solicited or unsolicited;
- Handling communications interface with the supervisor when communications devices are used by a problem program. The interface is described in Appendix A.

The minimum operating system functions are done automatically as required. However, the supervisor allows computer-to-operator communication with a possible solicited reply from the operator in response to a macro instruction executed by the program. The supervisor also allows operator-to-computer communication in response to unsolicited keyins from the operator.

1.2. MACRO INSTRUCTIONS

A macro instruction is similiar in format to a source code instruction. It may or may not contain an entry in the label field, but it must contain an operation code in the operation field and one or more parameters in the operand field. The macro instructions described in this document are classified as either declarative macro instructions or imperative macro instructions. The declarative and imperative macro instructions differ in three aspects: the purpose for which they are used, the format of the parameters specified in their operand fields, and the type of code they generate.

1.2.1. Declarative Macro Instructions

Declarative macro instructions use keyword parameters to describe to the system all the aspects of the file to be processed. These aspects include parameters, constants, storage areas, special conditions, status, and options. Essentially, the declarative macro instruction defines each file required by the problem program. The code generated by the declarative macro instruction is nonexecutable and therefore should be separated from the inline file processing code.

The term keyword parameter refers to parameters which can be written in any order within the operand field. Keyword parameters must be separated by commas, but it is not required that the omission of a keyword be indicated. Keyword parameters are recognizable by their format which consists of a word or code immediately followed by an equals (=) sign which is, in turn, followed by one specification.

1.2.2. Imperative Macro Instructions

Imperative macro instructions are used to point to the files described by the declarative macro instructions. In addition, imperative macro instructions are also used in providing additional details specifying the processing action to be taken. When executed, the imperative macro instruction generates many lines of inline, executable code.

The parameters contained in the operand field of the imperative macro instruction are positional parameters rather than the keyword parameters used in the declarative macro instruction. Positional parameters, as signified by their name, must be written in the order specified and separated by commas. When a positional parameter is omitted, the comma must be retained to indicate the omission except in the case of omitted trailing parameters.

1.3. STATEMENT CONVENTIONS

The conventions used to illustrate statements in this manual are:

- Capital letters and punctuation marks (except braces, brackets, and ellipsis) are information that must be coded exactly as shown.
- Lowercase letters and terms represent information that must be supplied by the programmer.
- Information contained within braces ({}) represents necessary entries one of which must be chosen.
- Information contained within brackets ([]) represents optional entries that (depending on program requirements) are included or omitted. Braces within brackets signify that one of the entries must be chosen if that operand is included.
- An ellipsis (...) indicates the presence of a variable number of entries.

Commas are required after each parameter, except after the last parameter specified. When a positional parameter is omitted from within a series of parameters, the comma must be retained to indicate the omission.

2. GENERATING AND LINKING INSTRUCTIONS

2.1. GENERAL

The minimum operating system is maintained as object code card decks of the supervisor and the job control program. These object code card decks are created by the generation, assembly, linking, and loading of various source modules that provide the parameters and specifications necessary for the building of the supervisor and job control programs. Generation of these modules is accomplished by use of macro instructions which, when submitted to the preassembly macro pass, indicate the generation of the selected source code modules. The assembly of the source code modules is performed by the card assembler which produces the modules in relocatable code. The modules are then linked by the linker program resulting in the object version of the supervisor and the job control program. The object versions of these programs can then be loaded into main storage by use of the card load routine.

This section describes the macro instructions, control cards, and routines required for the generation, assembling, linking, and loading of the supervisor and the job control program.

2.2. THE SUPERVISOR

The supervisor is comprised of the supervisor source code module, the logical and physical unit tables, and, if desired, the tape and/or the disc dispatcher. Descriptions are provided for the macro instruction required for generating and linking these modules to create the supervisor. A discussion of the card load routine used to load the supervisor into main storage is also provided.

2.2.1. Macro Instruction for Supervisor Source Code Module Generation

The MOS (minimum operating system) declarative macro instruction enables the programmer to specify the parameters from which the preassembly macro pass will generate the required supervisor source code module. The source code output from the preassembly macro pass is assembled by the card assembler program which produces a relocatable program deck. This deck will be linked with the LU and PU tables and the disc dispatcher, if required, to create a loadable supervisor.

The following is the format of the MOS macro instruction which shows the required and optional keyword parameters which may appear in the operand of the instruction. A description is provided for each of the parameters of the MOS instruction and for the END card which is used with the MOS macro instruction.

LABEL	♭ OPERATION ♭	OPERAND
[name]	MOS	ALTR = {ALL / FOUR}, SIZE = {8 / 12 / 16 / 24 / 32} [,TPC1=n] [,TPC2=n] [,DSC1=n] [,COMM=YES] [,STSZ=n] [,CTSZ=n] [,DCU=YES]

- ALTR (Alterations) Keyword Parameter

  This required keyword parameter pertains to alterations to main storage.

  – ALTR=ALL

    This format of the ALTR keyword parameter specifies that main storage alterations are to be unrestricted.

  – ALTR=FOUR

    This format of the ALTR keyword parameter specifies that main storage alterations are limited to location 4.

- SIZE Keyword Parameter

  This required keyword parameter specifies the actual size of main storage.

  – SIZE=8

    This format of the SIZE keyword parameter specifies a main storage of 8K (8191) bytes.

  – SIZE=12

    This format of the SIZE keyword parameter specifies a main storage of 12K (12,287) bytes.

  – SIZE=16

    This format of the SIZE keyword parameter specifies a main storage of 16K (16,383) bytes.

– SIZE=24

This format of the SIZE keyword parameter specifies a main storage of 24K (24,575) bytes.

– SIZE=32

This format of the SIZE keyword parameter specifies a main storage of 32K (32,767) bytes,

- TPC1 (Tape Channel Number) Keyword Parameter

This optional keyword parameter is used to specify the channel number to which the UNISERVO VI C magnetic tape units are connected. This parameter is omitted for those MOS configurations which do not include UNISERVO VI C magnetic tape units. The format for the TPC1 keyword parameter is:

TPC1=n

where n is the decimal number of the channel to which the tape units are connected.

- TPC2 (Tape Channel Number) Keyword Parameter

This optional keyword parameter specifies the channel number of the second tape channel for those MOS configurations in which a second UNISERVO VI C magnetic tape unit channel exists. This keyword parameter is omitted if the MOS configuration does not provide a second tape channel or if UNISERVO VI C magnetic tape units are not included in the MOS configuration. The format for this optional keyword parameter is:

TPC2=n

where n is the decimal number of the second tape channel.

- DSC1 (Disc Channel 1) Keyword Parameter

This optional keyword parameter specifies the channel number to which the UNIVAC 8410 disc unit is connected. The DSC1 parameter is omitted for those MOS configurations which do not contain discs. The format for the DSC1 keyword parameter is:

DSC1=n

where n is the decimal number of the channel to which the 8410 disc unit is connected.

- COMM (Communications) Keyword Parameter

This optional keyword parameter is used to specify that the Data Communications Subsystem handler is required during the execution of the problem program. If communications handling is not required, then this keyword parameter is to be omitted. The format for the COMM parameter is:

COMM=YES

- STSZ (Scan Table Size) Keyword Parameter

  This optional keyword parameter is used to specify the number of entries contained in the scan table. This parameter is omitted when communications handling is not required during execution of the problem program. The format for the STSZ keyword parameter is:

  STSZ=n

  where n is the decimal number for the number of entries in the scan table.

- CTSZ (Clocking Table Size) Keyword Parameter

  This keyword parameter specifies the number of entries contained in the clocking table. This parameter is omitted when communications handling is not required during execution of the problem program. The format for the CTSZ keyword parameter is:

  CTSZ=n

  where n is the decimal number for the number of entries contained in the clocking table.

- DCU (Disc Unit) Keyword Parameter

  This optional keyword parameter is necessary if the 8411/8414 disc dispatcher is to be linked to the minimum operating system. This keyword parameter is omitted if the 8411/8414 disc dispatcher is not required. The format for this keyword parameter is:

  DCU=YES

2.2.1.1. END Card Requirement

The MOS macro instruction card must be followed by the END card. This card is passed through the preassembly macro pass to become the END card for the assembler performed by the card assembler program. The END card indicates to the assembler that the last card has been encountered. The format of the END card is:

LABEL	♭ OPERATION ♭	OPERAND
unused	END	E?XS

2.2.2. Macro Instructions for Logical Unit and Physical Unit Table Generation

The logical unit (LU) and the physical unit (PU) tables must be generated into a module and assembled before they can be linked in the building of the supervisor and the job control program. These tables are generated by use of CFGxx (system configuration) and the PUTBL (physical unit table) macro instructions. The module created by the LU and PU tables need be generated only once and then linked as part of the supervisor and the job control program.

The information used to create the LU and PU tables is obtained from the following hardware specifications:

- The channel number to which each peripheral (I/O) unit is assigned.
- A unit number for each magnetic tape unit of the system configuration.
- The type of heads (9-track or 7-track) installed for each tape unit.
- Whether or not the data conversion feature is installed for each control unit.

The logical unit numbers assigned to each peripheral device are supplied by the user. The tape units are assigned decimal numbers 0, 1, 2, . . . , n–1, where n is equal to the total number of tape units in the system configuration. The remaining peripheral devices are assigned hexadecimal numbers n, n+1, ... , 63 without regard to device type or channel.

The user also assigns physical unit numbers to each of the peripherals in the system configuration, irrespective of the channel assignment of the unit. These numbers, which may be 0 through 15, are usually assigned sequentially for multiple units such as the tape units.

Descriptions and formats are provided for the macro instructions used to generate the LU and PU tables.

2.2.2.1. CFGxx Macro Instruction

This macro instruction contains the information necessary to set up the variable-length LU and PU tables according to the number of devices in the system configuration. The format for the CFGxx macro instruction is:

LABEL	OPERATION	OPERAND
unused	CFGxx	unused

The xx position of the operation code is a two-digit hexadecimal number which represents the highest logical unit number assigned in the system configuration.

2.2.2.2. PUTBL Macro Instruction

The PUTBL macro instruction is used to specify the parameters for each entry contained in the LU and PU tables. Therefore, a PUTBL macro instruction card must be prepared for each peripheral device in the system configuration.

The PUTBL macro instruction cards for the tape and/or disc devices must be arranged by channel/unit number in ascending sequence and must immediately follow the CFGxx macro instruction card. If both tape and disc units are present in the system, a decision must be made as to which cards come first (tape cards or disc cards). The decision made is based upon the orientation of the system. That is, a system having its system software on tape is considered a tape-oriented system and the cards for the tape devices precede those for the disc devices. If the system software is on discs, then the system is considered a disc-oriented system and the cards for the disc devices precede those for the tape devices. Other devices are arranged after the tape and/or disc device cards and must be arranged so that the logical unit numbers are in continuous, ascending sequence. An END card must be the last card in the deck.

This deck of macro instructions, the CFGxx card and the several PUTBL cards, comprises the input to the preassembly macro pass which then generates the source coding for the module for the logical and physical unit tables.

The source code module of the tables is then used as input to the card assembler. The output from the assembly is the relocatable code module which can now be linked first with the supervisor module and then with the job control program module.

The format for the PUTBL macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	PUTBL	{ TAPE | CRD | CRP | PRNT | RRP | CC | CDVC | DISC | DPRT | PPT | RDR4 | PRT4 | PCH4 | OCR }, device-channel-number, hardware-unit-number, physical-unit-number, logical-unit-number [, { 7 | 9 | B | C }] [, { ss | channel-number }]

- Positional Parameter 1

  TAPE – specifies magnetic tape unit device

  CRD – specifies card reader device

  CRP – specifies card read/punch device

  PRNT – specifies printer device

  RRP – specifies row read/punch device

  CC – specifies card controller

  CDVC – specifies communications device

  DISC – specifies 8410/8411/8414 DAS device

  DPRT – specifies 0768 drum printer

  PPT – specifies paper tape reader/punch

  RDR4 – specifies 1004 reader

  PRT4 – specifies 1004 printer

  PCH4 – specifies 1004 punch

  OCR – specifies optical document reader

- Positional Parameter 2

  device-channel-number – the decimal number (1–12) for the channel on which the device specified in positional parameter 1 is placed.

  – the receiving channel number if the device specified by positional parameter 1 is a communications device.

  – the decimal number 16 if the device specified by positional parameter 1 is an 8411/8414 DAS unit.

- Positional Parameter 3

  hardware-unit-number – the decimal number (0–7) for the hardware unit of the specified channel.

  – the hardware device address if the channel is for the 8411/8414 DAS unit.

- Positional Parameter 4

  physical-unit-number – two-digit hexadecimal number (0–15) assigned to each device on the system configuration; multiple units of a given type, such as tape units, should be numbered sequentially starting with zero.

- Positional Parameter 5

  logical-unit-number – two-digit hexadecimal number (0–63) assigned to a peripheral device.

- Positional Parameter 6

  7 or 9 – specifies number of tracks when TAPE is specified in positional parameter 1.

  B – used only when CDVC is specified in positional parameter 1 and the device address format is 0100CCCC.

  C – used only when CDVC is specified by positional parameter 1 and the device address format is 0000CCCC.

