SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS REVISION

9200/9300 Series

Remote Communications Program

Programmer Reference

UP-7607 Rev. 3

This SPERRY UNIVAC® 9200/9300 Series Library Memo announces the release and availability of "SPERRY UNIVAC 9200/9300 Series Remote Communications Program Programmer Reference", UP-7607 Rev. 3. This is a Restricted Distribution Item (RD). Order where necessary.

This revision contains updated, corrected and added information throughout the manual. The following are major areas of changes or additions.

1. VS/9 operating system
2. REM1 declarative macro instruction
3. User supplied card input and output routines
4. Loading and executing the remote program (MOS, NCOS, and COS)
5. Supervisor generation (MOS)
6. Remote program generation (MOS, NCOS, COS)
7. Supervisor/Job Control generation (NCOS/COS)
8. Data translation
9. Code relationships

Destruction Notice: This revision supersedes and replaces "UNIVAC 9200/9200 II/9300/9300 II Systems Remote Communications Program Programmer Reference" UP-7607 Rev. 2, released on Library Memo dated November 25, 1970. Also destroyed is Updating Package A, UP-7607 Rev. 2–A released on Library Memo dated November 1972. Please destroy all copies of UP-7607 Rev. 2 and UP-7607 Rev. 2–A and their Library Memos.

Additional copies may be ordered by your local Sperry Univac Representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS:
Mailing Lists 217, 630 and 692	Mailing Lists 51, 51D, 52, 53, 53D, 54, 54D, 55, 55D, and 56 (Covers and 43 pages)	Library Memo for UP-7607 Rev. 3
		RELEASE DATE: April, 1976

SPERRY UNIVAC
9200/9300 Series
Remote Communications Program

Programmer Reference

SPERRY UNIVAC
COMPUTER SYSTEMS

UP-7607 Rev. 3

SPERRY UNIVAC
9200/9300 Series
Remote Communications Program

Programmer Reference

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

Sperry Univac is a division of Sperry Rand Corporation.

AccuScan, FASTRAND, PAGEWRITER, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are trademarks of the Sperry Rand Corporation.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-7607 Rev. 3

Part/Section	Page Number	Update Level
Cover		
Title Page		
PSS	1	
Contents	1 thru 3	
1	1, 2	
2	1 thru 6	
3	1 thru 7	
4	1 thru 8	
Appendix A	1, 2	
Appendix B	1 thru 7	
Appendix C	1, 2	
Appendix D	1, 2	
User Comment Sheet		

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Contents

1. Introduction

1.1. GENERAL

This manual describes the remote communications program provided for the SPERRY UNIVAC 9200/9300 Series. This section provides a general description of the contents of this manual.

A knowledge of the card assembler programmer reference, UP-4092 (current version) or the tape/disc assembler programmer reference, UP-7508 (current version) will aid in understanding the material presented here.

Section 2 describes the remote communications program; Section 3 describes the program operation by the remote operator; and, Section 4 provides information for generating and linking the program. Appendixes A through D are provided to describe the required hardware and software interface to the 9200/9300 Series remote communications program. Appendix A lists the data communications subsystem options with which this program is designed to run. Appendix B describes the message formats and conventions used by the remote communications program. Appendix C describes the system data translation and Appendix D describes the input and output routines that the user can substitute.

1.2. DECLARATIVE MACRO INSTRUCTIONS

A problem program must inform the system of the parameters, special conditions, current status, and options pertaining to a file. This is accomplished by including a declarative (file definition) macro instruction for each file required by the problem program. These declarative macro instructions, which are similar in form to a source code instruction, generate nonexecutable code, such as constants and storage areas for variables. Therefore, these macro instructions should be separated physically from the inline file processing coding.

The declarative macro instruction, named in the operation code, and selected keyword parameters, designated in the operand, define the file. A keyword parameter consists of a word or code immediately followed by an equal sign which is followed by one specification.

The format of the declarative macro instruction is:

LABEL	△OPERATION△	OPERAND
[symbol]	code	keyword-1=x, keyword-2=y,...,keyword-n=z

A symbol can appear in the label field. The symbol can have a maximum of four characters and must begin with an alphabetic character. The appropriate code must appear in the operation field. Keyword parameters written in the operand field may be in any order, but must be separated by commas. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation statements.

The alternate form of writing the declarative macro instructions is:

LABEL	Δ OPERATION Δ	OPERAND	72
[symbol]	code	keyword-1=x,	X
		keyword-2=y,	X
		.	.
		.	.
		.	.
		keyword-n=z	

In the alternate form, a continuation mark is necessary in column 72 of every line except the last. Each keyword parameter and specification except the last must be followed by a comma.

1.3. STATEMENT CONVENTIONS

The conventions used to illustrate statements in the manual are as follows:

- Capital letters and punctuation marks (except braces, brackets, and ellipses) 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. Remote Communications Program

2.1. GENERAL

The SPERRY UNIVAC 9200/9300 Series remote communications program permits the following operating systems to use any one of the 9200/9300 systems as a remote peripheral device:

- the EXEC II and EXEC 8 operating systems for the SPERRY UNIVAC 1106/1107/1108 Systems;
- the Virtual Memory Operating System/9 (VS/9) for the SPERRY UNIVAC 90/60 or 90/70 virtual memory systems (90/60, 90/70V);
- the OMEGA operating system for the SPERRY UNIVAC 494 System; and
- the RTOS operating system for the SPERRY UNIVAC 418-III System.

Note that, for VS/9 on a 90/60 or 90/70 virtual memory system, only the RMS-1 type of simulation is supported. Throughout this manual, the 9200/9300 Series remote communications program is referred to as the remote program. The large scale central processor using any one of the 9200/9300 Series as a remote peripheral device is referred to as the central computer. To illustrate a remote communications configuration, consider that a 9300 system is linked over communication lines to an 1108 system installation. The remote communications program enables the remote user to send his program and data over the communication lines for processing by the 1108 executive software system. The remote user later receives the complete output at the point of origin.

The remote program is the software element which interfaces with the data communications subsystem (DCS). Thus, any system may use the 9200/9300 Series as a remote peripheral device as long as it conforms to DCS specifications (Appendix A). The remote software operates as the primary program and is never linked to a user program; user input and output subroutines may be linked to the remote program in place of the suggested reader/printer/punch input/output control system (IOCS) modules, if desired.

2.2. CAPABILITIES

The remote program is capable of:

- transmitting messages comprised of data read from cards;
- receiving messages from the central computer, and
  - — printing the message; or
  - — punching the message onto cards.

In addition, the remote program provides for retransmission of messages incorrectly received due to losing the carrier.

The functions performed by the remote program are controlled by messages received from the central computer and by operator keyins at the remote computer control console.

The remote program is used as a main program in the 9200/9300 Series. In the minimum or nonconcurrent operating system (NCOS) environment, this dedicates the 9200/9300 Series to the remote program. In the concurrent operating system (COS) environment, symbionts may be run concurrently with the remote program.

2.3. MINIMUM HARDWARE AND SOFTWARE CONFIGURATIONS

The minimum hardware configuration required to use the 9200/9300 Series remote program is listed as follows:

- 8K main storage
- serial reader
- card punch (either serial or row)
- bar printer
- SPERRY UNIVAC DCS-1 or DCS-4 with the options listed in Appendix A

The minimum software configuration required to use the 9200/9300 Series remote program is the minimum operating system (MOS).

The remote program can be generated, assembled, linked, and operated in an 8K main storage configuration.

2.4. REM1 DECLARATIVE MACRO INSTRUCTION

The remote program may be generated by assembling the REM1 declarative macro instruction, which is distributed as a part of the macro library with all tape and disc software releases. The coding generated depends on the parameters specified by the user.

To make it possible to generate the remote program on an 8K card system, the REM1 macro is also distributed as three card macro libraries, named REM1A, REM1B, and REM1C. Each macro library is submitted to the preassembly macro pass with the same keyword parameters and specification that would be used with the REM1 macro instruction. The operation code for each generation is changed to REM1A, REM1B, and REM1C, respectively. The output of the three macro passes is then combined for input to the card assembler.

The format of the REM1 declarative macro instruction is:

LABEL	Δ OPERATION Δ	OPERAND	72
symbol	REM1	RCHN = nn,	X
		RMID = nn,	X
		SYST = { CARD / TAPE } ,	X
		TCHN = nn,	X
		TYPE = { RMS1 / FAS2 }	[X]
		[,ARID = nn]	[X]
		[,IN = OWNCD]	[X]
		[,OUT = OWNCD]	[X]

The symbol which is punched in the label field of the REM1 declarative macro instruction is generated into the label field of the START line and thus becomes the program identification of the assembled relocatable module.

