XDS 901808A
$1.25

DIAGNOSTIC PROGRAM MANUAL

SIGMA 5 AND 7
SUPER SWAP (102)

PROGRAM NO. 706146-A

July 1971

Prepared by
Field Engineering Publications

Xerox Data Systems 701 So. Aviation Blvd., El Segundo, Calif. 90245, 213 679-4511

XEROX

TABLE OF CONTENTS

TABLE OF CONTENTS CONT.

SECTION 1
GENERAL DESCRIPTION

SWAP was designed to detect system failures and as far as possible to isolate that failure to a specific hardware function within the Sigma 5/Sigma 7 System. Its prime ability is to detect intermittent or solid failures that tend to occur only when a large portion of the system is running with a high throughput rate. Further additions to SWAP D00, 101, and 102 allow the program to be run on any Sigma 5, 6, 7, 8, or 9 system plus the ability to run selected diagnostics either alone or in a timeshare environment (FOREGROUND) with SWAP (BACKGROUND) and through the use of the partitioning codes to reduce the failing loop to its lowest limits for troubleshooting purposes.

SWAP consists of four automatic passes, each of which are broken into five phases. The four passes differ only in blocksize and are approximately 15 minutes in length. Unlike the System Exerciser, SWAP retains the same source device and data mode unless otherwise specified under the Test Strategy parameter. The five phases allow the running of all devices, including the source device, in all combinations of READ, WRITE, and TOGGLE. During this period, when the devices are running at maximum throughput rate, the CPU acts as a data checker checking unused buffers on a percentage basis.

SECTION 2
PASSES

SWAP is divided into four automatic passes of approximately 15 minutes duration each, plus the operator has the ability to define a manual pass with any blocksize and data mode desired. The four passes and their blocksizes are as follows:

PASS 0 X'400' Bytes

PASS 1 X'C01' Bytes

PASS 2 X'5' Bytes

PASS 3 X'3FB' Bytes

The data for PASS 0 is memory word bound, all other auto passes are byte bound.

SECTION 3
PHASES

Each pass is divided into five phases which control the direction of data movement for all devices. The PHASES and their functions are as follows:

PHASE 0 - Sourcing Phase - During this phase the CPU will generate a data buffer in the first available block of core. The buffer is then written on the source device which is the first entry under HERE in the TEST STRATEGY or the device specified under SOURCE (must be RAD, MT, or DP). Upon completion of writing the buffer, a new buffer is generated in the same core location and written to the source. It takes two passes through the source routine to write the surface of the RAD. This was done to reduce latency as much as possible. Upon completion of the source routine, the CPU clears core and tags all block ending locations as CPU buffers. At this point, the CPU switches to a Data Checker, the source device is locked to Read only, and all bi-directional devices are written. As soon as RAD 2 is completely written, the program enters Phase 1.

PHASE 1 - During this phase all devices are locked to Read only.

PHASE 2 - During this phase, the source device (normally RAD 1) remains in a Read only, while RAD 2, MT 1, and MT 2 are toggled (i.e., WRITE and READ).

PHASE 3 - The source device is toggled while RAD 2, MT 1, and MT 2 are Read only.

PHASE 4 - All RAD's and MT's are toggled. All undirectional devices (CR, LP, CP, etc.) are exercised during all PHASES.

SECTION 4
AUTO COUNTERS

Seven (7) Auto Counters, C1 through C7, are used to control the switching of PHASES and PASSES. C1 through C3 control the PHASES while C4 through C7 control the PASSES. The counters are under the control of COUNTER 4 Count pulse interrupts and will all decrement on each clock-tick approximately every 130 ms. Counter control of phases and passes is done in the following manner:

During sourcing (PHASE 0), the counters are set to their initial values; however, the clock is inhibited until PHASE 0 is completed (RAD 2 completely written). At this time, the clocks are enabled and all counters begin decrementing. When C1 goes negative, PHASE 2 runs until C2 goes negative, moving the program to PHASE 3. When C3 goes negative, PHASE 4 is entered and run until C5 goes negative. At this time, C1 through C3 are restored to their initial values, a new blocksize is picked for PASS 1, and PHASE 0 starts again.