NOTE: This positional parameter is omitted if anything other than TAPE or CDVC is specified by positional parameter 1.

- Positional Parameter 7

  ss – a two-digit hexadecimal value that indicates tape density and mode. The ss value is used only when 7-track TAPE is specified by positional parameter 6. The values for ss are the same as those of the ASSGN control card; refer to 5.4.4 for possible values for ss.

  channel-number – the number of the transmitting channel when CDVC is specified by positional parameter 1.

NOTE: This positional parameter is omitted if positional parameter 1 specifies anything other than TAPE or CDVC or if positional parameter 6 specifies the value 9 (9-track tape) when TAPE is specified by positional parameter 1.

The ss specified for a 7-track tape establishes the normal mode for reading and writing on the tape unit. This establishes the initial value of byte 2 of the associated PU table entry.

2.2.3. Macro Instructions for 8410, 8411, and 8414 Disc Dispatcher Generation

The disc dispatchers provide an interface between the problem program and the individual disc drives. The disc dispatchers are called upon by the problem program to perform the functions necessary to execute all disc input/output commands and to handle the resulting interrupts, as well as error conditions.

The generation of the disc dispatchers is accomplished by submitting the proper macro instruction to the preassembly macro pass. A description is provided for the generation of the 8410 disc dispatcher and the 8411/8414 disc dispatcher.

2.2.3.1. Macro Instruction for 8410 Disc Dispatcher Generation

The DISC declarative macro instruction enables the programmer to specify the parameters from which the preassembly macro pass will generate the required 8410 disc dispatcher source code module. Once generated, this module must be assembled and then linked with the supervisor and the LU and PU tables. The format of the DISC macro instruction required to generate the 8410 disc dispatcher is as follows:

LABEL	♭ OPERATION ♭	OPERAND
[name]	DISC	DSC1=n [, STOP=YES]

- DSC1 (Disc Channel Number) Keyword Parameter

  This keyword parameter specifies the channel number to which the 8410 disc unit is connected. The formatof the DSC1 parameter is as follows:

  DSC1=n

  where n is the decimal number of the channel to which the 8410 disc unit is connected.

- STOP Keyword Parameter

  The STOP keyword parameter is an optional parameter. When specified, the STOP parameter causes code to be generated which permits the programmer to retry unsuccessful disc calls. For example, if the STOP parameter has been specified in the DISC macro instruction an unsuccessful attempt to call a disc will cause a hexadecimal display stop of 66us to occur. The u in this display stop specifies the physical unit number of the disc unit involved in the error and the s represents the 4 most significant error bits of the status byte. The programmer may retry the disc call by performing the procedure described in 3.2.6.

  The format for the STOP keyword parameter is as follows:

  STOP=YES

  where YES specifies that the capability of retry for unsuccessful disc calls is required. If this capability is not desired, the STOP parameter must be omitted.

2.2.3.2. Macro Instruction for 8411/8414 Disc Dispatcher Generation

The 8411/8414 disc dispatcher is generated when the DISK macro instruction is submitted to the preassembly macro pass. The preassembly macro pass generates the required 8411/8414 source code.

LABEL	♭ OPERATION ♭	OPERAND
[name]	DISK	DCU{1...7}=n, UNT{1...7}=n

- DCU (Disc Control Unit) Keyword Parameter

  This required keyword parameter specifies base address for disc control units 1 through 7. The format of the DCU keyword parameter is:

  DCU{1...7}=n

  where n is the hexadecimal hardware base address of the selected disc control unit (units 1 through 7).

- UNT (Unit Number) Keyword Parameter

  This required keyword parameter specifies the number of the 8411/8414 disc units connected to the corresponding disc control unit. For example, the parameter UNT1 specifies the number of disc units connected to DISC control unit 1 and so on. The format of the UNT keyword parameter is:

  UNT{1...7}=n

  where n is the decimal number of the 8411/8414 disc units connected to the corresponding disc control unit specified by the DCU keyword parameter.

The relocatable module produced by the assembly of the source code output of the preassembly macro pass should be immediately preceded by the following linker control card:

LABEL	OPERATION	OPERAND
[name]	MOD	8

The module will then be included in the supervisor linker input deck immediately preceding the LUPU module.

2.2.4. Macro Instruction for 1004 Handler Generation

The generation of the 1004 handler is required when the MOS utilizes the UNIVAC 1004 as a subsystem. The 1004 handler is a separate PROC which is generated independently of the problem program and then linked for use with the MOS.

The format of the declarative macro instruction required for generating the 1004 handler is as follows:

LABEL	OPERATION	OPERAND
[name]	T4HR	TEN4=n

where n is the decimal number of the channel to which the 1004 subsystem is connected.

2.2.5. Linking and Loading the Supervisor

The supervisor is created when the supervisor source code modules, the LU and PU tables, and, if desired, the tape and/or the disc dispatchers are linked together and loaded into main storage. These modules, which have been assembled by the card assembler program and are in relocatable code, are linked together by means of the CTL and PHASE linker control cards. In order to load the supervisor, the modules must also be linked to the card load routine. A two-pass run of the linker is required to link the various modules of the supervisor together.

2.2.5.1. Linker Control Cards

The CTL and the PHASE linker control cards are the two cards used for linking the modules used to generate the supervisor. These two control cards must be the first two cards in the input deck to the linker. The last card must be an END card. The supervisor source code module should precede the module for the LU and PU tables. If the tape and/or the disc dispatchers are to be included, then the modules for these dispatchers should be inserted between the supervisor module and the module for the LU and PU tables. The format for the linker control cards is as follows:

- CTL Linker Control Card

The CTL linker control card is the first card of the input deck to the linker. This card is used to specify the number of passes to be performed, and the characteristics of main storage size and highest available address. The format of the CTL card is:

LABEL	OPERATION	OPERAND
[name]	CTL	2 , { 8191 / 12,287 / 16,383 / 24,575 / 32,767 } , { 8191 / 12,287 / 16,383 / 24,575 / 32,767 }

- Positional Parameter 1

  2 – denotes a two-pass operation of the linker.

- Positional Parameter 2

  8191 – decimal number representing the largest available address during linking (8K main storage).

  12,287 – decimal number representing the largest available address during linking (12K main storage).

  16,383 – decimal number representing the largest available address during linking (16K main storage).

  24,575 – decimal number representing the largest available address during linking (24K main storage).

  32,767 – decimal number representing the largest available address during linking (32K main storage).

- Positional Parameter 3

  8191 – decimal number representing the largest address available to the output element for an 8K main storage.

  12,287 – decimal number representing the largest address available to the output element for a 12K main storage.

  16,383 – decimal number representing the largest address available to the output element for a 16K main storage.

  24,575 – decimal number representing the largest address available to the output element for a 24K main storage.

  32,767 – decimal number representing the largest address available to the output element for a 32K main storage.

- PHASE Control Card

  The PHASE linker control card is the second card of the input deck to the linker. This card is used to define the name and initial storage address for the output element. The format of the PHASE card is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	PHASE	SUPR,256,A

- Positional Parameter 1

  SUPR – denotes that the supervisor is the phase name.

- Positional Parameter 2

  256 – decimal number representing the starting address at which the supervisor will be loaded into main storage (applies only when MOS, REV B is specified).

- Positional Parameter 3

  A – specifies that load address specified by positional parameter 2 is actual value.

2.2.5.2. Card Load Routine

The card load routine is used to load the supervisor into main storage starting at the location specified by the parameters of the PHASE linker control card. This routine may also be used to load the object version of the job control program or any problem (user's) program. A card load function is performed by placing the card load routine in front of the object version of the particular program to be loaded and by following the procedures for the card load routine. Procedures for the card load routine are described in *UNIVAC 9200/9200 II/9300/9300 II Card Assembler Programmers Reference, UP-4092* (current version).

Certain external definition cards must be used in creating a card load routine for the supervisor. The format and sequence of these cards is as follows:

LABEL	♭ OPERATION ♭	OPERAND
L?AR	EQU	7400
L?LO	EQU	128
L?HI	EQU	256
L?PG	EQU	80,L?AR
L?CH	EQU	X'A9'
L?AM	EQU	4

The EQU card is used to define (equate) the labels used by the card load routine with a specific value. The labels of the external definition cards and their meanings are provided in the following list.

LABEL	MEANING
L?AR	Start of the read area for the load routine (7400)
L?LO	First memory location to be cleared (128)
L?HI	Last memory location to be cleared (256)
L?PG	Start of the coding of the card load routine (80, L?AR)
L?CH	Character with which to fill area to be cleared (X 'A9')
L?AM	Specifies main storage location where alterations are to be stored (4)

The specification of L?AM can be overriden by the specification of the ALTER parameter of the MOS macro instruction.

There must be a guarantee that no part of the supervisor is destroyed when the problem program is loaded. The supervisor occupies main storage locations 256 through the location labeled E?HI. The absolute address of E?HI can be determined from the linker output listing produced when the supervisor was linked.

2.3. THE JOB CONTROL PROGRAM

The job control program is comprised of the job control source code module, the logical and physical unit tables, the card read routine, and the card-code-to-EBCDIC code translation table. A description of the macro instruction and control cards required for the generation and linking of the modules required to create the job control program is provided in this section. Since the macro instructions for the logical and physical unit tables need be generated only once and since the same card load routine is used for both the supervisor and the job control program, no further explanation is provided. See 2.2.2 and 2.2.5.2 for information concerning generation of the logical and physical unit tables and the card load routine.

2.3.1. Macro Instruction for Job Control Program Source Code Module Generation

The JCMS macro instruction is used by the preassembly macro pass to generate the relocatable source code module for the job control program. A description is provided for both the JCMS macro instruction and for the required END card used with this instruction.

The format of the JCMS macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	JCMS	CSR= { CRD / CC } [,COMM=YES]

- CSR Keyword Parameter

  The CSR keyword parameter specifies the device used for the input control cards; for example:

  – CSR=CRD

  This format of the CSR keyword parameter specifies that the device for the input control cards is the serial reader.

  – CSR=CC

  This format of the CSR keyword parameter specifies that the device for the input control cards is the card controller.

- COMM (Communication) Keyword Parameter

  This optional keyword parameter specifies whether communications handling is required. If communications handling is required, the following format must be specified for the COMM parameter:

  COMM=YES

  If communications handling is not required, this keyword parameter is omitted from the JCMS macro instruction.

2.3.1.1. END Card Requirement

The END card is used to indicate the last card in the deck to the assembler. The format of the END card used with the JCMS macro instruction is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	END	STRT

2.3.2. Declarative Macro Instructions for Job Control Reader IOCS

The control cards for the job control program are read from either the serial reader or the card controller, depending on which device is specified in the operand of the JCMS macro instruction. A declarative macro instruction is required to define the input device specified. A description is provided for the macro instruction required for both devices.

2.3.2.1. Macro Instruction for Serial Reader

When the serial reader is used as the input device, the following declarative macro instruction is required as input to the preassembly macro pass:

LABEL	↓ OPERATION ↓	OPERAND
CARD	DTFCR	10A1=E?IB,ITBL=TBRD,SENT=NO,MODE=TRANS

- IOA1=E?IB (Input/Output Area) Keyword Parameter

  This required keyword parameter specifies the address of the input buffer area.

- ITBL=TBRD (Input Translation Table) Keyword Parameter

  This required keyword parameter specifies that the input translation table labeled TBRD is to be used. TBRD is the EBCDIC input translation table which is included as a relocatable module in the card libraries of the UNIVAC 9200/9300 software package. This module must be included with the relocatable source code module of the reader IOCS during the linker run which produces the object version of the job control program.

  A different EBCDIC translate table may be used. This may be inserted in the system by altering the ITBL specification and by including the relocatable module of the translate table in the linker run.

- SENT=NO (End of File Sentinel) Keyword Parameter

  This required keyword parameter specifies that no end of file sentinel is to be recognized.

- MODE=TRANS Keyword Parameter

  This required keyword parameter specifies that the cards are to be read after the translation specified by the ITBL keyword parameter has been performed.

2.3.2.2. Macro Instruction for Card Controller

When the UNIVAC 1001 Card Controller is used as the input device, the following declarative macro instruction is required as input to the preassembly macro pass to produce the appropriate job control input routine:

LABEL	↓ OPERATION ↓	OPERAND
CARD	DTFCC	MODE=TRANS,ITBL=TBRD,FUNC=CCXF*,CHNL=n

- MODE=TRANS Keyword Parameter

  This required keyword parameter specifies that input cards are to be read after they have been translated by the translation table specified in the ITBL keyword parameter.

- ITBL=TBRD (Input Translation Table) Keyword Parameter

  This required keyword parameter specifies that the input EBCDIC translation table labeled TBRD is to be used (see 2.3.2.1 for a detailed explanation of TBRD).

- FUNC=CCXF (Function) Keyword Parameter

  This keyword parameter identifies the function area label CCXF as the area where the user-supplied function code is stored.

- CHNL=n (Channel) Keyword Parameter

  This keyword parameter specifies the channel number to which the card controller is connected. The value for n can be any decimal value between 5 and 12.

2.3.3. Linking and Loading the Job Control Program

The job control program is completed when the job control module, the LU and PU tables, the card read unit routine, and the module of the card-code-to-EBCDIC-code translate table have been linked by the linker and loaded into main storage by the card load routine.