- Receive Channel Entry

  This required keyword parameter identifies the multiplexer subchannel to which the receiver of the data communications subsystem is connected. The allowable specification of this keyword parameter is one of the odd-numbered subchannels. The format for the receive channel entry keyword parameter is:

  RCHN=nn

  where **nn** must be one of the decimal odd-numbered subchannels from 17 through 31; these subchannels correspond to hardware device addresses X'41' through X'4F'.

- Remote Identification

  This required keyword parameter is used to specify the identification of the remote site. The remote identification value is sent as part of the initial ready message to the central computer and is used by the central computer operator for identifying the remote site if voice communications are requested. The format for the remote identification keyword parameter is

  RMID=nn

  where **nn** is any decimal number between 00 and 63.

  NOTE:

  The specification made for this parameter is stored as an 8-bit byte by the 9200/9300 Series, but when the specification is transmitted, the two most significant bits of the byte are lost. Only 6 bit characters are used by the remote program for transmission. (See note for the ARID keyword parameter.)

- System Type

  This required keyword parameter is used to specify whether the 9200/9300 Series operating system in use is the card-oriented MOS or the tape- or disc-oriented NCOS or COS. The format of the systems type keyword parameter is:

  SYST= { **CARD** / **TAPE** }

where the specifications to the keyword parameter are used as follows:

CARD
The CARD specification is used if the program is to run with MOS.

TAPE
The TAPE specification is used if the program is to run with NCOS or COS.

- Transmit Channel Entry

This required keyword parameter identifies the multiplexer subchannel to which the transmitter of the data communications subsystem is connected. The allowable specification is one of the even-numbered multiplexer subchannels. The format for the transmit channel entry keyword is:

TCHN=nn

where **nn** must be one of the even numbered subchannels from 16 through 30; these subchannels correspond to hardware device addresses X'40' through X'4E'.

- Simulation Type

This required parameter is specified so that the function of a hardware plugboard (RMS-1 or phase II), used in the SPERRY UNIVAC 1004 Card Processor, can be simulated. This simulation is necessary because the 9200/9300 Series device is used as a remote peripheral by the central computer in the same way the 1004 card processor is used. The format of the simulation type keyword parameter is:

$$\text{TYPE}=\left\{\begin{matrix}\text{RMS1}\\ \text{FAS2}\end{matrix}\right\}$$

where the specifications to the keyword parameter are used as follows:

RMS1
The RMS1 specification is used when an RMS-1 plugboard is to be simulated for communication with the EXEC 8 operating system, the OMEGA operating system for the 494 system, the RTOS operating system for the 418 III system, or VS/9 for the 90/60 or 90/70 virtual memory systems.

FAS2
The FAS2 specification is used when a phase II plugboard is to be simulated for communication with the EXEC II operating system.

- Additional Remote Identification

This optional keyword parameter is used in conjunction with the RMID keyword parameter to specify the identification of the remote site when there is a need to have more than one byte represent the number of unique sites. Specifying this parameter in effect creates a 2-byte identification-number in which ARID is the most significant byte and RMID is the least significant byte. The format for the additional remote identification keyword parameter is:

ARID=nn

where **nn** is any decimal number between 00 and 63.

NOTE:

The specifications made for the ARID and RMID keyword parameters reach the central site computer as 6-bit binary configurations. The specifications are stored as 8-bit bytes by the 9200/9300 Series, but the two most significant bits of each byte are lost when the specification is transmitted. If, for instance, the specifications were RMID=61 and ARID=06, the two 8-bit configurations are:

ARID	*RMID*
0000 0110	*0011 1101*

but the transmitted binary bits would be:

000110	*111101*

which equals 0675 in octal notation.

If ARID is not used, the remote site identification must be represented by the RMID specification.

When using VS/9, any value between 00 and 63 may be used for the ARID and RMID parameters. The specific value used is irrelevant.

- User-Supplied Card Input Routine

  This optional keyword parameter is used to specify that the user will supply his own card input routine and will provide card input to the remote program in EBCDIC code.

  The format of the user-supplied card input routine keyword parameter is:

  IN=OWNCD

- User-Supplied Card Output Routine

  This optional keyword parameter is used to specify that the user will supply his own card output routine and will accept card output from the remote program in EBCDIC code. The format of the user-supplied card output routine keyword parameter is:

  OUT=OWNCD

  NOTE:

  See Appendix D for coding suggestions and examples in the use of user-supplied input and output routines.

The REM1 macro instruction keyword parameters are summarized in Table 2—1.

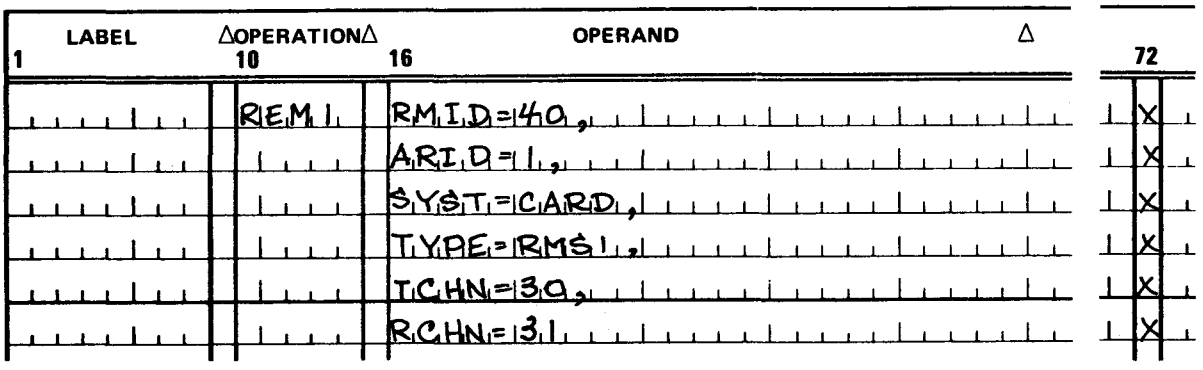
Table 2—1. Summary of REM1 Macro Instruction Keyword Parameters

Keyword	Specification	Files	Remarks
ARID	nn = decimal number between 00 and 63	X	Used in conjunction with RMID to identify remote site
IN	OWNCD	X	Specifies user-supplied card input routine
OUT	OWNCD	X	Specifies user-supplied card output routine
RCHN	nn = odd-numbered subchannel between 17 and 31	R	Identifies receiver subchannel
RMID	nn = decimal number between 00 and 63	R	Identifies remote site
SYST	{CARD {TAPE	Y Y	Identifies operating system
TCHN	nn = even-numbered subchannel between 16 and 30	R	Identifies transmitter subchannel
TYPE	{RMS1 {FAS2	Y Y	Identifies 1004 plugboard being simulated

LEGEND:

R = required parameter
X = optional parameter
Y = one specification must be used

Example:



3. Operation

3.1. GENERAL DESCRIPTION

When the remote program is loaded, it loops waiting for a ready keyin (05_{16}) from the remote operator. When this keyin is made, the remote program sends an initial ready message containing the remote identification (specified by ARID/RMID) to the central computer. This message is repeated at 5-second intervals until one of the following occurs as a result of receiving a message from the central computer:

- The remote program begins processing a message from the central computer either by printing or punching it into cards.
- A 6601_{16} is displayed on the remote control console, indicating that the central operator wants to speak to the remote operator. When operating with VS/9 and a 90/60 or 90/70 virtual memory system, a 6601_{16} display on the remote console indicates that the host has disconnected the 9200/9300 system.
- The remote program prints READY and homes paper on the 9200/9300 Series printer. This action signals the remote operator that communications have been established with the central computer and that the remote program is now ready to accept further direction from the remote operator.

  NOTE:

  The DATA switch on the data set must be pressed after the program is loaded and looping, but before the ready keyin is made.

3.2. REMOTE OPERATOR KEYINS

The remote operator can direct the operation of the remote program by setting the data entry switches to the desired hexadecimal value and then pressing the OP REQ (operator request) switch.

The following table illustrates acceptable keyin settings; all others are ignored by the remote program.

Hexadecimal	Designation
05	Ready
06	Read
07	Halt
08	Halt-go-voice
09	Abort-print
0A	Abort-punch
0C	Offline

The following sections describe the use of and action caused by each keyin.