The initial counter values and pass/phase control are as follows:

Succeeding lines are the values at the end of each phase.

				PASS 3	PASS 0	PASS 1	PASS 2
PASS 0	C1	C2	C3	C4	C5	C6	C7
PHASE 0	600	C00	1200	6000	1800	3000	4800
PHASE 1	0	600	C00	5A00	1200	2A00	4200
PHASE 2	NEG.	0	600	5400	C00	2400	3C00
PHASE 3	NEG.	NEG.	0	4E00	600	1E00	3600
PHASE 4	NEG.	NEG.	NEG.	4800	1	1800	3000
PASS 1							
PHASE 0	600	C00	1200	4800	NEG.	1800	3000
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
PASS 3							
PHASE 2	NEG.	0	600	C00	NEG.	NEG.	NEG.
PHASE 3	NEG.	NEG.	0	600	NEG.	NEG.	NEG.
PHASE 4	NEG.	NEG.	NEG.	0	NEG.	NEG.	NEG.

At the end of PASS 3, SWAP types out PASS COMPLETED. If the program is allowed to continue, all counters will be reinitialized and PASS 0 PHASE 0 started again. The second program pass will differ in that a new Data Mode is selected, normally INCREMENTAL since RANDOM is usually requested for the first program pass.

SECTION 5
DATA BUFFERS AND MODES

All data buffers used in SWAP during Random Data Mode have the same format, the only variable being the blocksize. Each buffer consists of two reflection buffers and a tag byte. SWAP's definition of a reflection buffer is that the second half of the block is identical to the first half of the block. To generate a buffer, the CPU subtracts one from the blocksize and divides the result by two (2). The quotient defines the size of the two reflection buffers. The CPU will generate the data for the reflection buffer and then store into both buffers. Whenever the blocksize is even, there will be a remainder. The remaining byte will be the same as the first data byte and will be placed in the next-to-last byte of the buffer. It should be noted that in cases of data errors the RIT BYTE (right) and ERR BYTE (error) will be displaced by one byte location in their respective words.

The one byte initially subtracted will become the TAG byte which is located at the last byte location of the buffer. The TAG bytes are signed by each device when the device has finished with the buffer.

When using Incremental or Controlled Data, the buffer is filled with the desired pattern and no reflection is generated, however, the last byte location is still retained as the TAG byte.

Under SWAP, there are three data modes which are selected by the CONTROL WORD of TEST STRATEGY. The selection bits are random DATA or CONTROLLED DATA: lack of both bits implies INCREMENTAL DATA.

CONTROLLED DATA - When this mode is requested, the operator must provide the data information to the SEED parameter of TEST STRATEGY. Under SEED, the first byte is the pattern seed and the second byte is the pattern increment. The first buffer is filled with the pattern seed. The second buffer will be filled with the value of the seed + increment, the third buffer with the value of the seed + 2X increment, etc. If no values are loaded into SEED, the parameter is defaulted to a pattern of 00 and an increment of 01.

INCREMENTAL DATA - In this mode, SWAP will pick an initial value which will become the seed and increment. The initial value will be the first byte of the first buffer. The second byte will be the first byte incremented by itself, the third byte will be the second byte + the increment byte (first byte), etc.

The second buffer generated will be the first buffer left shifted by one byte position. The third buffer generated will use the increment + 1 as a seed and increment. The fourth buffer will be the third left, shifted one byte position, etc. (i.e., with an initial byte of 09, the first word of the first five buffers would be:

Buffer 1 09 12 1B 24

Buffer 2 12 1B 24 2D

Buffer 3 0A 14 1E 28

Buffer 4 14 1E 28 32

Buffer 5 0B 16 2C 37

SECTION 6
PROGRAM INTERFACE

All of the directives and control information used by SWAP are contained in the eight TITLES of ABSTRACT. Two methods are available for the operator to get the ABSTRACT listing. It will be typed on the TTY if the operator depresses ? at the first opportunity for input. A listing can also be made on the LP by calling ABSTRACT in with the foreground diagnostics which are loaded onto RAD 1.