Linking is accomplished by use of the CTL and the PHASE linker control cards. These two control cards are the first two cards in the input deck. They must be followed by the job control module, the LU and PU table modules, the card read unit routine, the translate table module, and an END card in the order listed. A two-pass run of the linker is required to link these modules together. The output of the linker run becomes the object version of the job control program which can then be loaded by means of the card load routine. The card load routine is the same as that used for loading the supervisor. See 2.2.5.2 for a detailed discussion of the card load routine.

2.3.3.1. Linker Control Cards

The CTL and the PHASE linker control cards are the two cards used for linking the modules that comprise the job control program. The CTL linker control card is identical in format to the CTL linker control card used in linking the supervisor. See 2.2.5.1 for a detailed explanation of the CTL linker control card. A description for the PHASE linker control card is provided in the paragraph that follows.

The PHASE control card is used to define the name and initial storage address for the output element (job control program) produced by the linker. The format of the PHASE card used in the linking of the job control program is:

LABEL	♭ OPERATION ♭	OPERAND
[name]	PHASE	JCMS,S,A

- Positional Parameter 1

  JCMS – denotes that the job control program is the phase name.

- Positional Parameter 2

  S – decimal value which represents the starting address at which the job control program is to be loaded. The starting address specified by this parameter must be greater than the highest address occupied by the supervisor (represented by E?HI on the linker output listing).

- Positional Parameter 3

  A – specifies that the address specified in positional parameter 2 is actual value.

3. OPERATING INFORMATION

3.1. GENERAL

The operating information presented in this section pertains to the interpretation of displays which may occur during program execution and to the instructions required for loading the supervisor and the job control program.

3.2. DISPLAYS

Displays provide a visual means of communications between the computer and the operator. The operator can interpret the various hexadecimal displays to determine the status of program runs, errors which may have occurred, operator action required, and functions that are to be performed next. Displays are provided for the supervisor, tape dispatcher, job control program, communication devices, and 8410 and 8411/8414 disc dispatchers. The various displays provided are listed with the reason for each display and if applicable, the necessary actions required by each display. The listings are grouped by the program responsible for generating the display.

3.2.1. Supervisor Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
41FF	Load Stop Supervisor is ready; load the first program.
41EF	End-of-Job Load the next program.
41EE	Cancel Load the next program.
413F	SRC or keyin table entry accessed when blank (not initialized) Press START switch to ignore the interrupt.

3.2.2. Tape Dispatcher Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
60x0	The START switch has been pressed in an attempt to recover from an unrecoverable error.
60x1	The tape dispatcher has tried five times, unsuccessfully, to recover from a read or write tape error. To cause the tape dispatcher to skip the request packet on which the error is occurring and go on to the next request packet, key a 1 into memory location 4 before pressing the START switch.
60x2	The invalid command sense bit (bit 0 of sense byte 0) has been set. One cause of this error is an attempt to perform a write, write tape mark, or erase operation on a file-protected tape unit. In this case, the operation can be effected by inserting the write enable ring in the tape reel and pressing the START switch.
60x3	The noise bit (bit 0 of sense byte 1) has been set while an erase command was being executed. To continue, key a 1 into memory location 4 and then press the START switch.
60x4	The equipment check bit (bit 3 of sense byte 0) has been set. This is a nonrecoverable error.
60x5	The noise bit (bit 0 of sense byte 1) and the tape fault bit (bit 6 of sense byte 4) have been set during a write or write tape mark operation. This is a nonrecoverable error.
60x7	The intervention required bit (bit 1 of sense byte 0) has been set. One cause of this error is the attempt to perform a tape operation on a nonready unit. In this case, recovery consists of making the unit ready and pressing the START switch.
60xC	A condition code of 01 has been returned when initiating the current tape operation. This generally occurs because the function presented to the control unit is invalid. To retry initiating the operation, depress the START switch. If the error persists, the job should be canceled.
60xD	Unit check has occurred but all bits of sense bit 0 are reset. A read backward, backspace block, or backspace file operation has been attempted on a tape unit on which the tape is at load point.

DISPLAY (HEXADECIMAL)	REASON AND ACTION
60xE	When the supervisor is generated, the channels for tape control units are specified as parameters. When a request is made of the tape dispatcher, the request specifies a logical unit number. The tape dispatcher verifies that the physical unit table entry specified by the logical unit number contains a channel number that corresponds to one of those specified when the supervisor was generated. If this is not the case, a 60xE display is made. This is a nonrecoverable error.
60xF	A condition code of 11 has been returned when initiating the current tape operation. This indicates a nonoperational GPC. Ready the GPC and press the START switch.

To cancel, an unsolicited keyin must be made while the computer is stopped. The DATA ENTRY switches must be set to F0. Before pressing the START switch, a 1 must be keyed into location 4 to enable the computer to enter the processor PSC so it may accept the cancel keyin.

The x in the display carries the physical unit number of the UNISERVO VI C magnetic tape unit involved with the error.

3.2.3. Job Control Program Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
4101	Second unit already allocated in SWAP. Replace SWAP card; reinitialize card reader; press START.
4102	Control card out of sequence. Resequence control cards and reinitialize reader; press START.
4103	Tape unit not 7-track tape in ASSGN. Replace ASSGN card; reinitialize reader; press START.
4108	Second unit listed on ALT card not free. Replace ALT card; reinitialize reader; press START.
4109	Second unit listed on ALT card not up. Replace ALT card; reinitialize reader; press START.
410C	Opcode in MTC card cannot be interpreted. Replace MTC card; reinitialize reader; press START.
410F	Device described on a DVCDN or DVCUP control card cannot be located in the physical unit table. Replace card; reinitialize reader; press START.
41CF	Not a control card. Replace card; reinitialize reader; press START.
41CE	Cannot locate file-ID specified in FILES control card. Reload control card; press START after rewinding file.

3.2.4. Communications Devices Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
4120	Addition to scan table cannot be made. Table size would be exceeded. Press START switch to cancel.
4121	Addition to clocking table cannot be made. Table size would be exceeded. Press START switch to cancel.

3.2.5. Card Load Routine Displays

The 61xx displays are described under card load routine errors in the *UNIVAC 9200/9200 II/9300/9300 II Card Assembler Programmers Reference, UP-4092* (current version).

3.2.6. 8410 Disc Dispatcher Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
66us	Occurrence of an error during an attempt to issue a disc order. The error is indicated by s which represents the four most significant error bits of the status byte. To retry the order, resubmit the order by pressing the START switch on the display console. If this fails to produce the desired results, pass status onto the problem program by keying in a nonzero character other than a hexadecimal E7 into location 4 and then press the START switch. This action causes the error to be returned to the I/O routine and a standard stop to occur.
67ul	Request packet is a member of an existing chain of packets being processed. Pressing the START switch with no keyin causes the Disc Dispatcher to process the packet. This will produce erroneous results and the operator should make an unsolicited external cancellation keyin.
67uF	Disc I/O command was issued for a nonperational channel. To retry the operation, correct the situation and press the START switch. To skip the operation, key a nonzero into location 4 and press the START switch.

The u in the display carries the physical unit number of the Univac disc unit involved with the error.

3.2.7. 8411/8414 Disc Dispatcher Displays

DISPLAY (HEXADECIMAL)	REASON AND ACTION
69u1	Nonoperational channel (a cause of this display could be an attempt to address a nonexistent channel or device). Press START switch to skip packet and go on to the next request.
69u2	A data error has occurred on the channel or on the device (command chain has been reissued ten times if bit 0 of byte 8 is a 0). Press START switch to retry. Keyin a nonzero in location 4 and press START switch to skip packet and go on to next request.
69u3	Intervention required (a cause of this type of display could be the device being offline or an attempt to write on a device for which the FILE PROTECT switch has been pressed). Press START switch to retry. Keyin a nonzero into location 4 and press START switch to skip packet and go on to next request.
69u4	An invalid command sequence was attempted. Press START switch to skip packet and go on to next request.
69u5	The command address word in the request packet is not set to a double word boundary. Press START switch to skip packet and go on to next request.
69u6	The device identification does not match the ID in the physical unit table. (A cause of this display could be an attempt to address a nonexistent device.) Press START switch to skip packet and go on to next request.

The u in the display will carry the physical unit number of the disc unit involved with the error.

3.3. OPERATING INSTRUCTIONS

The following steps are required to load the supervisor:

(1) Initialize the serial reader or the card controller by inserting the loader deck followed by the supervisor card deck into the input hopper. One blank card must follow the deck.

If the serial reader is used, press the CLEAR and FEED switches.

If the 1001 card controller is used:

(a) Set ALT 1 switch to the ON position.

(b) Press LOAD PRI (load primary) switch.

(c) Press CLEAR, START, and RUN switches.

(2) Set the device address of the proper card read unit into the DATA ENTRY switches.

(3) Press the processor CLEAR switch.

(4) Press the LOAD switch.

(5) Press the START switch.

(6) Reset the LOAD switch.

(7) Press the START switch. The program will be loaded.

(8) After the supervisor is loaded, it stops and a hexadecimal 41FF display is generated.

At this time, the job control program may be loaded and executed by following the above procedure. When it has finished, the job control program executes an EOJ display, 41EF.

The job control program only needs to be loaded initially or between jobs if its services are required. Otherwise, the user may directly follow the loading of the supervisor by loading and executing the programs of his choice. The supervisor need not be loaded between programs unless it has been accidentally destroyed or if an external cancel has taken place.

4. THE SUPERVISOR

4.1. GENERAL

The supervisor is the component of the minimum operating system that provides the basic control network required for program execution, interrupt handling, error recovery, and input/output control. The supervisor in the minimum operating system is made up of the following:

- Interrupt Table
- Supervisor Request Call (SRC) Table
- Boundary Table
- Physical Unit (PU) Table
- Logical Unit (LU) Table
- Keyin Table
- Exec Activity Sum
- Address Table
- Magnetic Tape Dispatcher (optional)
- 8410 Disc Dispatcher (optional)
- 8411/8414 Disc Dispatcher (optional)

Each of these features is explained in this section.

4.2. INTERRUPT TABLE

There are up to 32 (0 through 31) entries in the interrupt table. Each entry consists of two bytes. With the exception of the first entry, each is associated with a channel number. The first entry in the interrupt table is used to distinguish SRC interrupts from I/O interrupts. See Table 4–1 for the association of the other interrupt entries with the proper device or channel.

CHANNEL NUMBER	RELATIVE ADDRESS	DEVICE/CHANNEL INTERRUPT ADDRESS
–	0	SRC INTERRUPT
1	2	READER
2	4	READ/PUNCH
3	6	PRINTER
4		NOT AVAILABLE FOR USE
5	10	CHANNEL 5
6	12	CHANNEL 6
7	14	CHANNEL 7
8	16	CHANNEL 8
9	18	CHANNEL 9
10	20	CHANNEL 10
11	22	CHANNEL 11
12	24	CHANNEL 12
.	.	.
.	.	.
.	.	.
31	62	CHANNEL 31

Table 4–1. Interrupt Table

The interrupt table base address is at a halfword boundary, and its location is maintained in the address table for use by problem programs. If the routine addressing the interrupt table is being assembled or linked with the interrupt table, the table can be addressed by its label E?IN. The I/O control routines expecting interrupts on a particular channel must place the address of the routine handling the interrupts into the interrupt table entry associated with that channel. This must be done before the execution of any XIOF instructions specifying that particular channel.

When an interrupt occurs, the supervisor interrogates the device address byte to determine if an interrupt occurred because the operator pressed the OP REQ (operator request) switch. If it is not an operator request, the supervisor determines if it is an I/O or an SRC interrupt in the following manner. After an I/O interrupt has been serviced, the device address byte is set to zero. If the device address byte still contains a zero at the next interrupt, it is an SRC interrupt. If the byte is not zero, it is an I/O interrupt.

When an interrupt occurs which is not an operator request, the supervisor takes the channel number as found in the device address byte, doubles it, and adds the result to the base address of the interrupt table. The supervisor then loads a general register with the contents of the memory location whose address was determined by the preceding calculation and branches unconditionally to that address.

When the supervisor has branched to the problem program's interrupt routine by using the interrupt table, the processor is still in I/O mode. Therefore, when the interrupt routine has serviced the interrupt, it should branch unconditionally to the reentry routine, the address of which is a fixed location in the supervisor. At reentry, the supervisor resets (without destroying the SRC field) the I/O PSC in preparation for the next interrupt and returns to processor state control. The reentry routine can be addressed by its label E?RE if it has been assembled or linked with the routine using it.

4.3. SUPERVISOR REQUEST CALL TABLE

There are up to 24 entries in the SRC table, each consisting of two bytes. The second, third, fourth, and sixth through thirteenth entries are associated with channel numbers in the same way the interrupt table entries are associated with channel numbers. The first, fifth, fourteenth, and fifteeth through twenty-fourth entries direct control to the supervisor macro routines.

The SRC table base address is at a halfword boundary and its location is maintained in the address table for use by problem programs. If the routine addressing the SRC table is being assembled or linked with the SRC table, the table can be addressed by its label E?SC. See Table 4–2 for the association of entries with channel numbers and issue subroutines.