NOTE:

Due to systems requirements, the OP REQ switch may not always be operable. This is indicated when the OP REQ light is off. Requests are accepted only when the light is on.

3.2.1. Ready (05_{16})

Execute this keyin after loading the program. This causes a message containing the remote identification to be sent to the central computer at 5-second intervals until the central computer responds, causing one of the actions described in 3.1.

3.2.2. Read (06_{16})

Execute this keyin to indicate that a card deck is available for reading at the remote computer. Upon receipt of the proper response from the central computer, the card deck is read and its data transmitted to the central computer.

The first time a read keyin is made, the card input file is opened. After each read keyin, the card input file is read until an image containing @RUN in columns 1—4 is encountered; this is the first card image transmitted to the central computer.

No additional cards will be read after a stop card has been detected by the program. However, unless an offline keyin has been made, card reading may be reinitiated by keying in an 06_{16}(read).

NOTE:

The card deck should contain a stop card (7—8 punch in columns 1 and 2) to indicate the end of the deck. The stop card is not included in the data message.

3.2.3. Halt (07_{16})

Execute the keyin to halt the computer when it is necessary to perform some operation before proceeding with transmission. The display associated with this keyin occurs after a delay of approximately 15 seconds. To continue, press START switch; the remote program returns to the operational state present before the keyin was initiated.

3.2.4. Halt-Go-Voice (08_{16})

Execute this keyin to request voice communication with the central operator before proceeding with transmission. The display associated with this keyin occurs after a delay of approximately 15 seconds. To continue, press START switch; the remote program returns to the operational state present before the keyin was initiated. When running with VS/9 and the 90/60 or 90/70 virtual memory system, the halt-go-voice command from the 9200/9300 is treated as a halt command. When received from the host; the halt-go-voice command indicates that the host has disconnected the 9200/9300 and no other transmissions will be sent or should be expected.

3.2.5. Abort-Print (09_{16})

Execute this keyin to request the central computer to discontinue transmitting data for the current print file.

3.2.6. Abort-Punch ($0A_{16}$)

Execute this keyin to request the central computer to discontinue transmitting data for the current punch file.

3.2.7. Offline (OC_{16})

Execute this keyin to go offline. If sending data, the remote program will continue until a stop card is detected, or if receiving data, the remote program will continue until all output files have been processed. When running with VS/9 and the 90/60 or 90/70 virtual memory system, the offline command is treated as an immediate disconnect.

The remote program does not go to its end-of-job routine until the offline keyin is made. This keyin must be used to ensure the closing of all files. Since the remote program print routine always lags one line behind the print file received from the central computer, the last line of the print file may not appear until the offline keyin is made.

After closing all the files, the remote program returns control to the operating system through the EOJ macro. This causes a X'41EF' display under MOS; under higher level operating systems, job control is loaded so that processing of the control stream can resume.

3.3. REMOTE CONTROL CONSOLE DISPLAYS

Several control console displays are provided to inform the operator of the status of the remote program. Listed in Table 3—1 are the hexadecimal displays, reasons for the displays, and operator actions to be taken. The hexadecimal displays are usually indicative of DCS failures, which in many cases are unrecoverable; however, the operator has the option to attempt recovery.

NOTE:

IOCS displays may also occur during the operation of the remote program. For these displays, refer to halt displays programmer/operator reference, UP-7719 (current version).

Table 3—1. Remote Control Console Displays (Part 1 of 2)

Hexadecimal Display	Reason	Action
6601	Halt-go-voice from central computer	At completion of voice communications, press START switch to continue. For VS/9, see note 3.
6602	Halt keyin by remote operator	Press START switch to continue.
6603	Halt-go-voice keyin by remote operator	At completion of voice communications, press START switch to continue. For VS/9, see note 3.
6604	Carrier lost on input line terminal	Press START switch to request retransmission.
6605	Nonoperational control unit or channel, or offline	Correct problem; press START switch to retransmit.
6609	Specifically applies to input line terminal (LT) of DCS. Indicates a nonoperational control unit or channel.	Check setting of input LT and line terminal controller (LTC) switches; press START switch to attempt recovery. See notes 1 and 2.

Table 3-1. Remote Control Console Displays (Part 2 of 2)

Hexadecimal Display	Reason	Action
6610	Send-data command to output LT of DCS rejected	See notes 1 and 2.
6611	Look-for-sync command to input LT of DCS rejected	See notes 1 and 2.
6612	Sense command to output LT of DCS rejected	See notes 1 and 2.
6613	Turn-off command to input LT of DCS rejected	See notes 1 and 2.
6615	Sense command to input LT of DCS rejected	See notes 1 and 2.
6618	Sense bytes stored from output LT of DCS indicate a status not specifically tested for.	See notes 1 and 2.
6619	Sense bytes stored from input LT of DCS indicate a status not specifically tested for.	See notes 1 and 2.
6620	Initial turn-on to input LT of DCS was not accepted.	If DCS was offline, place switches on the DCS to ON position; press START switch on processor console to continue.

NOTES:

1. The command, in most cases, has been issued to the LT of the DCS five times. Press START switch to try again.
2. To cancel, key in nonzero to main storage location 4, then press START switch.
3. Refer to 3.2.4.

3.4. PRINTER CONSIDERATIONS

The remote program is designed to operate with a 132-print position bar printer. When operating with a 120- or 96-print position printer, the central program should restrict the size of its print messages to the number of print positions available on the remote computer. If the central program sends a print message that exceeds the number of available print positions, the message will be truncated on the right, and the excess will not be printed. Note that VS/9 for REM1 only supports a 132 print-position printer.

3.5. LOADING AND EXECUTING THE REMOTE PROGRAM (MOS)

In a MOS environment, the supervisor must be loaded into main storage before the remote program can be executed. The procedures specified in the following subsections and in Figure 3—1 assume that the remote program (object deck labelled REM1) is to be loaded through the standard (0711) card reader immediately after the supervisor (deck labeled MOS) is loaded.

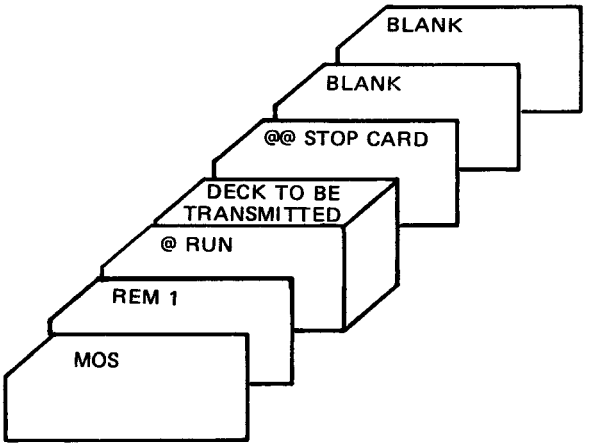


Figure 3—1. Control Stream for MOS Environment

3.5.1. Control Stream Format

Place the MOS card deck, immediately followed by the REM1 card deck, in the card reader, face down, 9-edge leading. If no card data is to be transmitted to the central processor, the REM1 card deck should be followed by one blank card. If a data card deck is to be transmitted, its first card should contain @RUN in columns 1—4. The data card deck should be followed by the stop card (@@ in columns 1 and 2) and two blank cards.

3.5.2. Loading the Supervisor

To load the supervisor into main storage, proceed as follows:

1. Set the DATA ENTRY switches to X'01'.
2. Press the READER CLEAR and READER FEED switches. This feeds the first card of the MOS card deck to the read station.
3. Press the PROC CLEAR and CHANNEL CLEAR switches.
4. Press the LOAD switch to ON and press the START switch.
5. Press the LOAD switch to OFF and press the START switch. This initiates the loading of the supervisor into main storage.
6. When the supervisor is loaded, X'41FF' is displayed.

3.5.3. Loading the Remote Program

Without feeding any cards in the reader, press the PROC CLEAR switch, then press the LOAD switch to ON and press the START switch. Press the LOAD switch to OFF and press the START switch. The remote program now loads into main storage and loops.

You may now establish communications with the central processor by pressing the DATA switch on the modem and initiating a ready keyin.

3.6. LOADING AND EXECUTING THE REMOTE PROGRAM (NCOS/COS)

In a NCOS or COS environment, the supervisor must first be loaded into main storage from tape or disc. The remote program is executed by submission of a control stream consisting of 9200/9300 Series job control statements and, optionally, a data deck to be transmitted to the central computer. The procedures specified in the following subsections and in Figure 3—2 assume that the remote program is named REM1, that it is the first program to be executed, and that the standard (0711) card reader is the control stream reader.