TITLE ONE contains a listing of all of the directives used by SWAP including the Remote Terminal controls. TITLES two through six describe the insertion of control parameters and the running Foreground programs. TITLE SEVEN explains the method for updating or adding Foreground programs to SWAP, and TITLE EIGHT is an explanation of the Data/Parity error formats.

The rest of this section is a partial description of the ABSTRACT, what the directives accomplish, and the controls available to the user.

TITLE ONE - DIRECTIVES

All directives are entered via the keyboard. To get keyboard control when running the program locally, depress CPU INT. All directives must be terminated with a T and in some cases those requiring inputs must be terminated by two (2) T's; one to indicate end of input, and the second to terminate keyboard control. When running remotely, the terminal is always available for input and no interrupts are necessary except to ALTER TEST STRATEGY. Some directives must still be followed by a T, such as the M directive and insertion of control parameters. All inputs must be followed by a CTRL W (See Remote Special Characters).

G - BRANCH

Used to branch within the program to user loops in the patch areas. SWAP will still honor interrupt requests while running the user routine and will return to the routine after processing the interrupt.

Example: 10*G causes the program to branch to location X'10'

H - ADDRESS STOP or HEX DIGITS

H is a toggle code (sets/resets) for the programmed address stop. When the desired address is reached, the program will enter a WAIT loop, type the address and the contents of the General Registers. To continue SWAP, type T. To remove an Address Stop, type H. Only one address stop may be set at a time.

H is also used throughout the ABSTRACT as a Hex character indicator.

I - I/O INSTRUCTIONS

All I/O instructions are available to the user for execution at any time and will return CC1 and CC2, and the contents of registers R and Rul. When using the SIO instruction, the user must preload the CDW's in the patch area and the CDW address into location 4F. Locations X;10-25' and X'30-3F' are available for CDW's or program loops (Refer to Branch). To load the CDW's and location X'4F', use the SET LOCATION COUNTER's directive.

```
Example:  10*03000050.20000002.04000054.00000004.01000000.
          4F*8.
          1F0*4I
          15*1P
```

The example above would load locations X'10-13' with a SEEK/CC/SENSE to BAND 2 of the HS RAD. The SEEK data is in location X'14' and the SENSE information is returned to X'15'. 15*1P will print the SENSE information on the TTY.

J - SENSE SWITCHES

J is also a toggle code used to set and reset the Sense Switches. Two (2) types of sense switches are available - swap software and foreground Diagnostics.

The SWAP Software SSW4 is set by J. This inhibits all recoverable I/O error printouts and report only catastrophic failures or in the case of the RAD's more than one failure on the same track and sector. It is normally used to suppress MT error reports or acceptable Data overruns such as running both a HS and an EP RAD in the same memory door continuously.

The foreground SSW's are set by preceding the J with a hex character from 1-F. This digit represents the binary weight of the switch(es) being set. SSW=8, SSW2=4, SSW3=2, SSW4=1, SSW's 1 & 3=A. To clear the foreground SSW's use 10J.

L - OUTPUT THE ERROR LOGS

Outputs a numerical summary of the I/O errors that have occurred on the system while running SWAP.

M - OUTPUT BUFFER ADDRESSES

The M directive causes SWAP to output the current buffer addresses on two lines using the memory log format - the first five characters are the word ADDRESS and the sixth character is the byte position. At the end of each line the keyboard is selected for input to allow the operator to reassign buffers. The input format is BYTE ADDRESS. If you desire to start a buffer in word location X'10000' type in 040000 followed by a period (.). Addresses that are not to be changed should have a comma (,) typed under them which causes SWAP to space to the next address. If no changes are desired or all changes to a line have been made type T. After terminating the second line a third T is necessary to terminate keyboard input control, if so desired.

N - CONSOLE INTERRUPT TO FOREGROUND DIAGNOSTIC

All foreground diagnostics are run under SWAP interrupt control. The N directive is used to simulate CPU INT for foreground. To gain access to a diagnostic it is necessary to depress CPU INT. When the keyboard is selected for input, type the bit mask for the diagnostic desired to interrupt followed by N, then T.

Example: 08000000NT

The next time 5th diagnostic receives its time slice, SWAP will print foreground input, at which time the keyboard is again selected for inputting parameters to the foreground diagnostic being run.