When an SRC entry is associated with a given channel number, an I/O control routine can execute an SRC instruction with an immediate value equal to twice the number of the channel on which its device is located. Control is returned in I/O mode. Such a method is used to prevent interrupt before issuing an input/output function specification.

When an SRC interrupt occurs, the supervisor takes the immediate value of the SRC instruction causing the interrupt and adds it to the base address of the SRC table. (The value is found in the I/O PSC.) The supervisor then loads into a general register the contents of the memory location whose address was determined by this addition, and branches unconditionally to that address.

An I/O control routine having a device located on channels 0 through 15 and expecting to use an SRC instruction to branch to the issue subroutine must place the address of the issue subroutine into the SRC table entry with which its channel is associated. This must be accomplished before the execution of any SRC instructions with an immediate value indicating that particular SRC table entry. I/O routines operating on channels 16 through 31 must use the SRC 0,34 instruction format (explained in Appendix A) to transfer control to the I/O handler at the address specified in processor register 15. This is accomplished in the I/O mode.

When the supervisor branches to the issue subroutine by using the SRC table, the processor is still in I/O mode. Therefore, when the issue subroutine exits, it should branch unconditionally to the reentry routine.

As mentioned, the first, fifth, and fourteenth entries of the SRC table direct control to the cancellation, message, and end-of-job supervisor routines, respectively. The CANCL, MSG, and EOJ macro instructions use these entries to communicate their requirements to the supervisor.

ENTRY NUMBER	IMMEDIATE RELATIVE VALUE	FUNCTION ADDRESS
1	0	CANCELLATION ROUTINE
2	2	READER ISSUE
3	4	READ/PUNCH ISSUE
4	6	PRINTER ISSUE
5	8	MESSAGE ROUTINE
6	10	CHANNEL 5 ISSUE
7	12	CHANNEL 6 ISSUE
8	14	CHANNEL 7 ISSUE
9	16	CHANNEL 8 ISSUE
10	18	CHANNEL 9 ISSUE
11	20	CHANNEL 10 ISSUE
12	22	CHANNEL 11 ISSUE
13	24	CHANNEL 12 ISSUE
14	26	END-OF-JOB ROUTINE
.	.	.
.	.	.
.	.	.
24	48	COMMUNICATIONS RELATED CHANNELS

Table 4–2. Supervisor Request Call Table

4.3.1. Supervisor Macro Instructions

As previously indicated, the supervisor responds to the CANCL, MSG, and EOJ macro instructions. The functions of these macros are described in the paragraphs which follow.

4.3.1.1. CANCL Macro Instruction

The CANCL macro instruction is used to cancel all the remaining steps of a job.

The format of the CANCL macro instruction is:

LABEL	OPERATION	OPERAND
unused	CANCL	

This macro instruction requires no operand. When presented to the preassembly macro pass, the CANCL macro instruction causes the following supervisor request call to be generated:

LABEL	OPERATION	OPERAND
unused	SRC	0,0

The supervisor responds to this code by stopping on a CANCL display. The card assembler does not incorporate the ability to assemble the CANCL macro instruction; the preassembly macro pass must be used or the cancel SRC instruction must be written explicitly. After the execution of a CANCL macro instruction, the supervisor should be loaded.

4.3.1.2. EOJ (End-of-Job) Macro Instruction

The EOJ macro instruction is used to specify the end of the job.

The format of this macro instruction is:

LABEL	OPERATION	OPERAND
unused	EOJ	

The EOJ macro instruction requires no operand. The preassembly macro pass generates the following code for the EOJ macro instruction:

LABEL	OPERATION	OPERAND
unused	SRC	0,26

The supervisor responds to this interrupt call by halting on an EOJ display on the console. The card assembler does not incorporate the ability to assemble the EOJ macro instruction.

4.3.1.3. MSG (Message) Macro Instruction

The MSG macro instruction is used to display a message on the console and, if specified, to accept a reply from the operator.

The format of the MSG macro instruction is:

LABEL	OPERATION	OPERAND
unused	MSG	message [,REPLY]

- Positional Parameter 1

  message – any two-byte hexadecimal expression supplied by the user and in the form X'nnnn'.

- Positional Parameter 2

  REPLY – an optional parameter which, when used, specifies that an operator's response is required.

When encountered, the MSG macro instruction causes the supervisor to generate the following code:

LABEL	OPERATION	OPERAND
	~~SCR~~ SRC	0,8
	DC	Y(message)
	DC	CL1['A']
	DC	X'0'

The code generated identifies the call as a message routine and provides the required define constant statements where:

message – is defined as a halfword address constant consisting of any two-byte expression expressed in Y'nnnn' form. The expression may be absolute or relocatable.

CL1['A'] – specifies that the character length is a one-byte constant consisting of the character A. The character A is an optional parameter and is used only if the REPLY parameter is specified in the MSG instruction.

X'0' – specifies a hexadecimal constant having a length of one byte containing all 0's.

The last byte in the MSG macro instruction calling sequence is referred to as the reply byte.

In response to this macro, the supervisor does a branch and link (BAL), using processor register 15, to its own display subroutine. It moves the message from the calling sequence of the SRC instruction to the calling sequence of the BAL instruction before executing the BAL instruction. The display subroutine sets location 4 to binary zero and displays "message" by means of a halt and proceed (HPR) instruction. When the START switch is pressed, the display subroutine returns control to the supervisor, which then moves the contents of location 4 to the reply byte of the calling sequence and returns control to the problem program.

If a program is in I/O mode, it must display messages by using the display subroutine of the supervisor directly. If a reply is expected after execution, the program must have its own routine to examine the contents of location 4.

The following instructions are required to execute a display:

```
LH      14,266
BAL     15,0(,14)
DC      XL2'message'
```

The above instruction is used to load register 14 with the contents of location 266. The brance and link instruction provides an unconditional branch to address 0 and stores the address of the next instruction to be executed into register 15. The constant to be displayed is defined as a two-byte hexadecimal expression by the DC instruction.

If the routine addressing the display subroutine is being assembled or linked with the display subroutine, the following instructions can be used:

```
BAL     15,E?DS
DC      XL2'message'
```

4.4. BOUNDARY TABLE

The boundary table is made up of two two-byte entries. The boundary table is immediately followed in memory by a two-byte top-of-memory entry specifying the address of the highest order byte available in memory. The contents of the top-of-memory entry are determined at the time the supervisor is generated.

4.5. PHYSICAL UNIT TABLE

The physical unit (PU) table contains the information required to identify a device, its channel number, the unit number within the channel, and certain specification information. There are as many entries in the table as there are devices. The entries are arranged in such a way that device types are grouped, and the devices appear within groups in ascending order by channel and unit number. The table has a maximum of 64 entries. The format is shown in Figure 4–1.

	0								1								2								3							
0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
For System Use	Device ID								PU Number				Allocation Code			Up or Down	D	D	M	M	M	0	1	1	Channel/Unit							

Figure 4–1. Physical Unit Table Format

- Byte 0

  Bit 0 – For System Use

  Bits 1–7 – Device ID (hexadecimal code)

  - 00 – Magnetic Tape Unit
  - 01 – Reader
  - 02 – Serial Read/Punch
  - 03 – Printer
  - 04 – Row Read/Punch
  - 05 – 1001 Card Controller
  - 06 – Communications Device
  - 07 – 8410 DAS Device
  - 08 – 8411/8414 DAS Device
  - 09 – Paper Tape
  - 0A – 1004 Reader
  - 0B – 1004 Printer
  - 0C – 1004 Punch
  - 0D – Optical Document Reader

- Byte 1

  Bits 0–3 – Physical Unit Number (one hexadecimal digit)

  - 0–n (numbered sequentially for each device type unit)

  Bits 4–6 – Allocation Code

  - 000 – Unallocated
  - 110 – Allocated to Main Program
  - 111 – Allocated to System

  Bit 7 – Up or Down Code

  - 0 – Up
  - 1 – Down

- Byte 2

  Bits 0–4 – DDMMM – Tape Density and Mode (Represented in function specifications for magnetic tape units. For 9-channel tape units, DDMMM is set to 10000 when the supervisor is generated.)

  Bits 5–7 – Always set to 011

  Bits 0–7 – For communications devices, the input channel address for the device (see Appendix A).

- Byte 3

  Bits 0–7 – Device address of the unit as determined by hardware specifications. For communications devices, this is the address of the output channel.

If the routine addressing the PU table is being assembled or linked with it, the table can be addressed by its label E?FT. The last entry in the PU table is followed by a halfword sentinel of all 1 bits.

4.6. LOGICAL UNIT TABLE

The tape and peripheral facilities of a configuration are addressed by a fixed set of numbers called logical unit numbers, which take the form nn, where nn is 00, 01, 02, ... for as many logical unit numbers as needed up to a maximum of 3F (hexadecimal). Each logical unit number is associated with an entry in the logical unit (LU) table. The LU table is used to associate a file with the hardware unit on which the file is located. The format of the table is shown in Figure 4–2.

LOGICAL UNIT NUMBER (HEXADECIMAL CODE)	0								1							
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
00	Swap Indicator	Alternate LU Table Entry							PU Table Pointer							
01																
02																
.																
.																
.																
3F																

Figure 4–2. Logical Unit Table Format

As may be seen, each entry consists of three fields: swap indicator, alternate LU table entry, and PU table pointer.

- Swap Indicator – Initially, the value of the swap indicator is zero. When the first swap is made, the value is changed to one. With the second swap, the value reverts to zero. On the third swap, the value is again one, and so on. Thus, the value of the swap indicator is zero when the relation between the logical and physical units is in the original state; the value is one when the relation is in the swap state.

- Alternate LU Table Entry – If a logical unit is assigned to two physical tape units to provide for swapping, the value in the alternate LU table entry field points to the LU table entry assigned to the alternate physical tape unit. Thus, when an end-of-volume is detected and a tape handler swap occurs, the PU table pointers associated with the two LU table entries linked by the value in the alternate LU table field are swapped.

  If an alternate entry is not required, the alternate entry address points to its own logical unit entry. In this way, the same tape swap coding routine can be used whether or not an alternate is provided.

  For units other than tapes, the alternate entry field contains binary 0's.

  An alternate LU table entry address may be calculated by adding the base address of the LU table to the value appearing in the alternate LU table entry field.

- PU Table Pointer – The value in this field points to the entry in the PU table which describes the unit assigned to the logical unit number associated with the LU table entry. A PU table entry address may be calculated by adding the base address of the PU table to the value appearing in the PU table entry field.

No two LU table entries may point to the same PU table entry.

The hexadecimal value of a logical unit number is supplied in the request packet by the requesting program. The dispatcher doubles the number and adds it to the base address of the LU table to locate the proper entry in the table.

For a UNIVAC 9200 II/9300/9300 II configuration with n UNISERVO VI C magnetic tape units, the first n entries in the LU table must be used to address the tape units. The last entry in the table is followed by a halfword sentinel of all 1 bits.

4.7. KEYIN TABLE

The keyin table has 16 entries, each two bytes long. Each entry contains the address of a routine to process the unsolicited keyin associated with the entry.

When an unsolicited keyin is made, the first four bits inserted in the DATA ENTRY switches indicate the association with the proper keyin table entry. If the keyin is to be ignored, the associated entry contains a binary zero. If the entry contains other than binary zero, the supervisor branches to the address in the entry. The associated routine receives control in I/O mode. Therefore, it should return control to the supervisor at reentry when the processing is completed.

In the minimum operating system, when the supervisor is loaded, the first 15 keyin table entries contain binary zeros. The last entry contains the entry address for a routine to process an unsolicited keyin request for external cancellation.

The first entry of the keyin table is assigned to the main program. If a main program is to receive unsolicited keyins, it places the address of the processing routine in its keyin table entry. The main program can suppress its response to unsolicited keyins by resetting this entry to binary 0. When the job control program is loaded, it clears the main program keyin table entry to binary 0.

4.7.1. Unsolicited Keyins

The operator makes an unsolicited keyin by pressing the OP REQ (operator request) switch on the console. This causes the byte set up in the DATA ENTRY switches to be stored in memory location 5, and also causes an interrupt for which the supervisor looks before using the device address byte channel number to enter the interrupt table. When the supervisor detects an unsolicited keyin, it interrogates the first four bits in memory location 5 which contain the code for the action the supervisor is requested to take. The last four bits are used to encode a message when the action requested is an unsolicited keyin to a main program.

In the minimum operating system, there are two supervisor actions which can be requested by means of an unsolicited keyin. The responses of the supervisor to these unsolicited keyins are given in the following paragraphs.

4.7.1.1. Keyin to Main Program (Action Code 0000)

When an unsolicited keyin is made to a main program, the supervisor inspects the main program keyin table entry. If the entry contains a binary zero, the supervisor ignores the keyin. If the entry contains an address, the supervisor branches to that address. Control is thus turned over to the main program.

When the main program receives control because of an unsolicited keyin, the processor is in I/O mode. The main program can inspect the keyin which is stored in memory location 5, take appropriate action, and return control to the supervisor at reentry. This keyin acceptance routine should be written to be executed in I/O mode.

4.7.1.2. External Cancellation (Action Code 1111)

The supervisor responds to an external cancellation keyin in the same way it responds to the execution of a CANCL macro instruction: it halts on a CANCL display.