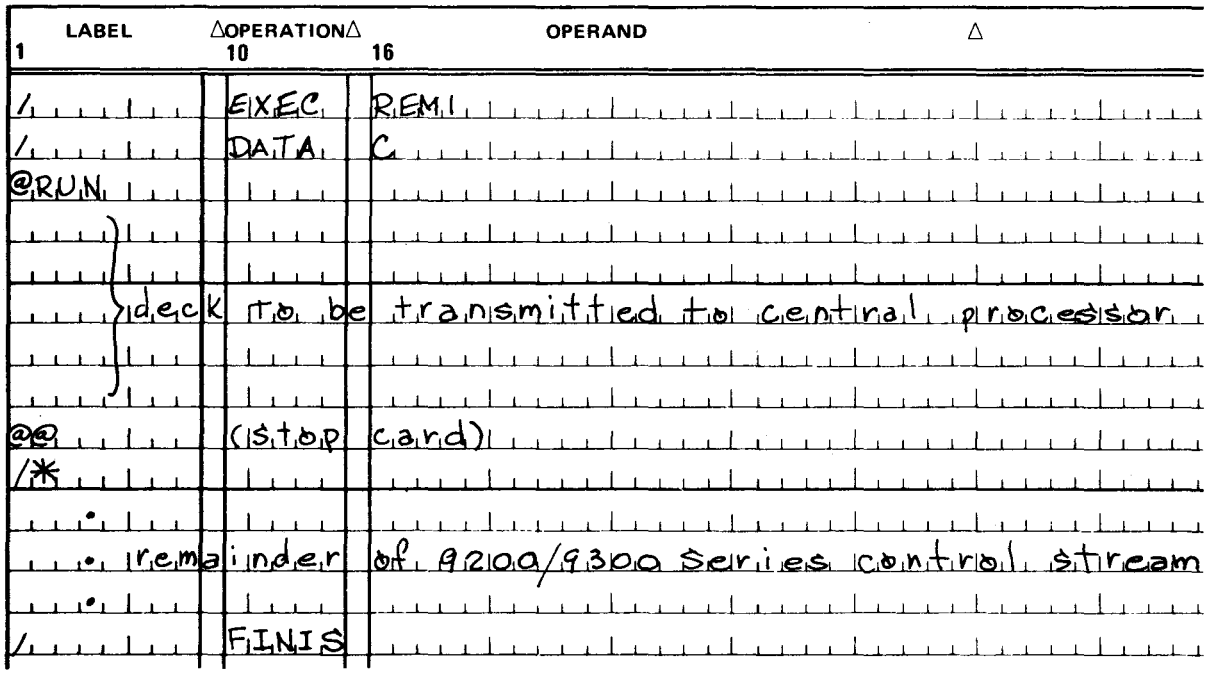


Figure 3—2. Control Stream for NCOS/COS Environment

3.6.1. Control Stream Format

Place the 9200/9300 Series card deck (as shown in Figure 3—2) in the card reader, face down and with the 9-edge leading. Press the READER CLEAR and the READER FEED switches. This feeds the first card of the 9200/9300 Series card deck to the read station. The card deck that is used for executing REM1 may be preceded or followed by any other desired 9200/9300 Series job control statements. If no card data is to be transmitted to the central processor, the cards from / DATA C through /* must be omitted.

3.6.2. Loading the Supervisor

To load the supervisor into main storage, proceed as follows:

1. Mount the system tape or disc on the unit to be used as the system resident drive. Set the device address of this unit (usually X'C0' for tape or X'30' for disc) using the DATA ENTRY switches.

2. Press the PROC CLEAR and CHANNEL CLEAR switches.

3. Press the LOAD switch to ON and press the START switch.

4. Press the LOAD switch to OFF and press the START switch. This initiates the loading of the supervisor and job control into main storage. The card deck will start to be read.

3.6.3. Loading The Remote Program

When the / EXEC REM1 statement is encountered during the processing specified in 3.6.2, step 4, the supervisor locates and loads the remote program, which loops until an input is received to activate it.

You may now establish communications with the central processor by pressing the DATA switch on the modem and initiating a ready keyin.

4. Assembling and Linking

4.1. GENERAL

The program is generated from the REM1 macro instruction in the MOS, NCOS, or COS environment. Prior to execution, REM1 must be linked with various other relocatable modules as described in this section.

REM1 must be linked so that it is loaded at a starting address that is a multiple of 128. Linking may be accomplished by using an absolute address in the PHASE or PRGM card or by using a MOD as input to the linker. (See tape/disc assembler programmer reference, UP-7508 (current version).)

4.2. MINIMUM OPERATING SYSTEM GENERATION

To use the remote program in a minimum operating system environment, a MOS supervisor which includes communications capability must be available. The following subsections describe the generation of the supervisor and remote program for use in an 8K system. Both generations can be accomplished on the 8K system.

In order for the remote program to operate in 8K of main storage, the MOS supervisor must be generated without the tape dispatcher or disc dispatcher. The following example shows the keyword parameters of the MOS macro call which must be specified to generate an MOS for use with the remote program:

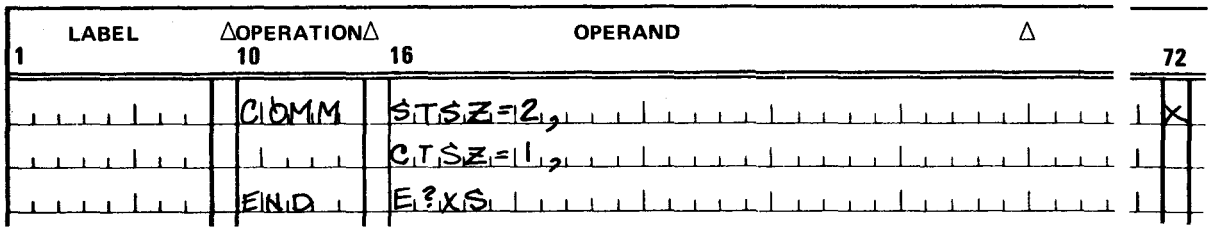
```
LABEL      ΔOPERATIONΔ      OPERAND                    Δ
1          10               16                                72
           MOS              ALTR=ALL,                         X
                            SIZE=8,                           X
                            COMM=YES,                         X
                            STSZ=2,                           X
                            CTSZ=1                            X
           END
```

NOTE:

The specification for the SIZE parameter is the main storage size of the device in which the MOS is to be used. For additional information concerning the MOS, see the minimum operating system programmer reference, UP-7547 (current version).

4.2.1. Supervisor Generation (MOS)

To generate the MOS communications handler, the following additional macro call and keyword parameters must be specified:



The output of the two preassembly macro passes may then be combined, discarding the END card produced by the first macro pass, and assembled by the card assembler.

The logical/physical unit (LU/PU) tables should also be generated and assembled. Subsection 4.3.1 contains a sample set of specifications for LU/PU table generation that may be adapted to the MOS environment.

NOTE:

Due to the restrictions of the 8K environment in which it must operate, the remote program does not inspect the LU or PU tables to verify the existence or availability of the communications device. Therefore, you may eliminate these tables from your MOS if you have no other need for them. The EXTRN statements for E?LT and E?FT should be removed from the output of the MOS macro generation to avoid linker errors.

To produce a loadable MOS supervisor, the card linker must be used to link a card program loader to the output of the assemblies of the main supervisor module and of the logical/physical units.

4.2.2. Remote Program Generator (MOS)

To generate a remote program for use in the MOS environment, the following procedure must be followed:

1. Macro-generate REM1A, REM1B, and REM1C source decks, using REM1 parameters and specifications described in Section 3.

2. Combine the output of the three macro generations, removing the END cards from each of the first two decks. The END card for the combined deck should contain the operand G?Z. Submit this combined deck to the card assembler so that the main relocatable element of the remote program can be generated.

3. Macro-generate and assemble the reader, printer, and punch IOCS routines, using parameters and specifications described in 4.4 through 4.6.