When running only one foreground diagnostic the bit mask preceding the N is not required.

O - PRINT 7 AUTO COUNTERS

The seven (7) Auto Counters (See description in General Information) are used to control the phases and passes whenever running under Automatic Data Modes. Typing O whenever the keyboard is available for input causes the contents of the counters to be printed. After the counters have been printed type T to continue SWAP.

U - LOAD SUPER SWAP DIAGNOSTIC TAPE

Typing U causes SWAP to list on the TTY all of the Foreground Diagnostics available to the revision level being run. The listing is followed by a printout of DIAGNOSTICS WANTED and a 32 bit (8 hex digits) mask and the keyboard is enabled for input. At this point the operator inserts a hex representation of the diagnostics desired. If you desired to load the first, third, fifth, sixth, and seventh diagnostics, you would type in AE000000T. Only the diagnostics selected are read in and loaded sequentially on the SWAPPER RAD (RAD1) starting at the track and sector specified in STK1.

Whenever running foreground STK1 (starting track RAD1) is automatically set above the foreground storage area and should not be changed by the operator unless foreground is no longer desired in which case the operator may set STK1=0 and must set the foreground mode Q=0.

Q - SET THE FOREGROUND PARAMETERS

After foreground diagnostics have been loaded by U it is necessary to type Q to define which diagnostic(s) is to be run and the conditions (mode) under which to run it. The four modes are:

8 = FOREGROUND ONLY (swap defaults to this mode)

2 = SHARE SWAP AND ONE DIAGNOSTIC

1 = SLICE (swap and all diagnostics called)

0 = STOP FOREGROUND

3 = LOAD FROM TAPE AND ENTER FOREGROUND ONLY

4 = LOAD FROM TAPE AND SHARE SWAP AND ONE DIAGNOSTIC

When Q is typed SWAP will print MODE then 8 and enable keyboard input. The operator then enters the MODE number desired followed by a period (.). SWAP then prints out DIAGNOSTIC BIT MASK followed by eight 0's and the keyboard enabled as an input. At this time the operator enters a hex representation of which diagnostics loaded by the U directive he desires to run.

Two (2) additional modes are possible which do not use a RAD for diagnostics followed by a printout of DIAGNOSTIC BIT MASK and a 32 bit (8 hex digit) mask. The keyboard will then be selected for input at which time the operator types in the hex representation of the diagnostic desired.

R - EXIT CONTROL AND CLEAR CORE

This directive may be input at any time during the control period - insertion of DEVICE ADDRESSES, SYSTEM PARAMETERS, or DEVICE PARAMETER. The R causes SWAP to exit the control, clear core (unless partions

have been set) and type out INTERRUPT TO ALTER TEST STRATEGY. This directive may be used at any message input to clear input mode - core will be cleared but I/O will not be disturbed. The parameters skipped will assume normal defaults unless previously set to new values, in which case the new values are retained. In the automatic mode strategy will be altered by the program in order to reflect the state of operation of the I/O devices.

S - SCAN FOR MEMORY PARITY ERROR

This causes SWAP to SCAN memory for parity errors starting with location zero of CORE to the ending address entered under the parameter LAST plus the size of the block entered under parameter BLOCK.

After the error message is typed out SWAP is reset to the INPUT DEVICE ADDRESS.

T - TERMINATE INPUT AND EXIT CONTROL

Typing T causes SWAP to terminate keyboard input and continue in program sequence. All inputs must be terminated by a T or R.

@ - MODIFY DEVICES SELECTED

Typing C800@ would change the devices selected under HERE in the test strategy input loop. This allows starting or stopping the devices selected without restarting at the test strategy input loop.

Also this permits starting up any new device. There is a 30 second delay in starting or stopping devices except with RAD's and MAG TAPES which will start at PASS times.

W - WORD ALIGN

The W directive converts a byte address to a word address. Type in the byte address followed by a W.