4.8. EXECUTION ACTIVITY SUM

The execution activity sum is a counter located in the address table within the supervisor and is maintained by the dispatchers to keep a running total of I/O activity. It is incremented by a given value for each execution of an XIOF and decremented when is completed. The function of the counter is to prevent memory overload from occurring due to excessive I/O activity. Each dispatcher compares the execution activity sum against a predetermined limit before issuing any orders. If the counter exceeds the limit, the dispatcher creates a temporary loop by adjusting the processor PSC and returning control to the problem program at the point of its request. The execution activity sum has the label E?AS and is located in memory locations 264 and 265.

4.9. ADDRESS TABLE

A set of entries making up the address table is maintained in a fixed area of low order memory. Most of the entries in the table are addresses of different parts of the supervisor that problem programs may need to reference. However, some entries, such as the execution activity sum, are the value itself rather than the address of the value.

Two entries, the UNIVAC 8410 disc units and the UNISERVO VI C magnetic tape units, are values that are equal to two times the number of the channel on which the device is located. The location and contents of the entries are listed in Table 4–3.

LOCATION (decimal)	CONTENTS
260–261	PU TABLE BASE ADDRESS
262–263	LU TABLE BASE ADDRESS
264–265	EXEC ACTIVITY SUM
266–267	DISPLAY SUBROUTINE ADDRESS
268–269	BOUNDARY TABLE BASE ADDRESS
270–271	INTERRUPT TABLE BASE ADDRESS
272–273	SRC TABLE BASE ADDRESS
274–275	REENTRY ROUTINE ADDRESS
276–277	KEYIN TABLE BASE ADDRESS
278	VERSION NUMBER
279	UPSI BYTE
280–285	DATE
289	8410/8411/8414 DISC UNITS CHANNEL NUMBER
291	MAGNETIC TAPE UNITS CHANNEL NUMBER
292	TOP-OF-MEMORY ADDRESS

Table 4–3. Address Table

The version number is a value representing the number of the version of the supervisor in memory.

The UPSI byte is a value and is set to all binary zeros when the supervisor is generated.

The date is a value and is determined by the DATE control card.

4.10. MAGNETIC TAPE DISPATCHER

The tape dispatcher subroutine provides the interface between tape file control routines and the individual UNISERVO magnetic tape units themselves. It performs the functions necessary to execute tape I/O requests, to handle the resulting interrupts, and to perform error recovery.

4.10.1. Request Procedure

A tape input/output request is forwarded to the tape dispatcher in the form of the address which points to a tape request packet located within the problem program. This packet contains the information necessary to create the desired function, and also provides a working storage area which contains status information and a chaining address. The tape dispatcher links this request to the tape request chain and, when possible, executes the request at the top of the chain. By testing the status indicator within the request packet, the problem program can determine when and if the request has been completed.

A priority indicator may be included within the request packet which causes the tape dispatcher to give a priority to the request in the chain.

The tape request packet format is shown in Figure 4–3.

4.10.2. Program Procedure

The problem program submits a tape request by storing the address of the packet into register 15 and executing and SRC instruction as follows:

LABEL	OPERATION	OPERAND
	SRC	0,c

where c has a value that is twice the number of the channel on which the magnetic tape control unit is located. Once submitted, a packet should not be resubmitted until it has been serviced by the tape dispatcher.

When, after receiving a request, the tape dispatcher finds the chain for that channel empty, or through an interrupt determines that XIOF may be executed, it examines the request at the top of the chain. It then creates the tape order by using the logical unit number supplied in the request packet to reference the LU table which, in turn, points to the physical unit number assigned to the file in the PU table.

4.10.3. Error Recovery

There are three types of errors which may be encountered:

- Type A – XIOF is to be repeated.
- Type B – The tape must be repositioned before repeating the XIOF.
- Type C – Immediate operator intervention is required.

Automatic recovery of types A and B is attempted five times before a stop display is required. The last two times an attempt is made to recover from a type B read error, the attempts are made with low gain. An automatic error recovery attempt for a type B write error is to backspace a block, erase, and then reissue the write. Since the tape dispatcher is in I/O mode during this time, a direct access to the supervisor's display subroutine is required, rather than an execution of the MSG macro instruction. Memory location 4 contains a zero if a retry is desired; otherwise, the packet status indicator is set to 0010 (Error) and the order is skipped.

4.10.4. Mode Set

The tape dispatcher must have the ability to recognize whether or not the current XIOF requires the identical mode as the previously issued XIOF for this control unit. If it does not, the tape dispatcher must perform a "mode set" function in order to reset the control unit to the proper mode for the current XIOF. Mode need not be altered for 9-track tape units, because it is ignored for 9-track functions.

Although there is nothing built into the tape dispatcher to prevent it, the tape dispatcher does not anticipate that the problem program will request a mode set.

4.10.5. Tape Request Packet

The tape request packet contains 12 bytes of information and must be positioned at a halfword boundary. The address of the first byte is forwarded to the tape dispatcher for execution of the request. The format of the tape request packet is shown in Figure 4–3 and described in the following paragraphs.

0	1	2	3	4	5
Buffer Control Word Specification				Density and Mode	Device Address or Physical Unit Number

6	7	8	9	10	11
Counts	Function Specification	Indicators	Logical Unit Number	Chain or Last Byte Address	

Figure 4–3. Tape Request Packet Format

Bytes 4 and 5 and bits 4 through 7 of bytes 6 and 8 are set by the tape dispatcher when the packet is submitted. The dispatcher places the appropriate data in bytes 10 and 11 when the tape operation is completed.

- Bytes 0–3 — Buffer Control Word Specification – These bytes contain buffer control information which includes memory base address, byte count, and control data in the form required to load the buffer control word.
- Byte 4 — Density and Mode – When the packet is submitted, the tape dispatcher inserts in this type the tape density and mode information from the PU table entry indicated by the logical unit number found in byte 9.
- Byte 5 — Device Address/Physical Unit Number – When the packet is submitted, the tape dispatcher inserts in this type the device address from the PU table entry indicated by the logical unit number found in byte 9.

When the packet is returned to the problem program, the tape dispatcher places the physical unit number from the PU table entry into the four most significant bits of byte 5 and sets the other four bits of the byte to binary zeros.

- Byte 6 — Error Count and Noise Block Count –

  Bits 0–3 – This field is used by the tape dispatcher for error recovery procedures.

  Bits 4–7 – When a packet is submitted to the tape dispatcher, this field is set to binary zeros. The tape dispatcher then maintains in this field a binary count of how many noise blocks are skipped during the time this packet is being serviced.

- Byte 7 — Function Specification – This byte contains the function as required in the least significant byte of the XIOF instruction.

- Byte 8 — Indicators –

  Bits 0–3 – Priority Indicator

  0000 Normal priority
  0001 Top priority

  Bits 4–7 – Status Indicator

  0000 Not serviced
  0010 In error, skipped
  0011 Unit exception
  0100 Rewinding
  1000 Wrong length record
  1111 Completed

- Byte 9 — Logical Unit Number

- Bytes 10–11 — Chain or Last Byte Address – These bytes are used by the tape dispatcher to chain requests. When a request has been serviced, this contains an indication of the address of the last byte involved in the tape operation.

When a top priority request is encountered, the tape dispatcher inserts the tape request packet in the queue above all normal priority request packets in the queue waiting to be initiated, but below any other top priority request packets. Regardless of what priority a packet has when it is submitted to the tape dispatcher, the priority indicator is set to 0000 (normal) when it is returned from the tape dispatcher.

The tape dispatcher ensures that the status indicator is preset to 0000 (not serviced) before initiating an XIOF. After interpreting the interrupt, the dispatcher sets the proper status indication for interrogation by the problem program.

The tape dispatcher adjusts the status indicator within the request packet to a value of 0011 (unit exception) to indicate that a tape mark was encountered during the execution of a read or space block, or a write or erase was executed in the end-of-tape area.

The status of 0100 (rewinding) indicates that the tape on the unit addressed in the packet is in the process of being rewound. Consequently, the request was not serviced and should be resubmitted.

When a magnetic tape read has been successfully completed, the tape dispatcher checks to ascertain that the byte count in the buffer control word has been reduced to zero. If it has, the status of the tape request packet is set to 1111 (completed). If it has not, the status is set to 1000 (wrong length record).

When a tape operation is completed, the tape dispatcher transfers the data address, resulting from the operation from the last two bytes of the tape buffer control word to the chain address area of the request packet associated with the operation.

4.10.6. Channels

During generation of the supervisor, the tape dispatcher requires a parameter naming a channel on which the tape units are to be addressed.

4.10.7. Noise Blocks

If a block of 11 bytes or less is read without setting the noise bit in the sense byte information, the tape dispatcher ignores it as a noise block. A minimum block size 18 frames is recommended for blocks that are to be written. Even though a tape mark is only one frame in length, it is not a data block and, consequently, is never considered a noise block.

4.10.8. Tape Unit Status

A problem program can obtain the status of a tape unit by submitting a request with a NOP function specification (a mode set with a DDMMM field of 00000). The tape dispatcher accepts, queues, and initiates such a request packet normally, but gives it special treatment at interrupt time. At that time the sense information is obtained, and sense bytes 0 and 1 are placed in bytes 10 and 11 of the request packet. The problem program can then check the request packet for the status of the tape unit.

4.11. 8410 Disc Dispatcher

The 8410 disc dispatcher is an integral part of the supervisor. It provides the interface between the problem program and the individual UNIVAC 8410 disc unit within the MOS. When called upon by the problem program, the 8410 disc dispatcher performs the functions necessary to execute all disc input/output commands and to handle the resulting interrupts and error conditions.

4.11.1. Program Requirements

The 8410 disc dispatcher is entered when the problem program issues an SRC instruction which gives control to the supervisor. The immediate operand of the SRC instruction contains a value that is twice the number of the channel on which the disc unit is located.

The format of the SRC instruction is:

LABEL	♭ OPERATION ♭	OPERAND
	SRC	0,c

where: c is the value that is twice the number of the channel on which the disc controller is located. This value is stored in location 289_{10}, within the address table.

The supervisor adds this value, c, to the base address of the SRC table to locate within the table the address of the 8410 disc dispatcher routine to which the supervisor branches unconditionally.

Before the problem program issues the SRC instruction, it must first load register 15 with the address of a ten-byte request packet which contains a status indicator byte, a chain indicator, a chain address if I/O commands are to be chained together, and the information required to create the desired I/O function. The problem program may then issue the SRC instruction.

NOTE: Once an SRC instruction has been issued for a particular request packet, the problem program must not alter the contents of that packet.

4.11.1.1. Disc Request Packet

The disc request packet contains ten bytes of information and must be positioned at a halfword boundary. The format of the request packet is shown in Figure 4–4.

0	1	2	3	4
BUFFER CONTROL WORD SPECIFICATION				CHAIN INDICATOR

5	6	7	8	9
FUNCTION SPEC.	STATUS INDICATOR	LOGICAL UNIT NUMBER	CHAIN ADDRESS	

Figure 4–4. Disc Request Packet Format

A description of the contents of each of the ten bytes follows.

4.11.1.1.1. Buffer Control Word (BCW) Specification (Bytes 0–3)

These bytes contain the buffer control information required to load the buffer control word. The format of the BCW is:

0								1								2								3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
W	0	0	BYTE COUNT													0	DATA ADDRESS														

- Byte 0

  Bit 0 – W – Data Direction Bit
  0 = Read (Input) operations
  1 = Write (Output) or buffered control operations

  Bits 1,2 – Always 0

- Byte 0 and 1

  Bits 3–7, Bits 0–7 – Byte Count – This is a binary number specifying the number of bytes that are to be accessed. This count must also include the first five bytes of the data area.

- Byte 2

  Bit 0 – Always 0

- Byte 2 and 3

  Bits 1–7, Bits 0–7 – Data Address – These bytes contain the address of the data area. The length of the data area is specified in the byte count field. A more detailed explanation of the format of the data address is given in 4.11.1.2.

4.11.1.1.2. Chain Indicator (Byte 4)

This byte is used by both the problem program and the 8410 disc dispatcher to indicate the chaining of request packets.

Bit 0 – This bit is used only by the 8410 disc dispatcher. The problem program, however, initially sets the bit to 0. The 8410 disc dispatcher uses this bit to determine if the request packet is already a member of a chain of packets being processed.

Bits 1–6 – These bits are not used by the 8410 disc dispatcher; therefore, they are available for use by the problem program.

Bit 7 – This bit is used by the problem program to indicate to the 8410 disc dispatcher whether or not another request packet is chained to this packet. The problem program sets this bit equal to 1 and loads the address of the next packet in the chain into bytes 8 and 9 of this packet. There is no limit to the number of packets the problem program may chain together. The 8410 disc dispatcher does not change the setting of this bit.

4.11.1.1.3. Function Specification (Byte 5)

This byte specifies the disc I/O command to be performed, as required in the least significant byte of an XIOF instruction.