4. Link the relocatable output of the assemblies in steps 2 and 3 to a loader and to the translate tables G?XX and TBP8 to produce a loadable remote program. The following rules should be observed when preparing the input to the card linker.

   a. The base address specified in the PHASE card should be greater than the highest address occupied by the MOS supervisor and must be a multiple of 128_{10}; 1536_{10} (X'600') is usually satisfactory unless the supervisor includes optional features other than communications handling.

   b. The first relocatable element in the linker input deck should be the loader, followed by the main element of the remote program. This ensures that the communications buffers are located at addresses which are multiples of 128_{10}. Other elements may follow the main module in any sequence.

c. When the loader is linked to a program, certain external references within the loader must be satisfied by means of linker equate statements. The meanings of these references are discussed in Section 8 of the card assembler programmer reference, UP-4092 (current version). The following equate statements are satisfactory definitions for use with the remote program in an 8K system:

1 LABEL	Δ OPERATION Δ 10	16 OPERAND	Δ
L?AM	EQU	0	
L?AR	EQU	7800	
L?CH	EQU	0	
L?HI	EQU	129	
L?LO	EQU	128	
L?PG	EQU	7880	

Figure 4—1 illustrates the principal steps in generation of the remote program. The card deck setup for macro generation of each IOCS module is similar to that used for generating the remote program. IOCS modules may be assembled separately or may be combined into a single assembly, whichever is desired.

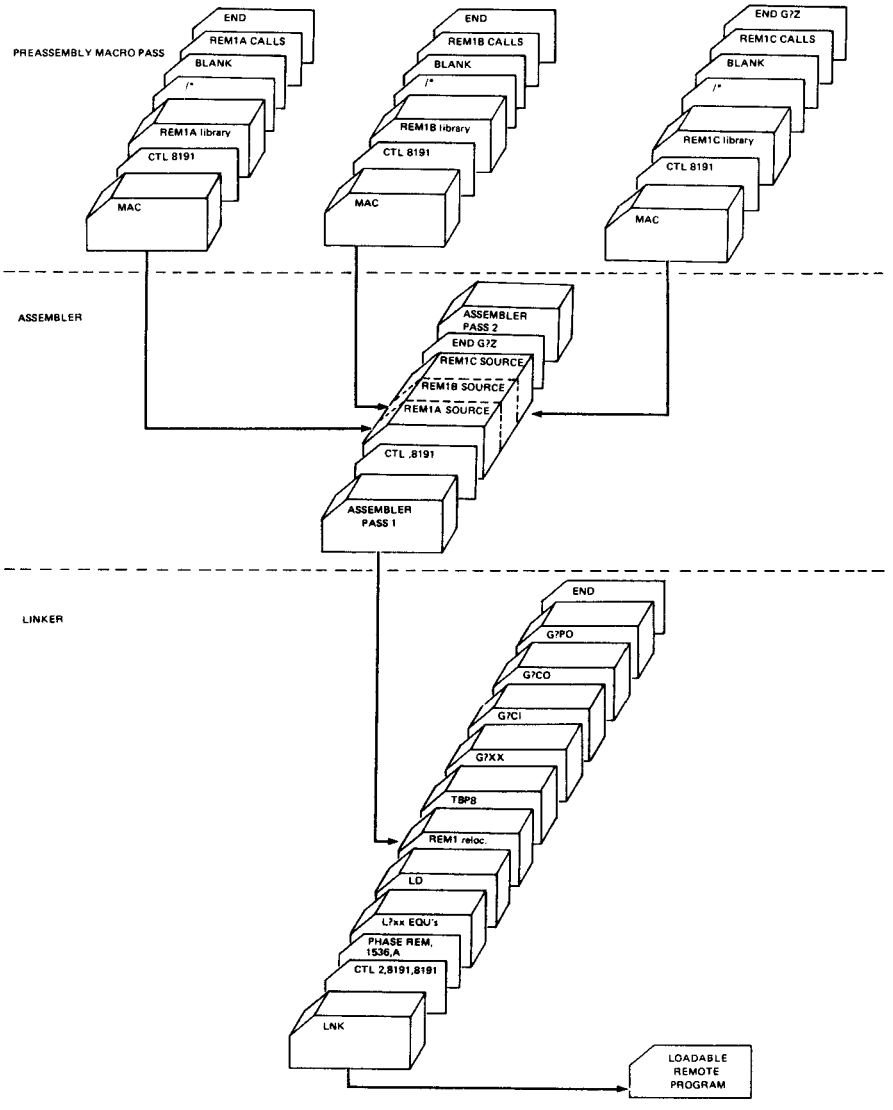


Figure 4—1. REM1 Generation Procedures

4.3. NONCONCURRENT OR CONCURRENT OPERATING SYSTEM GENERATION

To use the remote program in a NCOS or COS environment, supervisor and job control programs that have a communications capability must be available. This section describes the generation of the operating system and remote program for a tape-resident NCOS environment. The generation in a disc-resident or COS environment is similar. Consult the operating system programmer reference, UP-7531 (current version) for information on the operating system generation, and the tape/disc assembler programmer reference, UP-7508 (current version) for information on use of the various assemblers and linkers.

4.3.1. Supervisor/Job Control Generation (NCOS/COS)

The following examples show control card groups for assembling and linking the operating system. These examples are given under the following categories:

1. Assembling the NCOS
2. Assembling the LU/PU tables
3. Linking to obtain a loadable NCOS
4. Assembling the job control program
5. Linking the job control program

It should be noted that the examples given are presented as a guide for the remote program user; they are not intended to be a rigid procedure that the user should follow.

- Assembling the NCOS

LABEL (1)	OPERATION (10)	OPERAND (16)	Δ	COMMENTS
/,	EXEC,	ASMB,		
/,	PARAM,	0,070,,30,,MACROLIB/UNIVAC/EXECMODL,		
/,	DATA,	T,		
,	NCOS,	ALTR=ALL,,SIZE=116,,CSR=CRD,,DEVA=8,,TPCL=8,,		
,	,	COMM=YES,,STSZ=21,,CTSZ=1,		
,	END,	,		
/*	,	,		

- Assembling the LU/PU tables

```
LABEL     OPERATION     OPERAND                                          COMMENTS
1         10            16
/         EXEC          ASMB
/         PARAM         0070,30,MACROLIB/UNIVAC/EXECMODL/LUPUTABL
/         DATA          T
          CFGOD
          PUTBL         TAPE,8,0,0,0,9
          PUTBL         TAPE,8,1,1,1,9
          PUTBL         TAPE,8,2,2,2,9
          PUTBL         TAPE,8,3,3,3,9
          PUTBL         CRD,,,8,C
          PUTBL         CRP,,,,9
          PUTBL         PRNT,,,,X'0A'
          PUTBL         CC,7,,,X'0B'
          PUTBL         RRP,6,,,X'0C'
          PUTBL         CDVC,25,0,0,X'0D',,B,24
          END
/*
```

NOTE:

Save this relocatable element so that it can be linked with the job control program.

- Linking to obtain a loadable NCOS

```
/         EXEC          TLOO
/         DATA          C
          PRGM          ,SUPR,256
          INCLUDE       /SUPR,02
          INCLUDE       RLOCATBL/TBRD,00
          INCLUDE       /J?INT,02
```

- Assembling the job control program

```
/         EXEC          ASMB
/         PARAM         0070,30,MACROLIB/UNIVAC/EXECMODL
/         DATA          T
          JCNC          COMM=YES
          END           STRT
/*
```

- Linking the job control program

```
LABEL     ΔOPERATIONΔ     OPERAND                    Δ     COMMENTS
1         10              16                                       72
/         EXEC    T,L,O,O
/         DATA    C
          PRGM      JBCN,X'1000'
          INCLUDE /JBCN,02
          INCLUDE /J?INT,02
/*
```

4.3.2. Remote Program Generation (NCOS/COS)

To generate a remote program for use in a NCOS or COS environment, the REM1 declarative macro must be assembled, using the parameters and specifications described in Section 3. The reader, printer, and punch IOCS modules must be assembled using parameters and specifications described in 4.4 through 4.6. The following examples show control card groups for assembling and linking the remote program.

- Assembling the remote program and its IOCS modules

  There are several different ways the generation of IOCS modules can be handled. This example shows the control stream reader, card punch, and printer IOCS functions as one element which is linked with the REM1 element to form one loadable program.

```
/         EXEC    ASMA
/         DATA    T
RMSL      REM1    TYPE=RMS1,RMID=02,RCHN=25,TCHN=24,SYS=TAPE
          END     G?Z
/*
/         EXEC    ASMB
/         DATA    T
RETAIN    START   0
          USING   *,0
G?CI      DTFCS   EDFA=G?EE
G?CO      DTFRP   DEVA=9,OUAR=G?HA,OTBL=TBP8,MODE=TRANS,                X
                  PUNR=YES,TYPE=OUTPUT
G?PO      DTFPR   BKSZ=132,CNTL=YES,DEVA=X'0A',FONT=G3
          END
/*
/         SWAP    01,02   (IF FOLLOWED BY LINKER RUN)
```

- Linking the remote program and the IOCS functions

```
1          LABEL     ΔOPERATIONΔ     OPERAND                    Δ
                     10              16
/                    EXEC            TL00
/                    DATA            C
                     PRGM            EXC8,4480
                     LIBE            NONE
                     INCLUDE         /RMS1,02
                     INCLUDE         /RETAID,02
                     INCLUDE         /G?XX,00
                     INCLUDE         /TBP8,00
/*
```

NOTE:

The base address of the remote program must be a multiple of 128.