Example: 14ACW=532B

X - EXPLAIN THE CURRENT OUTPUT OR LOCAL INPUT

The X directive is used for two functions. The first is to explain the current output. When typed after any output SWAP will refer the operator to the applicable TITLE. The second function is used when running locally to enter the INPUT TEST STRATEGY loop. When the CPU INT is depressed after INTERRUPT TO ALTER TEST STRATEGY SWAP will type LOCAL INPUT and set the location counters to X'10'. Sixteen words may now be entered in locations X'10-1F' by typing in each word followed by a period (.). If nothing is to be input or the input is completed, type T to terminate the input or an X to get to test strategy input.

Y - LIST ALL I/O ADDRESSES

When Y is typed SWAP will issue TIO's to all IOP/DEVICE addresses from 000 to 7FF and list on the TTY all addresses that are recognized. In the case of HS RAD's a dual address will be listed. If a system has a HS of 2F0 SWAP will list 2F0 and 2F4.

: & ! - SET WORD AND MASS STORE

These two directives are used in conjunction to generate user defined buffers. The colon (:) is used to define the data word and the exclamation point (!) to define the inclusive areas.

Example: A1B2C3D4:4000*40FF! would store a 256 word buffer from locations X'4000-40FF' consisting of the pattern A1B2C3D4.

* SET LOCATION COUNTERS

The asterick (*) is used to set the location counter. Once the location counter has been set the user has the ability to store or display an unlimited number of successive words starting at the location defined by *.

. STORE

Unlike the : & ! which were used for generating large buffers of the same data the * & . are used in conjunction for loading patches, CDW's, user loops, or altering the SWAP sequence. The operator types in the desired address followed by an asterick (*). He then types in the address contents and follows with a period (.) which stores the word and increments the location counter by one.

Example: 1D05*00AB0000.00CA1301.00AB0000.00CA1301.

The example given would be used to set locations 1D05 & 1D07 to 00AB0000, and 1D06 and 1D08 to 00CA1301. The value 00AB0000 is Cylinder 171 for DP's and locations 1D05 and 1D07 are STTR2 (RAD2) in SWAP 101. Locations 1D06 and 1D08 are ENTR1 (end track RAD1) and ENTR2 (RAD2) in SWAP 101 and 00CA1301 is Cylinder 202 Head 19 Section 1. These parameters would limit DP operations to the last 31 cylinders if desired.

, - STEP

The comma (,) is used to step the location counter without effecting the contents of the location. It is used primarily when inserting control parameters to space properly across the control line.

Example: 1D06*CA1301.,CA1301. would store CA1301 in locations 1D06 and 1D08 and leave 1D07 undisturbed.

/ - IGNORE

The slash(/) is used to correct improper inputs and causes SWAP to ignore all hex characters to the previous asterick (*), period (.) or comma (,).

Example: 108C*237/22701079. 22701079 would be stored in 108C.

+ & - ARITHMETIC OPERATORS

Only add (+) and subtract (-) operations may be performed under SWAP. Only two operands may be added or subtracted. Both arithmetic functions assume the same format.

Operand*Operand ± Resultant

K - PROGRAM PARTIONING

Program partioning is primarily a troubleshooting aid which allows the user to freeze the system or selected devices to specific data blocks, eliminate message headings or complete messages, to lock the RAD(s) to a failing track, etc. This feature when used eliminates the need in many cases of having to call on other diagnostics or the loading of user test loops. The K directive is also a toggle directive for setting and reseting partitioning.

SECTION 7
PARTITIONING CODES

The following is a partial list of codes and their function. Refer to ABSTRACT TITLE ONE for the complete list.

07 - DOUBLES FOREGROUND OPERATING TIME QUANTUM

The normal operating ratio of a foreground diagnostic to SWAP is 1:5. 07K sets the ratio to 2:5.

33 - INHIBITS RESETTING CORE STARTING ADDRESS ON RESET

The core starting address for buffers (FIRST under SYSTEM PARAMETERS) is reset to 10000 when a System Reset is done. 33K inhibits the reset and is useful whenever the SYSTEM PARAMETERS have been changed to test a specific core area.