COMMAND	HEXADECIMAL SPECIFICATION
Write	11
Write and Check	51
Read	61
Seek Track	01
Magnitude Search	71
Search Equal	31

4.11.1.1.4. Status Indicator (Byte 6)

The 8410 disc dispatcher sets this byte and the problem program interrogates it to determine the status of the disc I/O command. When a request packet is submitted to the 8410 disc dispatcher, the disc dispatcher sets the status indicator to "not serviced." After the request is serviced, the 8410 disc dispatcher sets the status indicator byte to the proper value. The possible settings of the status indicator are:

BIT CONFIGURATION	HEXADECIMAL NOTATION	MEANING
00000000	00	Not serviced.
00010000	10	Unrecoverable abnormal line.
00100000	20	Unrecoverable output bus check.
01000000	40	No find.
01010000	50	Major failure.
01110000	70	Nonoperational channel.
10000000	80	Invalid function.
11110000	F0	The unload buffer command generated by the disc dispatcher was not completed successfully.
11111111	FF	Operation was successfully completed.

When the disc I/O command requires an unload buffer command to move the data from the disc buffer to memory, the status of the request packet remains not serviced until the unload buffer command has been executed.

4.11.1.1.5. Logical Unit Number (Byte 7)

The problem program stores the logical unit number of the disc to be accessed in this byte. The number is expressed in hexadecimal.

4.11.1.1.6. Chain Address (Bytes 8 and 9)

Both the problem program and the 8410 disc dispatcher use these bytes to store the address of the next packet in the chain. If bit 7, byte 4, of the current request packet indicates that the problem program has another packet chained to this one, the address of the next packet in the chain is stored in bytes 8 and 9. If the problem program has not chained its request packets, the 8410 disc dispatcher establishes its own chain by storing the address of the next packet issued by the problem program in bytes 8 and 9.

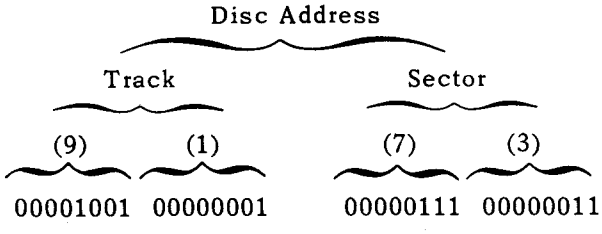
4.11.1.2. Address of Data Area

As mentioned in 4.11.1.1.1, the least significant 15 bits of the BCW contain the address of the data area. The first five bytes of the data area must contain the disc address of the sector to be accessed. The disc address has the following format:

0	1	2	3	4
U	T_1	T_2	S_1	S_2

where: U is the unit address of the disc.
T_1 is the most significant digit of the track address.
T_2 is the least significant digit of the track address.
S_1 is the most significant digit of the sector address.
S_2 is the least significant digit of the sector address.

The 8410 disc dispatcher sets the unit address from the logical unit number specified in the request packet. The problem program must store the disc address of the track and sector to be accessed into bytes 1 through 4 of the data area. The two-digit track number ranges from 00 to 99. The two-digit sector number ranges from 00 to 54 for the outer sectors and from 55 to 99 for the inner sectors. For example, if the disc address to be accessed is track 91, sector 73, the bit configuration of bytes 1 through 4 would be:



The length of the data area, as specified in the byte count field of the BCW, must include the first five bytes containing the data address.

When a disc I/O command has been completed, the 8410 disc dispatcher replaces the unit address in byte 0 with the physical unit number of the disc. The number is stored left-justified in bits 0 through 3 of the byte; bits 4 through 7 contain 0's. It is this physical unit number that is used for display purposes.

4.11.2. Operating Considerations

The 8410 disc dispatcher receives control:

- when the problem program issues an SRC instruction, which causes an interrupt;
- when a disc I/O command generates an interrupt.

A description follows of the operation of the 8410 disc dispatcher for the two types of interrupts, particularly with regard to the setting and interrogating of the chain indicator bits.

4.11.2.1. SRC Interrupt

After the problem program has placed the required information in the request packet and has issued an SRC instruction, an interrupt is generated which gives control to the 8410 disc dispatcher through the supervisor.

The 8410 disc dispatcher checks the internal chain indicator, bit 0, byte 4, of the request packet, to determine if this packet is a member of an existing chain of packets being processed. If this is the case, the 8410 disc dispatcher displays '67ul' (u=physical unit number of the disc) to indicate that a request packet has been resubmitted before initial processing. When the operator presses the START switch with no keyin, the 8410 disc dispatcher processes the packet as resubmitted by the problem program. This action may result in a number of disc I/O commands never being initiated, because the existing chain of packets is broken and their status remains not serviced. This is a programming error and the operator should make an unsolicited external cancellation keyin (see 4.7.1.2) to terminate the job in process.

If the chain indicator, bit 0, byte 4, indicates that this is a new request and the 8410 disc dispatcher accepts the packet for processing, the status of the packet is set to not serviced and the chain indicator bit is set to 1.

The 8410 disc dispatcher then checks the external chain indicator, bit 7, byte 4, to determine if other packets are chained to this one. If the problem program has chained this packet, the 8410 disc dispatcher checks the external chain indicator bit in the other packet(s) in the chain. At the same time, it checks the internal chain indicator, bit 0, byte 4, to make certain that no packet is being resubmitted. If the 8410 disc dispatcher has already established a chain, it adds this new chain of packets from the problem program to the existing chain and sets bit 0, byte 4, to 1 for all the packets in the chain. Processing of the packet is initiated, if possible, and the 8410 disc dispatcher returns control to the problem program.

4.11.2.2. Disc I/O Command Interrupt

When an interrupt occurs at the successful completion of a read, magnitude search, or search equal command, the 8410 disc dispatcher automatically generates an unload buffer command as the next disc I/O command to be executed. The number of bytes to be unloaded from the disc buffer to memory is obtained from the byte count in the request packet for the initial command.

The status of the packet remains not serviced until the buffer is unloaded. At this point, the status byte is set to indicate that the operation is completed or that an error condition occurred during the operation of either the initial command or the unload buffer command. The chain indicator, bit 0, byte 4, is then set to 0 to indicate that this packet is no longer considered a member of the chain of packets being processed by the disc dispatcher.

If a magnitude search command is executed with a byte count greater than five, the search key is destroyed when the unload buffer command is executed.

If an output bus check or an abnormal line error occurs during the operation of the initial command, the proper error indication bit is not set in the status byte until after the unload buffer command is completed. If an error other than an output bus check or an abnormal line occurs, the unload buffer command is not generated; the error indication is set, and the 8410 disc dispatcher attempts to initiate the next disc I/O command in the chain of packets. If there are no other packets to be serviced, the 8410 disc dispatcher returns control to the problem program.

When the initial command does not require an unload buffer command upon its completion, the status or error indication of the result is stored in the status byte of the request packet; the chain indicator, bit 0, byte 4, is set to 0 to indicate that this packet is no longer considered a member of the chain of packets being processed. The 8410 disc dispatcher then attempts to initiate the next disc I/O command in the chain of packets. If there are no other packets to be serviced, the 8410 disc dispatcher returns control to the problem program.

4.11.3. Error Recovery

For certain types of errors, the 8410 disc dispatcher attempts to execute the I/O command three times before it considers the error unrecoverable. Other types of errors are considered unrecoverable when they occur and no attempt is made to retry them.

4.11.3.1. Output Bus Checks and Abnormal Line Errors

The 8410 disc dispatcher retries the I/O command if an error occurs on an XIOF instruction because of an output bus check or because the abnormal line is active. If the error condition exists after the third attempt, the error is considered unrecoverable. The 8410 disc dispatcher sets the status byte to indicate the type of error, resets the chain indicator, bit 0, byte 4, to 0 and attempts to execute the I/O command of the next request packet. Control is then returned to the problem program.

4.11.3.2. Other Errors

For other errors, the 8410 disc dispatcher does not retry the operation. The status byte is set to indicate the type of error, the chain indicator bit is reset to 0, and the I/O command of the next packet is initiated. Control is then returned to the problem program.

On a seek command, the status is returned when arm motion begins. If a malfunction occurs after the arm is in motion, the error indication is not set in the status byte until the next I/O command is initiated. For search commands, the normal procedure is to repeat the operation. This repetition could result in an error indication of no find. Therefore, there is the possibility that a no find may indicate a hardware malfunction rather than a programming error.

4.11.3.3. Nonoperational Channel Error

When a nonoperational channel error occurs, the 8410 disc dispatcher displays '67uF', where u is the physical unit number of the disc drive. The operator may either skip the operation by keying a nonzero into location 4 and pressing the START switch or retry the operation by pressing the START switch without making a keyin.

If the operation is to be skipped, the 8410 disc dispatcher sets the status byte to indicate a nonoperational channel, resets the chain indicator, bit 0, byte 4, to 0, and attempts to execute the I/O command of the next packet. Control is then returned to the problem program.

4.12. 8411/8414 DISC DISPATCHER

The 8411/8414 disc dispatcher subroutine provides the interface between the disc file control routines and the individual disc units.

4.12.1. Request Procedure

An 8411/8414 disc input/output request is forwarded to the 8411/8414 disc dispatcher in the form of an address which points to a disc request packet located within the problem program. This packet contains the information necessary to create the desired function(s), and also provides a working storage area which contains status information and a chaining address. The 8411/8414 disc dispatcher links this request to the disc request chain and, when possible, executes the request at the top of the chain. By testing the status indicator within the request packet the problem program can determine when and if the request has been completed.

4.12.2. Program Procedure

The problem program submits a disc dispatcher request by storing the address of the packet into register 15 and executing an SRC instruction as follows:

LABEL	OPERATION	OPERAND
	SRC	0,42

Once submitted, a packet should not be resubmitted until it has been serviced by the 8411/8414 disc dispatcher.

4.12.3. Program Commands

The UNIVAC 8411/8414 disc subsystem accepts command directives in the form of an address (command address word) which points to a command control chain. The command control chain may consist of one or many eight (8) byte command control words which are located in contiguous memory locations. One requirement of the system is that command address words must be set to double word boundaries. This can be accomplished by the use of the following assembler directive preceding each command control word chain in a problem program:

LABEL	♭ OPERATION ♭	OPERAND
	ORG	*,8

The START directive to the assembler should also be set to a double word boundary. When linking a relocatable module containing command control chains to other relocatable modules, the following linker directive should immediately precede the relocatable command control module:

LABEL	♭ OPERATION ♭	OPERAND
	MOD	8

4.12.4. Overlapping Capabilities

Because of the high data transfer rate of the 8411/8414, the central processor unit and other UNIVAC 8411/8414 disc units are locked out by the selector channel during command and data transfers to and from the disc units. The multiplexer channel is capable of operating simultaneously with the selector channel. Overlapping of disc units may be accomplished by the employment of a preselection technique of records in a file. If the cylinder is known for record n + 1, on disc volume 1 the command chain for record n would include as its last command a seek to record n + 1 on volume 1. The next command chain could then obtain record n on disc volume 2 using a seek as its last command to position to record n + 1 on disc volume 2. In this manner, seek command may be overlapped on the two disc volumes.

4.12.5. Error Recovery Procedures

The 8411/8414 disc dispatcher provides for automatic error recovery by reissuing the entire command chain. However, some command chains may have logic which precludes the reissuing of the entire command chain. In these cases, the user has the ability to inhibit automatic recovery. Information is returned to the user which will enable him to determine the point of abnormal termination of the chain. A partial chain may be formulated and issued to the dispatcher to complete the command chain logic.

Positioning to and from defective tracks is automatic.

4.12.6. Disc Request Packet

The disc request packet contains twelve bytes of information and must be positioned at a halfword boundary. The address of the first byte is forwarded to the 8411/8414 disc dispatcher for execution of the request. The format of the 8411/8414 disc dispatcher is:

- Bytes 0–1 — Command Address Word. These bytes contain the address of the first command control word in the command chain.
- Bytes 2–3 — Last Command Address Word. These bytes are supplied by the 8411/8414 disc dispatcher and contain the address of the last command and word executed plus eight (8).
- Byte 4 — Physical Unit. This byte is supplied by the 8411/8414 disc dispatcher and contains the physical unit number of the device addressed. It is extracted from the physical unit table.
- Byte 5 — Cylinder Address. This byte contains the cylinder address of the automatic seek if requested by the user.
- Byte 6 — Count of Automatic Reissue. This byte is supplied by the dispatcher and contains a binary count of the number of times automatic error recovery procedures were attempted.
- Byte 7 — Head Address. This byte contains the head address of the automatic seek if requested by the user.
- Byte 8 — Indicators –

  Bit 0 — A one (1) indicates no automatic error recovery and control is returned to the user with the appropriate status.

  Bits 1–2 — Operation mode

  where:

  – 00_2 indicates the dispatcher is to initiate an automatic seek to the cylinder and head addresses specified in bytes 5 and 7 of the request packet and transfer command control to the user chain.

  – 01_2 indicates an automatic seek is to be performed only.

  – 10 indicates execution of the user command chain only.

  Bit 3 — Priority indicator. Binary 0 indicates normal priority and binary 1 indicates top priority.

NOTE: Bits 0–3 of byte 8 will be set to binary zeros when the packet is returned to the user.

Bits 4–7 Status Indicator

0000 not serviced
0001 software error
0010 hardware error
0011 end of file
0100 no find on search
1000 wrong length error
1111 successful completion

- Byte 9 Logical Unit Number
- Bytes 10–11 Chain Address. These bytes are used by the dispatcher to chain requests. When the packet has been marked serviced, the last data address will be set.