4.4. CARD READ IOCS

The following macro instruction must be presented to the preassembly macro pass to generate the required card reader IOCS module:

LABEL	ΔOPERATIONΔ	OPERAND
G?CI	DTFCR	IOA1=G?C8,MODE=CC,SENT=NO

In a NCOS/COS environment, the control stream reader IOCS is generated as follows:

LABEL	ΔOPERATIONΔ	OPERAND
G?CI	DTFCS	EOFA=G?EE

4.5. CARD PUNCH IOCS

The following macro instruction must be presented to the preassembly macro pass to generate the required card punch IOCS module, if using the serial read/punch:

LABEL	ΔOPERATIONΔ	OPERAND
G?CO	DTFRP	OUAR=G?HA,OTBL=TBP8,MODE=TRANS, PUNR=YES,TYPF=OUTPUT

if using the row read/punch:

LABEL	Δ OPERATION Δ	OPERAND
G?CO	DTFRW	OUAR=G?HA,OTBL=TBP8,MODE=TRANS, PUNR=YES,TYPF=OUTPUT,CHNL=n

where:

n
Is the number of the multiplexer subchannel to which the row read/punch is attached.

When assembling the serial or row punch IOCS for use with NCOS or COS, DEVA=x must also be specified, where x is the LU number of the punch.

4.6. PRINTER IOCS

The following macro instruction must be presented to the preassembly macro pass to generate the required bar printer IOCS module:

LABEL	Δ OPERATION Δ	OPERAND
G?PO	DTFPR	BKSZ=132,CNTL=YES,FONT=63,PRAD=1

When assembling the bar printer IOCS with NCOS or COS, DEVA=x must also be specified, where x is the LU number of the punch.

Appendix A. Data Transmission Conventions

A.1. GENERAL

The 9200/9300 Series remote communications program is designed to be run with the following conventions:

- The communications interface (CI) specification numbers are F1002—03, —04, or —05. These match Bell Telephone Company's data set models 201B1, 201A3, 301B, and 303C. The F1002—03 CI is used with the 201B1 data set; the F1002—04 with the 201A3; and the F1002—05 with the 301B or 303C. For the 303C data set, the switched carrier selection option must be selected in the CI.

- The longitudinal redundancy check (LRC) option specification numbers are F1008—99 for the DCS—1 and F1008—00 for the DCS—4.

- Input LT (F1005—98)

  1. The span of the LRC is from the first character following the synch characters up to, but not including, the end-of-message (EOM) character. The LRC character has no parity; the parity bit is always zero. The parity of the message is determined by an inclusive OR process (half-add with no carry). The LRC accumulation is even.

  2. Data parity checking is odd.

  3. Data mode synchronization is two synch characters.

  4. The synch character recognized is 35_{16}.

  5. There is no idle character detection, automatic LFS, or parity error flag recognition.

  6. The character width used is six bits plus one parity bit.

- Control character bit configurations

	Prior to Transmission	Online	After Reception
SOM	1000 0000	1 000 000	0000 0000
EOM	1101 0101	1 010 101 ↑ parity bit	0001 0101

The program uses an 80_{16} with odd parity for the start-of-message (SOM) character and a $D5_{16}$ with even parity for the EOM character for the output line terminal. On input, the character recognized as the SOM is a 40_{16} with odd parity, which appears in the 9200/9300 main storage as 00_{16}; the character recognized as the EOM is 55_{16} with even parity, but this character appears as 15_{16} in main storage.

For VS/9 and the 90/60 or 90/70 virtual memory system, the bit configuration for the control characters as they appear on the communications line, with parity, is as follows:

SOM	1 000 000
EOM	1 010 101
SYNC	0 110 101

- Output LT (F1005—98)

1. The LRC is as described previously.
2. Data parity generation is odd.
3. The hardware generated synch character is 35_{16}.
4. There is no idle character generation.
5. The character width is six bits plus one parity bit. The characters are transmitted least-significant-bit first, and parity bit last.

Appendix B. Software Conventions

B.1. SCOPE

This section describes the message formats and the operating conventions used by the 9200/9300 Series remote communications program. The information contained herein is not necessary for programming or operating the remote program when interfacing with any of the following at the central computer:

- EXEC II operating system on the 1106/1107/1108 systems
- EXEC 8 operating system on the 1106/1108/1110 systems
- OMEGA operating system on the 494 system
- RTOS operating system on the 418 III system
- VS/9 on the 90/60 or 90/70 virtual memory systems

However, this section is intended to describe the requirements for providing a software interface between the remote program and a central program for systems other than those listed. The messages are therefore given in octal notation as they are when received or transmitted by the central site computer.

B.2. REMOTE MESSAGES

All messages from the remote computer are preceded by three synchronization characters and followed by an LRC character, but these characters are automatically generated by the DCS and *will not be further mentioned in this description.*

B.2.1. Initial Ready Message

An initial ready message consists of:

1. an SOM character, which is a 00_8 with odd parity;
2. a READY character, which is a 05_8 with odd parity;
3. some number of 1004 XS-3 characters with odd parity which represent the site identification of the 9200/9300 Series. (These are the values specified for ARID and RMID at generation time.) If the simulation type is RMS1, the number of characters is 5, the first three of which are binary zeros and are ignored; if the simulation type is phase II, the number is 2.

4. an EOM character, which is a 25_8 with even parity.

For VS/9 on a 90/60 or 90/70 virtual memory system, the initial ready message consists of:

1. an SOM character which is an X'40';
2. a READY character which is an X'45';
3. a SITE-ID which is five XS-3 characters with vertical parity; and
4. an EOM character which is an X'55'.

B.2.2. Negative Acknowledge Message (NACK)

A NACK message consists of:

1. an SOM character;
2. a 57_8 with odd parity;
3. some number of nonsignificant characters with odd parity. If the simulation type is RMS1, the number of characters is 5; if the simulation type is phase II, the number is 1; and
4. an EOM character.

B.2.3. Acknowledge Message (ACK)

An ACK message consists of:

1. SOM;
2. an octal control character with odd parity;
3. some number of nonsignificant characters with odd parity. If the simulation type is RMS1, the number of characters is 5; if the simulation type is phase II, there are no insignificant characters; and
4. EOM.

B.2.4. Data Message

A data message consists of:

1. SOM;
2. an octal control character with odd parity;
3. some number of 1004 XS-3 data characters with odd parity. If the simulation type is RMS1, the number of characters is 330; if the simulation type is phase II, the number is 320; and
4. EOM.

B.2.5. Control Characters

Possible control characters are as follows:

Control Character		Designation
00_8	(00_{16})	No change
05_8	(05_{16})	Ready
06_8	(06_{16})	Read
07_8	(07_{16})	Halt
10_8	(08_{16})	Halt-go-voice
11_8	(09_{16})	Abort-print
12_8	$(0A_{16})$	Abort-punch
13_8	$(0B_{16})$	End-input
14_8	$(0C_{16})$	Offline

The end-input control character is applicable to data messages only. All others are applicable to acknowledge and data messages.

B.3. CENTRAL MESSAGES

All messages from the central computer are preceded by four synch characters and followed by an LRC character and an end-of-buffer (EOB) character. Of these characters, only the LRC character is stored in the 9200/9300 Series main storage. These characters are automatically handled by the DCS and will not be further mentioned in this description.

B.3.1. Function Messages

A function message consists of the following:

1. SOM;
2. an octal function character with odd parity; and
3. EOM.

Five nonsignificant characters with odd parity may optionally appear between the function character and EOM.

Possible function characters are as follows:

Function Character		Function
27_8	(17_{16})	Probe
10_8	(08_{16})	Output
11_8	(09_{16})	Output
44_8	(24_{16})	Input
45_8	(25_{16})	Input
25_8	(15_{16})	Halt-go-voice

In the case of an output message, the function message is immediately followed by a data message, which consists of:

1. SOM;

2. a maximum of 330 XS-3 data characters (1004 card processor) with odd parity. If the simulation type is RMS1, the number of data characters is 330; if the simulation type is phase II, the number is 320; and

3. EOM.

The data can be either information to be printed or information to be punched, but both kinds of information cannot be mixed in one data message.