35 - REDUCES PASS TIME BY A THIRD

The auto counters are reduced by 1/3 cutting the pass time to approximately 10 minutes. 35K sets C1-C7 to:

C1	C2	C3	C4	C5	C6	C7
400	800	C00	4000	1000	2000	3000

16, 2B, 3A, 41, 43, 48, 44, 46, 4A, 3E - GET BLOCK INHIBITS

GET BLOCK INHIBITS prevent a device(s) from getting a new buffer thus freezing the device(s) to the same buffer continuously. This is particularly useful on Memory Port problems or in cases where it desired to use the same data on a device continuously such as doing RAD or MT data path checks or adjustments. Use of the M directive allows setting of buffer addresses desired. 2B is a system freeze code. Ref: Partioning code 13K.

15, 39, 3D, 40, 42, 47, 45, 49 - BLOCK TAGGING BYPASS

Block tagging is done by each device as it finishes using a buffer. This is SWAP's method of determining which device had the buffer in which a data error is detected. In some cases it is desirable to generate a buffer with a sync byte(s) by using the mass store and block tags are not wanted in the buffer. It should be noted that other devices whose block tags have not been bypassed can have access to the buffer and either destroy or tag it unless the appropriate GET BLOCK INHIBITS are set.

11 - BYPASS CLEAR MEMORY ON RESET

Whenever a SYSTEM RESET is done SWAP returns to the INPUT DEVICE ADDRESSES parameter and core will be cleared after insertion of DEVICE PARAMETERS or typing an R. In cases where the user has used the Mass Store to generate a buffer and the GET BLOCK INHIBITS to freeze a device(s) to a specific block it would be undesirable to clear core or Reset. An 11K will inhibit the clearing. Also in cases of Data Errors it may be desirable to terminate I/O with a System Reset but allow the data checker to continue.

On local, this can be accomplished by the following:

a. Hitting the I/O reset and type a T or R.
b. Hitting a system reset and clearing the input parameters with a T or R.
c. By using a XXX*2I and clearing the input parameters with a T or R, this allows you to HIO individual devices.

On remote, this can be accomplished by the following:

a. Hitting two rub characters to simulate a system reset and then clearing the input parameter with a T or R.

13 - BYPASS CLEAR BLOCK ADDRESS REGISTERS ON RESET

13K should be used in conjunction with any GET BLOCK INHIBIT. After a Reset the BLOCK ADDRESS REGS are set to all F's. NOTE: After a Reset, if a GET BLOCK INHIBIT is used without a 13K, memory errors will occur.

0B, 0C, 0D, 0E, 10, 20, 25, 26, 30, 17, 19, 18, 2F, 2E, 00, 29, 38, 01, 2A, 3C, 02, 40, 03, 04, 05, 06, 08, 0F - MESSAGE INHIBTS

These codes allow the operator to eliminate message headings or if desired the complete message. In the cases of 0C, 0E, & 10 the Device Address, System Parameter and Device Parameter inputs are eliminated so that the operator doesn't need to reinsert them after each SYSTEM RESET.

1B - ALWAYS RUN BOMB

Bomb is a memory testing routine that is run on every other clock tick concurrently with the data checker. The normal sequence is get a block, data check it, or get a block, data check it, run bomb on it. In the automatic mode 1BK causes bomb to use the following sequence on every clock tick; get a block, data check it, run the bomb.

24 - ALWAYS RUN BOMB ON THE SAME MEMORY BLOCK

Bomb is locked to the CPU buffer and will continuously run until the directive is reset. The sequence is get a block, data check it, and then run the bomb - the bomb is frozen on current block.

1C-23 - ALWAYS INTERRUPT ON PARITY DURING STB (IN BOMB)

Bomb uses eight store byte instructions which might alter the contents of a memory parity location before the error could be printed. These codes allow the error to be printed before the STB effects the location.

14 - BYPASS DATA CHECKING BY BRANCHING TO X'10'

This code allows the user to load X'10' and successive cells with his own routine to check for some specific condition such as a memory location be changed at the wrong time. This routine will entered by a Branch whenever the Data Checker is called. Once the routine is entered the program will remain there except to process interrupts after which the program will return to the routine.