4.12.7. 8411/8414 Disc Dispatcher Generator

The procedures required for generating the 8411/8414 disc dispatcher are described in Section 2.

4.13. CARD LOADER

Program loading is a function of the card load routine which must be incorporated with each problem program. The problem program thus can be loaded by means of the card read unit. The card load routine is described in *UNIVAC 9200/9200 II/ 9300/9300 II Systems Card Assembler Programmers Reference, UP-4092* (current version).

There must be a guarantee that loading a problem program does not destroy the supervisor. This guarantee is obtained by setting the base of the problem program and the addresses assigned to the labels L?AR and L?PG in the card load routine above the upper boundary of the supervisor.

5. JOB CONTROL PROGRAM

5.1. GENERAL

The job control program is that component of the minimum operating system that is used to process the statements of a control card stream and to perform the various cleanup and housekeeping duties required to prepare the minimum operating system for the execution of each problem program.

To perform its function, the job control program must be loaded into main storage prior to the execution of a problem program (job) or between the execution of any two jobs where the functions of the job control program are required. The loading of the job control program may take place after the supervisor has been loaded. Since the job control program is contained on a self-loading card deck, it is not necessary to employ the use of the card loader routine.

5.2. FUNCTIONS OF THE JOB CONTROL PROGRAM

When loaded into main storage, the job control program removes the main program entries from the PU and keyin tables of the supervisor. It also restores the alternate pointers and swap indicators specified in the LU table of the supervisor to their initial state. If a swap indicator is found on, the relation between the logic and physical units is returned to the original state before the alternate pointer and swap indicators are restored. After these housekeeping functions have been performed, the job control program will organize the system to the requirements of the next job to be executed. The requirements for each job are contained on the control card which are read as part of the control stream by the job control program.

5.3. CONTROL STREAM

The control stream is the fundamental input to the UNIVAC 9300 minimum operating system. It is comprised of both the controls to which the system responds and which govern data processing and the data which is to be processed. Both of these inputs are introduced into the minimum operating system by means of the card reader unit. The advantage of this approach is that the user can stimulate the system to perform a series of operations in an automatic sequence by preparing one properly organized card deck. Since the job control program responds only to the control card portion of the control stream, this section shall be limited to a discussion of control cards and the response of the job control program to these cards.

5.4. CONTROL CARDS

Control cards are read from the card read unit by the job control program. Control cards are identified by a slash (/) in column 1 and an operation code left-justified in columns 10–14, and they may contain a series of positional parameters beginning in column 16. Cards without a slash in column 1 are considered data cards and are processed as such. The control cards processed by the job control program are listed with a brief description of their function in Table 5–1. The DVCDN, DVCUP, JOB, ASSGN, SWAP, ALT, FILES, and MTC control cards are processed in the order listed and therefore must also appear in this sequence with the control card deck. The UPSI, DATE, and PAUSE control cards can appear any place in the control card deck and are processed when read. The last control card in the control card deck must be the FINIS card; the other control cards are optional. The JOB and FINIS control cards may only appear once in a control card deck; the other cards may appear more than once. The job control program enforces the control card sequencing as described in this discussion.

A description of the response of the job control program is given for each control card.

CONTROL CARD	DESCRIPTION
DVCDN	Specifies to the minimum operating system that device is down and not available for use.
DVCUP	Specifies to the minimum operating system that device is available for use.
JOB	Initializes the minimum operating system for each job to be processed.
ASSGN	Used to set tape density and mode for 7-channel tape units.
SWAP	Used to interchange logic address of two tape or disc units.
ALT	Specifies alternate tape or disc unit for system use.
FILES	Used to preposition tape to the beginning of a named file in a multifile tape reel.
MTC	Used for tape manipulation.
UPSI	Used to specify desired bit configuration for the UPSI byte in address table.
DATE	Used for supplying current date to minimum operating system for checking and writing dates in labels.
PAUSE	Used to halt or temporarily suspend control stream operation.
FINIS	Used to indicate to the minimum operating system that the control stream has ended.

Table 5–1. Control Cards for Minimum Operating System

5.4.1. DVCDN

The DVCDN (device down) control card is used to specify that a device is down or offline and is unavailable for I/O operations.

The format for the DVCDN control card is:

LABEL	↹ OPERATION ↹	OPERAND
/	DVCDN	{ CRP | CRD | PRINT | TAPE | CC | RRP | DISC | CDVC }, physical-unit-number

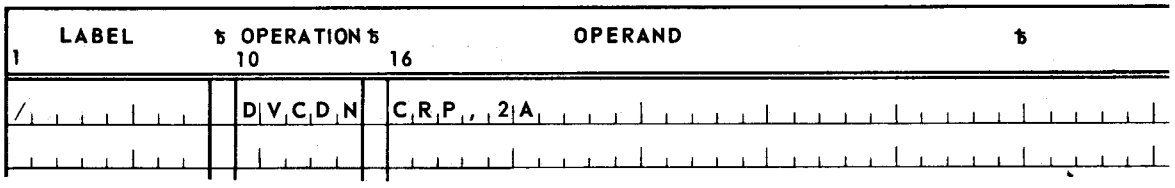
- Positional Parameter 1

  CRP – ID for online serial read/punch
  CRD – ID for card reader
  PRINT – ID for printer
  TAPE – ID for magnetic tape unit
  CC – ID for card controller
  RRP – ID for row read/punch
  DISC – ID for 8410/8411/8414 DAS units
  CDVC – ID for communication devices

- Positional Parameter 2

  physical-unit-number – a two-digit hexadecimal number for the device specified by keyword in positional parameter 1.

Example:



The job control program uses the parameters of the control statement shown to isolate the PU table entry for device – ID (CRP) and to set this entry to down (unavailable for use).

5.4.2. DVCUP

The DVCUP (device up) control card is used to specify that the device is online and available for I/O operation.

The format for the DVCUP control card is:

LABEL	♿ OPERATION ♿	OPERAND
/	DVCUP	{ CRP | CRD | PRINT | TAPE | CC | RRP | DISC | CDVC } , physical-unit-number

- Positional Parameter 1

  CRP – ID for online serial read/punch
  CRD – ID for card reader
  PRINT – ID for printer
  TAPE – ID for magnetic tape unit
  CC – ID for card controller
  RRP – ID for row read/punch
  DISC – ID for 8410/8411/8414 DAS units
  CDVC – ID for communication devices

- Positional Parameter 2

  physical-unit-number – a two-digit hexadecimal number for the device specified by a keyword in positional parameter 1.

Example:

LABEL	♿ OPERATION ♿	OPERAND ♿
1	10	16
/	DVCUP	CRP,,2A

The job control program uses the parameters of the DVCUP statement to set the PU table entry for device – ID (CRP) to position up (available for I/O operations).

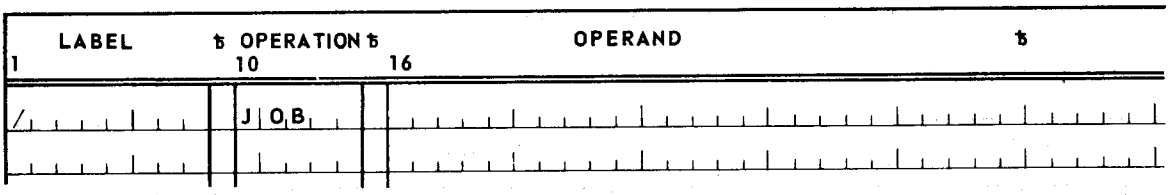
5.4.3. JOB

The JOB control card is used to reinitialize the minimum operating system for the execution of a new program run.

The format of the JOB control card is:

LABEL	♭ OPERATION ♭	OPERAND
/	JOB	unused

Example:



The job control program response to the JOB control card statement is to set to their initial state all LU table entries that do not point to a PU table entry marked down (unavailable for I/O operations). With the exception of the device allocation code and the up or down fields, it also sets all PU table entries back to their initial state. The UPSI byte is set to all binary zeros when the JOB control statement is read.

5.4.4. ASSGN

The ASSGN control card is used to set the tape density and mode fields (byte 2) of the appropriate PU table entry for 7-channel magnetic tape units which are compatible with 9-channel magnetic tape units.

The format for the ASSGN control card is:

LABEL	♭ OPERATION ♭	OPERAND
/	ASSGN	logical-unit-number,ss

- Positional Parameter 1

  logical-unit-number – a two-digit hexadecimal number that specifies the LU table entry for the appropriate tape unit.

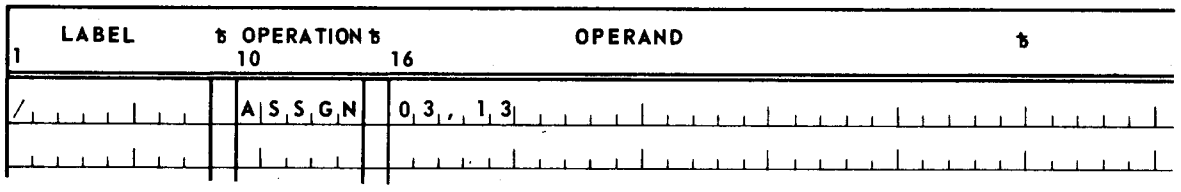
- Positional Parameter 2

  ss – a two-digit hexadecimal value that indicates tape density and mode. Possible values for the ss parameter are:

VALUE	DENSITY (BYTES PER INCH)	PARITY	CONVERSION
13	200	Odd	On
23	200	Even	Off
33	200	Odd	Off
53	556	Odd	On
63	556	Even	Off
73	556	Odd	Off
93	800	Odd	On
A3	800	Even	Off
B3	800	Odd	Off

NOTE: The ASSGN control card can be used whenever the tape dispatcher is included in the minimum operating system.

Example:



The job control program uses the above parameters to isolate the LU table entry specified by logical-unit-number (03). Once this has been accomplished, the job control program selects the PU table entry indicated by the PU table pointer for the LU table entry. It acquires the tape unit status for the tape unit specified by the LU number from the tape dispatcher and verifies that the tape unit is 7-channel. It then sets the density and mode fields of this PU table entry as indicated by the ss value (13) specified in positional parameter 2. The value 13 illustrated in the example indicates odd parity and a tape density equal to 200 bytes per inch of tape.

5.4.5. SWAP

The SWAP control card is used to interchange the logical addresses between two magnetic tape units or between two DAS units.

The format for the SWAP control card is:

LABEL	OPERATION	OPERAND
/	SWAP	logical-unit-number,logical-unit-number

- Positional Parameter 1

  logical-unit-number – a two-digit hexadecimal number contained in the LU table; this parameter can be used to specify tape or disc units.

- Positional Parameter 2

  logical-unit-number – a two-digit hexadecimal number contained in the LU table; this parameter can be used to specify either tape or disc units.

Example:

LABEL	OPERATION	OPERAND
/	SWAP	02,03

The job control program uses the parameters of the SWAP control statement to verify that the units specified (tape or disc) are not allocated. It then swaps the PU table pointers in the LU table for the devices specified by the control statement. The one exception to this process exists when one of the specified devices is not allocated and if the other device is allocated to the system but is in a down state. Under these circumstances, the job control program proceeds to make the swap; however, the unallocated device is marked as down and the other device is allocated to the system.

5.4.6. ALT

The ALT (alternate) control card is used to specify an alternate tape or disc unit for system use.

The format for the ALT control card is:

LABEL	OPERATION	OPERAND
/	ALT	logical-unit-number,logical-unit-number

- Positional Parameter 1

  logical-unit-number – two-digit hexadecimal number representing the logical-unit-number of the tape or disc unit allocated to the system.

- Positional Parameter 2

  logical-unit-number – two-digit hexadecimal number representing the logical-unit-number assigned to the tape or disc unit that is to be allocated to the system as an alternate for the device specified by positional parameter 1 of this control statement.

Example:

LABEL	OPERATION	OPERAND
/	ALT	04,05

The job control program uses the parameters of the control statement to determine which device requires an alternate and the specific device that is to be allocated to the system as the alternate. Prior to allocating the second device to the system, the job control program first verifies that the second device is not allocated and is up (available for use). It then marks the appropriate LU table entry for this device to indicate to the operating system that the device is now allocated to the main program and is the alternate of the first device.

5.4.7. FILES

The FILES control card is used to preposition the tape of a magnetic tape unit to the beginning of a named file in a multifile tape reel.

The format of the FILES control card is:

LABEL	↹ OPERATION ↹	OPERAND
/	FILES	logical-unit-number,file-ID

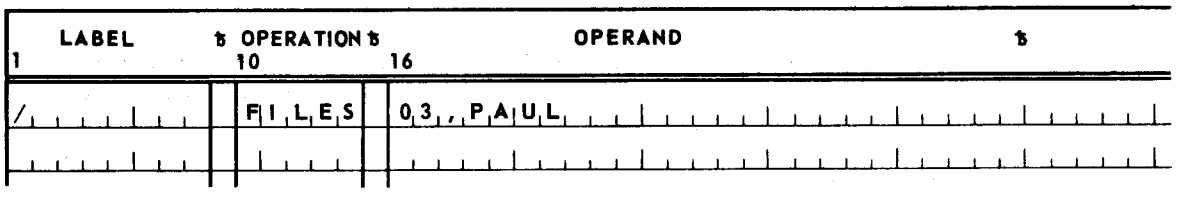
- Positional Parameter 1

  logical-unit-number – two-digit hexadecimal number for the tape unit to be searched for specified file-ID.