B.3.1.1. Output Function Characters

The function character for output is either 10 or 11. The first output message has a function character of 10. The function character then alternates between 11 and 10 for each new output message. Only if an output message is retransmitted does the function character remain the same.

B.3.1.2. Input Function Characters

The function character for input is either 44 or 45. The first input message has a function character of 44. The function character then alternates between 45 and 44 for each new input request. If a request for retransmission is made, the function character remains the same.

B.4. INPUT DATA MESSAGES

When the remote program receives an input function request from the central computer, the read routine of the program begins to read cards, compress out spaces where possible, and pack the compressed data into a buffer area. The remote program removes all trailing spaces and many interior spaces from the data on any card not containing either of the special characters (⌑ and ≠) used by the compression routine. A lozenge (⌑) placed in the message represents from one to eight deleted spaces. The unequal sign (≠) is used to indicate the end of the card. Two contiguous signs (≠ ≠) indicate the end of data in the buffer.

A card containing either a lozenge or an unequal sign cannot be compressed and its data is preceded by a B in the message; data from a compressed card is preceded by a C.

The compression routine operates as follows. The 80-column card is divided in 10 tab stops of 8 characters per stop. Only trailing spaces in each tab stop will be removed. For example, no spaces will be removed from the following tab stop because they are followed by data:

dd△△△ddd

where:

d = data

△ = space

However, if the eight characters were as follows:

dddd△△△△

the four spaces at the end would be removed and replaced by a lozenge in the message.

Cards are packed in the buffer until an overflow condition occurs; the last card, which caused the overflow, is saved to be placed first in the next buffer.

The REM1—VS/9 package supports the compression features as described. However, special input characters denoting compression and end-of-logical message are the multipunches 0—7—8 and 0—2—8, respectively. The presence of either of these characters within an input message causes REM1 to generate a noncompressed message preceded by a B character.

B.5. OUTPUT DATA MESSAGES

The remote program is used to decompress output data messages received from the central computer and to prepare these messages for punching and printing. To obtain proper reconstruction of an output message, the central computer must compress data messages by the method described in B.4.

To decompress output messages and to initiate the actions required for punching, printing, or spacing, the output routine checks the first data record in the buffer to determine the necessary action to be taken. When the action to be taken has been determined, the first record is moved to an appropriate buffer where the data is decompressed and then punched or printed as required. The record removed from the buffer is replaced by the next record in the buffer and so on until the buffer is empty or an end-of-message is specified.

The action of punching, printing, and spacing is determined by the interpretation of special characters associated with each record. For example, a record consists of 80 characters. The end of each record is signified by a fieldata unequal symbol which immediately follows the 80 characters. The output routine recognizes this symbol as the end of record and therefore makes certain that only those data characters up to but not including the unequal symbol form part of the data message. The specification for punching, printing, or spacing is determined by the special character preceding each data record; this character will always be the first character in the buffer. If this character is a B, the output routine moves the record associated with the character to an appropriate buffer where the record is decompressed and then punched onto a data card. When the character is a C, the output routine moves the record associated with the character to an appropriate buffer where the record is compressed and then punched onto a data card. An asterisk in the first character position of the buffer specifies that the paper in the printer is to be advanced to its home position before printing the compressed message. If the first character position is not a B, C, or asterisk, the first two characters of the message are treated as the number of spaces to move the printer carriage before printing a compressed message. The routine, upon completion of this action, checks the first character in the buffer to determine the next action to be performed.

When none of the aforementioned special characters is found as the first character of the buffer, the output routine assumes that the first two characters of the data record represent a space code. The routine therefore spaces the paper in the printer accordingly. When printing is specified, the first two characters of the data record always signify the number of line spaces that are to appear between each line of print.

As previously stated, the output routine repeats its entire process each time a data record is moved out of the buffer. This continues until an end-of-message or an empty buffer condition exists. This condition is conveyed to the routine by two consecutive unequal signs in the data message. When the two consecutive unequal signs are found, an ACK message is constructed and sent to the central computer.

The REM1—VS/9 package supports the decompression feature as described. Printer output supports all 9200/9300 graphics listed except the ¢ and " graphics which denote end-of-logical record and compression, respectively.

B.6. REMOTE PROGRAM OPERATING CONVENTIONS

This section describes various points of operation of the remote program in greater detail than has previously been given in this manual. This detail is provided to emphasize certain considerations which must be made when interfacing with the remote program.

B.6.1. Remote Response to Central Function Messages

In the absence of a keyin from the remote operator after the initial ready keyin, the remote program will acknowledge the following function messages from the central computer:

- Probe

  The remote program responds to a probe function message with an acknowledge message (ACK).

- Output

  Data accompanies an output message. If any part of the output message is incorrectly received, the remote program responds with a negative acknowledge message (NACK). If the output message is correctly received, it is checked to see if it is data not yet printed or punched or if it is a retransmission of data already printed or punched. In the former case, the data is printed or punched and an acknowledge message is sent to the central computer; in the latter case, the only action taken is to send an acknowledge message to the central computer.

- Halt-go-voice

  The remote program responds to a halt-go-voice function message with a halt-and-proceed instruction. No more messages will be accepted from the central computer until the remote operator presses the START switch. Then the remote program sends a ready message to the central computer and returns to the operational state present before the halt-go-voice message was received. For VS/9 and the 90/60 or 90/70 virtual memory systems, see 3.2.4.

After having sent an acknowledge message containing a read control character, the remote program will respond to a fourth function message, input. If any part of an input function message is incorrectly received, a negative acknowledge message is sent to the central computer. The remote program checks a correctly received input message to determine if it is a request for retransmission of input that has already been sent. If it is, the input is retransmitted. If it is not, a message containing data newly read from the card deck is transmitted. When a stop card is detected in the data read from the card deck, the control character in the message containing the last of the card data is set to end-input (13_8). After an end-input control character is sent, the remote program will not accept an input function message from the central computer.

B.6.2. Start After Halt Keyin

When the remote operator presses the START switch after having made a halt keyin, the remote program sends an acknowledge message containing a ready control character to the central computer.

B.6.3. Start After Halt-Go-Voice Keyin

When the remote operator presses the START switch after having made a halt-go-voice keyin, the remote program sends an acknowledge message containing a ready control character to the central computer.

B.6.4. Relation of Operator Keyins to Control Characters

Upon receiving an operator keyin, the remote program inserts the corresponding control character in an acknowledge message for transmission to the central computer. The keyins and their corresponding control characters are as follows:

Hexadecimal Keyin	Octal Control Character	Designation
NA	00	No change
05	05	Ready
06	06	Read
07	07	Halt
08	10	Halt-go-voice
09	11	Abort-print
0A	12	Abort-punch
NA	13	End-input
0C	14	Offline

Appendix C. Data Translation

C.1. CARD INPUT

Card input to the remote program is read in 9200/9300 Series compressed code, processed in compressed code, then translated to XS-3 by means of translate table G?CX for transmission to the central computer. G?CX is one of two translate tables that comprise the Sperry Univac-supplied relocatable element G?XX. Refer to Table C—1 for the relationship of card input codes to other codes used by the remote program.

C.2. PRINTER OUTPUT

Print data is received by the 9200/9300 Series in XS—3 from the central computer, translated into EBCDIC through the translate table G?XE, processed, and printed in EBCDIC. G?XE is one of the two translate tables which comprise the Sperry Univac-supplied relocatable element G?XX. Refer to Table C—1 for the relationship of printer graphics to other codes used by the remote program.

C.3. CARD OUTPUT

Card output is received by the 9200/9300 Series in XS-3 from the central computer, translated into EBCDIC through the translate table G?XE, processed in EBCDIC by the remote program, then translated into 9200/9300 Series compressed code through the translate table TBP8 and punched by the punch IOCS routine. G?XE is one of two translate tables which comprise the Sperry Univac-supplied relocatable element G?XX. TBP8 is a separate relocatable element supplied by Sperry Univac. Refer to Table C—1 for the relationship of card output codes to other codes used by the remote program.