37 - BYPASS BYTE GENERATION DURING SOURCE

This code allows the user to generate a source buffer and store it with the Mass Store Routine for writing to the source. This would be the only data used during the program. The buffer must be stored in the first block location and must be of the size defined by the SYSTEM PARAMETERS. This is the starting core address under the parameter FIRST plus the block size under parameter BLOCK.

28 - BYPASS MAX ERROR TEST

SWAP will abort if more than 512 errors occur on a device. This code inhibits the error message and the abort.

27 - BYPASS RAD TRACK AND SECTOR INCREMENTS

This code locks both RAD1 & RAD2 to the same track and sector(s) (the number of sectors depends on the BLOCK SIZE). 3B & 3F limit the by-pass to either RAD1 or RAD 2 only.

31 - BYPASS TRACK AND SECTOR INCREMENT DURING WRITING SOURCES

This code locks the program into the sourcing routine continuously. The sourcing routine uses TIO's instead of INTERRUPTS to determine when a block has been written. This also eliminates all of the error messages except the MAXIMUM ALLOWABLE ERRORS EXCEEDED which can be inhibited with a 28K.

09, 0A - CHECK WRITE DATA TRANSFERS

SWAP does not use the check write order for RAD's. If a check write is needed insert these codes for RAD1 or RAD2 respectively.

4B - SETS CARD PUNCH CDW TO NO COMMAND CHAINING

This code must be inserted if running an Uptime CP Model 7165. Allows SWAP to use the same CP handler for both the Univac and Uptime CP's.

SECTION 8
REMOTE OPERATION

GENERAL

All directives and controls previously described are available for remote usage plus special characters have been added to simulate PCP functions. Since SWAP is designed to run Remote, the keyboard is always available for input once the user has logged on and there is no need to simulate the CPU INT. All operations are the same as when running locally except for the use of CTRL W following all inputs. See Special Remote Characters.

To run remote it is necessary for the on-site C.E. to load SWAP and insert the correct remote information. After loading and the title has been printed, SWAP will printout:

REMOTE ADDRESSING PARAMETERS

COC	W/D	INT
1000	0000	0060

The C.E. inserts the correct four bit COC address followed by a period (.). The defaults for W/D and INT addresses are normally correct and should not need changing in which case the C.E. types T after the COC address. If a change is necessary put in the correct four bit address (W/D range is 0-F and INT range is 60-7F) followed by a period (.). When the line is terminated with a T, SWAP will printout REMOTE TERMINAL HUNG UP and go into a wait loop until the remote operator calls in.

LOCAL RECOVERY

Should the on-site C.E. want to run locally at any time he can do a System Reset which causes SWAP to return to the INPUT DEVICE ADDRESSES loop. Inserting 1000 under COC address will give him local control again.

SECTION 9
SPECIAL REMOTE CHARACTERS

ZZ- PROMPT SWAP FOR A PASSWORD REQUEST

After the remote operator has called and receibed the COC carrier, he types in ZZ. SWAP will print PASSWORD PLEASE. The remote operator may enter any four characters as a password. When the password is accepted, SWAP prints OK COC ON and initializes to the INPUT DEVICE ADDRESSES loop. Once logged on the remote operator may hang up at any time. SWAP will continue where the operator left off in the control lines as soon as he redials and logs on again with the same password or if SWAP is already running it will continue until it has a message to print at which point SWAP enters a wait phase. As soon as the remote operator logs in again SWAP will print the message.

X-ON - INTERRUPT TO ALTER TEST STRATEGY

To simulate a CPU INT hold down CTRL and type Q (X-ON) followed by a CTRL and W. SWAP will print out INPUT TEST STRATEGY.

X-OFF - MESSAGE INPUT TO LOCAL TTY

SWAP contains an 80 character message buffer which can be transmitted at any time after the INTERRUPT TO ALTER TEST STRATEGY. To enter the message mode at the remote terminal depress CTRL and S(X-OFF) followed by CTRL and W. To enter locally depress CPU INT. To terminate the message from either end, type double slash (//).

BELL - OBSERVER LOG ON & MESSAGE

SWAP has provisions for one remote observer terminal. The observer is just that, he can log on and transfer messages, but has no other functions. To

log on the observer calls in and gets the carrier then types BELL (CTRL & G) followed by CTRL & W. The same method is used to enter the message mode - BELL. To terminate the message mode type double slash (//).