- Positional Parameter 2

  file-ID – value which is the identification of the file being searched for.

Example:



The job control program moves (positions) the tape mounted on the specified unit forward up to but not including the header label containing the file-ID specified. Prior to positioning the tape, the job control program verifies that the device specified in the FILES control statement is up (available for use) and is allocated to the system.

5.4.8. MTC

The MTC (manipulate tape) control card is used to manipulate tape position and to write tape marks as specified by the parameters of this control statement.

The format of the MTC control card is as follows:

LABEL	♭ OPERATION ♭	OPERAND
/	MTC	BSF BSR ERG FSF FSR REW RUN WTM ,logical-tape-unit-number[,n]

- Positional Parameter 1

  BSF – opcode for backspace file
  BSR – opcode for backspace record
  ERG – opcode for erase gap
  FSF – opcode for forward space file
  FSR – opcode for forward space record
  REW – opcode for rewind
  RUN – opcode for rewind and unload
  WTM – opcode for write tape mark

- Positional Parameter 2

  logical-tape-unit-number – a two-digit hexadecimal number used to select a specific magnetic tape unit.

- Positional Parameter 3

  n – a decimal digit that indicates the number of times the operation specified by positional parameter 1 is to be performed. If it is not specified, a value of 1 is assumed.

Example:

```
LABEL      ♭ OPERATION ♭      OPERAND                    ♭
1                10          16
/          MTC               BSF,,01,4
```

In the example illustrated, the job control program uses the parameter of the MTC control statement to backspace logical tape unit 01 four files.

5.4.9. UPSI

The UPSI (user program sense indicator) control card is used to set the UPSI byte in the address table with a specified bit configuration during control stream operation. The contents of the UPSI byte may be examined during execution of the problem program.

The format of the UPSI control card is:

LABEL	♭ OPERATION ♭	OPERAND
/	UPSI	[nnnnnnnn]

- Positional Parameter 1

  nnnnnnnn – an eight-bit binary configuration for the UPSI byte. The value of n can be a binary 1, a binary 0, a blank, or any combination thereof.

Example:

```
1    LABEL    ♭ OPERATION ♭    OPERAND              ♭
                10             16
/               UPSI           10011110
```

The job control program uses the parameter of the UPSI control card to modify the binary configuration of the UPSI byte in the address table. The bit configuration specified on the UPSI control card replaces the bit configuration specified by a previous UPSI control card. That is, a binary 1 will cause the corresponding bit of the UPSI byte to be set to a binary 1. It should be noted that any trailing characters that are not specified or any bits omitted within the configuration indicate that the corresponding bit of the original UPSI byte is to remain unchanged.

The UPSI byte is cleared to zero when the job control program encounters a JOB control card.

5.4.10. DATE

The DATE control card is used when it is desired to have the current data available to the problem program for creating and checking dates in tape and disc label blocks.

The format of the DATE control card is:

LABEL	♭ OPERATION ♭	OPERAND
/	DATE	'dddddd'

- Positional Parameter 1

  'dddddd' – a six-digit decimal number which specifies a specific calendar date.

Example:

```
LABEL      ♭ OPERATION ♭      OPERAND                    ♭
1                10              16
/            DATE          '010369'
```

The job control program stores the characters of positional parameter 1 (apostrophes not included) in the data area of the address table.

5.4.11. PAUSE

The PAUSE control card is used to halt or temporarily suspend control stream operations until an operator response takes place.

The format for the PAUSE control card is:

LABEL	♭ OPERATION ♭	OPERAND
/	PAUSE	xxxx

- Positional Parameter 1

  xxxx – a four-digit hexadecimal number.

Example:

```
LABEL      ♭ OPERATION ♭      OPERAND                    ♭
1                10              16
/            PAUSE         8BCD
```

The job control program halts the control stream operation when it encounters the PAUSE control statement and executes an MSG macro instruction to display the least significant 15 bits of positional parameter 1 of the statement. The system remains in a halted condition until the START switch is pressed by the operator at which time control stream operation resumes.

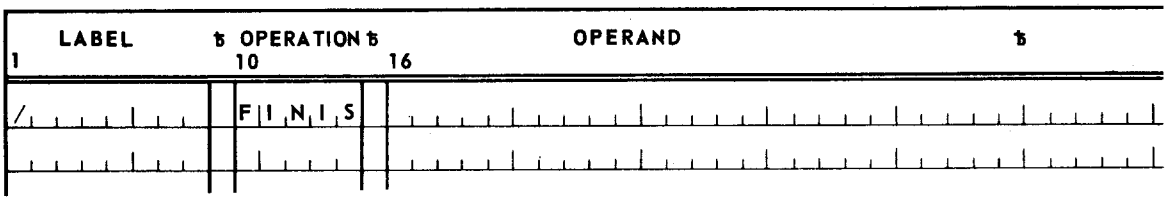
5.4.12. FINIS

The FINIS (finish) control card is used to indicate the end of the control stream.

The format for the FINIS control card is:

LABEL	b OPERATION b	OPERAND
/	FINIS	unused

Example:



The job control program response to the FINIS control card is to execute an EOJ (end of job) macro instruction which indicates to the supervisor that the control stream is completed.

5.4.13. Error Action

There are instances in which a control card is rejected by the job control program. In each case, an appropriate display is made to notify the operator of the error so that the proper corrective action can be taken.

5.5. BASE ADDRESS

The job control program can be linked to reside anywhere in memory above the bounds of the supervisor. Thus, as much low-order program memory area as needed can be set aside, which will remain undisturbed by the loading of the job control program. This undisturbed program memory area can be used to pass information from program to program, even when loading of the job control program intervenes.

APPENDIX A. INTERFACE OF COMMUNICATIONS ROUTINES WITH THE SUPERVISOR

A.1. GENERAL

When communications devices are used by a problem program, the communications I/O control routines interface with the supervisor in the following three areas:

- Issuance of I/O Requests
- Interrupt Processing
- Clocking

These three areas are described in the subsequent sections.

In a minimum operating system with communications handling devices, it is necessary to reload the supervisor if a job is canceled.

A.2. ISSUANCE OF I/O REQUESTS

Input/output requests are normally executed in the I/O mode. If a routine is in the processor mode, the load halfword (LH) and supervisor request call (SRC) instructions are required so that the routine can enter the I/O mode to execute an I/O request.

The formats of the LH and SRC instructions are:

LABEL	OPERATION	OPERAND
unused	LH	15,label
unused	SRC	0,34

The load halfword (LH) instruction transfers into processor register 15 the symbolic label of an address constant which points to the entry into the routine to execute an I/O request in the I/O mode.

The supervisor responds to the resulting SRC interrupt by executing an unconditional branch in the I/O mode to the address (label) in processor register 15. When execution of the request is completed, the communications control routine should branch unconditionally to the reentry routine, the address of which is a fixed location in the supervisor.

This set of instructions may also be used by any program that needs to enter the I/O mode temporarily to inhibit interrupt.

A.3. INTERRUPT PROCESSING

The supervisor arbitrarily assumes that the addresses of the buffer control words (BCW) normally assigned to the communications devices are sequentially associated with channels 16 through 31.

CHANNEL	BCW ADDRESS
16	512
17	516
.	.
.	.
.	.
31	572

This assumption permits a standard interrupt entry calculation and a uniform extension of the interrupt table.

If the communications control routine expects interrupts on a particular channel, it must load the address of the routine processing the interrupts into the interrupt table entry associated with that channel. This must be done before the execution of any XIOF instructions specifying that particular channel. The entry address loaded into the interrupt table for a particular channel is calculated by taking the channel number, doubling it, and adding it to the base address of the interrupt table.

When an interrupt occurs on that channel, or, following a line terminal (LT) summary interrupt, the associated BCW is found to be active, the supervisor loads into a general register the contents of the location whose address was determined by the interrupt entry calculation. The supervisor then branches unconditionally through the general register to the routine processing the interrupt.

The communications control routine should examine bits 4–7 of the BCW for the activity status. The routine must reset all status and buffer bits in the BCW as it determines necessary.

A.3.1. Scan Table

The choice of BCW formats and the channels to which the devices are connected are variable. Therefore, the scan table contains only the addresses of the BCW's that will be tested for activity when an LT summary interrupt occurs. If the BCW is found to be active, the supervisor branches unconditionally to the interrupt routine associated with the active BCW.

The size of the scan table is a supervisor generator option. Each entry in the table consists of four bytes in the following format:

0	1	2	3
BCW ADDRESS		INTERRUPT ROUTINE ADDRESS	

If the communications routine uses the alternate (LT) BCW format, it must enter the address of both the BCW to be scanned and the associated interrupt routine into the scan table. This is accomplished by loading into register 15 the address of the BCW packet. The packet is made up of the BCW address constant and the address of the associated interrupt routine; this is the address that is loaded into the related entry in the interrupt table. The following instruction may then be written:

LABEL	♭ OPERATION ♭	OPERAND
unused	SRC	0,32

The supervisor responds to the resulting interrupt by placing the contents of the packet into the scan table. Initial priorities can be assigned to the entries by the order in which they are placed in the table.

The procedure to remove an entry from the scan table is similar. The address of the BCW packet is loaded into register 15 and the following instruction is written:

LABEL	♭ OPERATION ♭	OPERAND
unused	SRC	0,32

The supervisor responds to the interrupt by matching the BCW address in register 15 with one in the scan table and clearing the entry to binary 0's. It is assumed that by the time a job is completed, all entries in the scan table have been cleared to binary 0's.

The supervisor controls reentry so that a total scan of all stored BCW's is completed before exiting from the original LT summary interrupt.

A.4. CLOCKING

The supervisor maintains a one second delay flip-flop for clocking purposes. This facility is available to any routine requiring interruptions at approximately one second intervals. The user can then maintain within a problem program a one- value of any given interval for such purposes as checking time limits and polling.

A.4.1. Clocking Table

The supervisor maintains an internal clocking table. The size of the table is a generator option. Each entry in the table consists of two bytes which contain the address of a user routine.

If a user routine is to be notified at one-second intervals, it must enter into the clocking table the address where notification is to take place. This is done by loading into processor register 15 the address in the user routine to which control is to be transferred and by writing the following instruction:

LABEL	♭ OPERATION ♭	OPERAND
unused	SRC	0,36

When the supervisor recognizes a one-second interrupt, it resets the one-second flip-flop and executes a branch and load (BAL) instruction through I/O register 15 to each address currently in the clocking table. When the individual routine has completed its task, control must be returned to the supervisor through I/O register 15.

The procedure to remove an entry from the clocking table is similar. The address in the user routine to which control is to be transferred is loaded into processor register 15 and the following instruction is written:

LABEL	♭ OPERATION ♭	OPERAND
unused	SRC	0,38

The supervisor responds to the resulting interrupt by matching the address in register 15 with one in the clocking table and clearing the entry to binary 0's. It is assumed that by the time a job is completed all entries in the clocking table have been cleared to binary 0's.

A.5. AUTOMATIC SUPERVISOR FUNCTIONS

The supervisor assumes the obligation of initiating the one-second interrupt and of permitting LT summary interrupts.

UP-7547 Rev. 2

Distributed to:
L. Sutherland - cc. K. Schweiger - Montreal.
J. Belanger - Quebec City
M.J. Spratt -cc. M.E. Arnold - Ottawa.
A. Caldwell - cc. T.A. Randall- Hamilton
E. Benson - Kitchener
J. Sandiford - cc. F.J. Ware - Toronto
J. D. Shapkin - Winnipeg
G. White - Regina
Norm Schwager - Saskatoon
W. Meyer - cc. L.J. Zimmel - Calgary
N.A. Rimstad - cc. C.G. Jackson - Edmonton
G. Jackson - cc. J.G. Kinvig - Vancouver
A. Shatford - Halifax.

Phil Gee,

Toronto, October 8, 1969.

Technical Services.

Standardization of Logical Unit Numbers

For your information -

Logical	Channel	System	Physical	
A	1	O.S. or M.O.S.	Card reader	
B	3		Printer	
C	2		Serial punch	
D	7		1001	
E	6		Row punch	
F	B		High speed printer	
x'10'	c		Paper tape	
x'11'	N/A		1004	
x'12'	48 & 49		DCS - 1	first
x'13'	4A & 4B			second
x'12'	40 & 41		DCS - 4	first
x'13'	42 & 43			second
x'14'	44 & 45			third
x'15'	46 & 47			fourth
0 to 9	8 or 9	M.O.S or T.O.S.	Tape only	
0 to 4	8 or 9		Tape of	Tape & Disc
5 to 9	5		Disc of	Tape & disc
0 to 4	5	D.O.S.	Disc of	Disc & Tape
5 to 9	8 or 9		Tape of	Disc & Tape
	Selector		8411 or 8414	

All systems generated after this date should conform to the above standards.

Phil Gee.

PG/ih