C.4. CODE RELATIONSHIP

Table C—1 presents the 400/1100 Series character set, in XS-3 collating sequence, with the XS-3 in both octal and hexadecimal representation, followed by the EBCDIC value to which the character is translated by G?XE. In the next column is the hexadecimal representation of the character in 9200/9300 Series compressed code to which the character is translated from EBCDIC by TBP8 and from which the character is translated into XS-3 by G?CX. The following column contains the punched card codes corresponding to the compressed code. These are the standard 400/1100 Series card codes. The last column is the 9200/9300 Series character which is usually associated with the punch combination.

Table C—1. Character, Code, and Punch Translation

400/1100 Character	XS-3 Octal	XS-3 Hex	EBCDIC	Card Codes Hex	Card Codes Punches	9200/9300 Character
b	00	00	40	00	none	b
]	01	01	5F	4A = 3D	11–5–8	)
- (hyphen)	02	02	60	02	11	- (hyphen)
0	03	03	F0	04	0	0
1	04	04	F1	30	1	1
2	05	05	F2	50	2	2
3	06	06	F3	10	3	3
4	07	07	F4	20	4	4
5	10	08	F5	40	5	5
6	11	09	F6	70	6	6
7	12	0A	F7	60	7	7
8	13	0B	F8	08	8	8
9	14	0C	F9	80	9	9
\	15	0D	6D	7C = 3E	0–6–8	>
;	16	0E	5E	7A	11–6–8	;
[	17	0F	4F	49 = 31	12–5–8	(
+	20	10	4E	01 = 33	12	&
:	21	11	7A	48 = 2E	5–8	' (apostrophe)
.	22	12	4B	19	12–3–8	.
?	23	13	6F	05	12–0	none
A	24	14	C1	31	12–1	A
B	25	15	C2	51	12–2	B
C	26	16	C3	11	12–3	C
D	27	17	C4	21	12–4	D
E	30	18	C5	41	12–5	E
F	31	19	C6	71	12–6	F
G	32	1A	C7	61	12–7	G
H	33	1B	C8	09	12–8	H
I	34	1C	C9	81	12–9	I
=	35	1D	7E	18 = 1F	3–8	#
<	36	1E	4C	79 = 10	12–6–8	+
#	37	1F	7B	69 = 00	12–7–8	| KILL
@	40	20	7C	68 = 00	7–8	" KILL
*	41	21	5C	2A	11–4–8	*
$	42	22	5B	1A	11–3–8	$
!	43	23	5A	06	11–0	none
J	44	24	D1	32	11–1	J
K	45	25	D2	52	11–2	K
L	46	26	D3	12	11–3	L
M	47	27	D4	22	11–4	M
N	50	28	D5	42	11–5	N
O	51	29	D6	72	11–6	O
P	52	2A	D7	62	11–7	P
Q	53	2B	D8	0A	11–8	Q
R	54	2C	D9	82	11–9	R
%	55	2D	6C	4C = 00	0–5–8	– (underscore) KILL
' (apostrophe)	56	2E	7D	28 = 20	4–8	@
△	57	2F	6A	6A = 00	11–7–8	¬ KILL
≠	60	30	4A	5C	0–2–8	none
(	61	31	4D	2C = 2D	0–4–8	%
, (comma)	62	32	6B	1C	0–3–8	, (comma)
&	63	33	50	58 = 11	2–8	none
/	64	34	61	34	0–1	/
S	65	35	E2	54	0–2	S
T	66	36	E3	14	0–3	T
U	67	37	E4	24	0–4	U
V	70	38	E5	44	0–5	V
W	71	39	E6	74	0–6	W
X	72	3A	E7	64	0–7	X
Y	73	3B	E8	0C	0–8	Y
Z	74	3C	E9	84	0–9	Z
)	75	3D	5D	29 = 1E	12–4–8	<
>	76	3E	6E	78 = 1D	6–8	=
¤	77	3F	7F	6C	0–7–8	?

NOTE: XS-3 X'30' = EOL
2 OF X'30' = EOD
X'3F' = TAB

NOTE:

When the standard 9200/9300 Series print bar is installed in the bar printer, the following character substitutions occur:

400/1100 Character	9200/9300 Character
}	¬
\	–
|	|
△	smudge
≠	¢
¤	"

Appendix D. User-Supplied Input and Output Routines

D.1. GENERAL

Remote program users may substitute, at their discretion, alternate input and/or output routines whose generations are specified in 4.4 through 4.6. The optional keyword parameters and specifications IN=OWNCD and OUT=OWNCD may be used to indicate the use of alternate card input and/or output routines when generating REM1. If an alternate routine for handling printer output is used, there are no changes required in the REM1 generation.

D.2. CARD INPUT

If IN=OWNCD is specified, REM1 will assume that its input will be in EBCDIC and will consist of images that would make up a normal 1108 run deck. The input file must be assigned the label G?CI; REM1 issues three macro instructions to this file, OPEN, GET, and CLOSE. If the complexity of the input data is such that REM1 should not attempt to reference the actual input file directly, the programmer should be aware that the assembler generates the following calling sequences for the various macro instructions:

1 LABEL	△OPERATION△ 10	16 OPERAND	△
OPEN	BAL	14,filename	
CLOSE	BAL	14,filename+4	
GET	BAL	14,filename+8	
	DC	Y(workarea address)	
PUT	BAL	14,filename+16	
	DC	Y(workarea address)	

If the programmer wished to cause the reading of 1100-byte tape records for transmission to the 1108, the following OWNCD input routine would probably suffice:

```
LABEL      OPERATION      OPERAND                          COMMENTS
1          10             16
INPT       START  0
           USING  *,0
TPIN       DTFMT  DEVA=2,BKSZ=400,RCSZ=100,WORK=YES,ETC.,ETC.
           HIDLR  DMTIO1 WORK=YES,ETC.,ETC.
           ENTRY  G?CI
G?CI       BC     15,OPEN
           BC     15,CLOS
GET        BC     0,PAR2                    BC15 IF 2ND HALF OF RECORD
           STH    14,GETX+2                 SAVE RETURN
           GET    TPIN,W100                 READ TAPE RECORD
           CLI    W100,C'@'                 1108 CONTROL CARD ?
           BC     8,*+8                     SKIP NEXT LINE IF SO
           MVI    GET+11,X'F0'              SET PART-TWO SWITCH
           LH     14,GETX+2
           LH     15,0(1,14)                LOCATE REM1'S WORKAREA
           MVC    0(80,15),W100             MOVE 1ST 80 BYTES OF TAPE RECORD
           AI     GETX+2,2                  ADJUST RETURN
GETX       BC     15,0                      RETURN
PAR2       MVI    GET+11,0                  RESET PART-2 SWITCH
           LH     15,0(1,14)                FIND REM1 WORKAREA
           MVC    0(120,15),W100+80         MOVE REMAINDER OF TAPE RECORD
           MVI    120(15,C' '               BLANK-FILL REST
           MVC    121(59,15),120(15)        OF REM1 WORKAREA
           BC     15,2(1,14)                RETURN
OPEN       STH    14,*+10
           OPEN   TPIN                      OPEN TAPE FILE
           BC     15,0
CLOS       STH    14,*+10
           CLOSE  TPIN                      CLOSE TAPE FILE
           BC     15,0
W100       DS     CL100                     TAPE WORKAREA
           END
```

If an actual 1108 run deck is being read from tape, REM1 will never cause the tape IOCS' EOFA routine to be entered. A halt or error recovery routine could be placed at the label which is assigned to the EOFA parameter. The OWNCD routine must store and reload any registers other than 14 and 15 which it uses.

NOTE:

If IN=OWNCD is specified, the TBP8 translate table must be linked to the remote program, regardless of how the punched output is being handled.

D.3. CARD OUTPUT

If OUT=OWNCD is specified, REM1 will assume that its output will be in EBCDIC and will deliver those images which are transmitted to it by the 1108 for punching. The output file must be assigned the label G?CO; REM1 issues three macro instructions to the output file, OPEN, PUT, and CLOSE. See the input own-code example (D.2) for typical coding where the complexity of the data being transferred does not permit REM1 to reference the actual I/O routine. If the only deviation from normal card output is to be translation to other card codes, this may be accomplished by substituting the desired EBCDIC-to-compressed translate table for TBP8 and by equating TBP8 to the name of the user-supplied table in the linker run.

Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: _____

Manual Title: _____

UP No: _____ Revision No: _____ Update: _____

Name of User: _____

Address of User: _____

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SPERRY UNIVAC

P.O. BOX 500
BLUE BELL, PA.
19422

ATTN: SYSTEMS PUBLICATIONS DEPT.

FOLD

CUT