NOTE: If the message buffer is in use by the local C.E. or the controlling terminal no action will occur. If the message buffer is free the response output on the teletype is new line, BELL.

RUBOUT - RESET, RETURN TO INPUT DEVICE ADDRESSES

A System Reset may be simulated by depressing RUB followed by a CTRL & W. A single rub out forces all error messages to be inhibited.

CTRL W

All characters that require stopping of SWAP must be followed by a CTRL W. These characters are: T, R, L, M, S, K, U, G, Q, and N.

ESC - STOP PRINTING (BREAK)

ESC simulates the local TTY BREAK switch and is used to terminate TTY outputs.

SECTION 10
OPERATIONAL CONTROL LINES

There are four control lines which are used by the program. These control lines are used to define the system addresses, core areas available for data buffers, unique parameters for the devices such as the area of the RAD to be tested or the number of tape records, and the actual program test strategy.

INPUT DEVICE ADDRESSES (TITLE THREE)

DEVICE ADDR											
RAD1	RAD2	MT1	MT2	LP1	TTY	CR	CP	COC	LP2	MSG	DIAG
80F0	81F0	8080	8180	0002	0001	0003	0004	1000	0010	0001	1000

This is the initial input loop of SWAP. It is also returned to after System Reset, Rubout and Memory Parity Errors. The user types in a four bit address for each device he desires to run. In the case of RAD, MT and LP you are limited to two of each and one each of the remaining devices. The MSG (Message) device may be changed to another TTY on the system, however, it cannot be made a LP.

All device addresses are entered in hex. Bit 0 of the address is multi-device controller which pertains to RAD's, DP's, and MT's only. Bit 0 has been changed from a one (1) to an eight (8) for Sigma 9 compatability. If it desired to run all devices on a multi-device controller type in 8 followed by the IOP/DEVICE CONTROLLER/DEVICE address and a period (.). If only one device on a multi-device controller is to be tested type in 0 followed by the address and a period (.). If the default address is correct or the correct address has been previously entered type in comma's under those addresses to be left unchanged. Terminate the line with a T and if running remote a CTRL W. If the user does not know

all the system addresses he may type Y to find them. After the addresses have been printed, do a System Reset before entering them.

NOTE: Disc Packs are treated as RAD's and are entered under either RAD1 or RAD2.

SYSTEM PARAMETERS (TITLE FOUR)

```
SYSTEM PARAM.
FIRST LAST  BLOCK MAP
10000 5FBFF 00400 00000
```

The system parameters define the core addresses available for data buffers. All entries are in byte addresses. FIRST is the starting buffer address. It is defaulted to 16K to allow for FOREGROUND running. If the foreground is not going to be used this may be changed to 08000. LAST is the ending address minus the block size. BLOCK is the buffer size in bytes. MAP checks the memory map option on a 1 to 1 basis if any non-zero character is entered. The MAP mode cannot be used if Foreground Diagnostics are to run which will change the MAP configuration, such as, SUFFIX.

If the user desires a block size other than those selected by the Auto Counters he may enter that under BLOCK. Also, should he desire to run on non-word boundaries he may change FIRST to set the desired byte boundary.

DEVICE PARAMETERS (TITLE FIVE)

```
DEVICE PARAM.
SCT1 SCT2 STK1 ETK1 STK2 ETK2 BLC1 BLC2 COCA PON  POFF
0004 0004 0000 07FB 0000 07FB 0400 0400 0000 0200 0100
```

The Device Parameters are controls that are unique to the applied devices. The first six parameters are the RAD increments and maps. BLC1 and BLC2

are the number of record blocks for MT1 and MT2. COCA is the Com Gear Controller Write Direct Address. PON and POFF are the peripheral on and off times for all peripherals except RAD and MT. With the default values of PON and POFF the peripherals will be on and off for approximately 60 seconds and 30 seconds, respectively.

The RAD sector increments will be defaulted to the proper values when running in Automatic Mode. If the blocksize is changed when running in Manual Mode they must be set to a value large enough to hold the block plus 4 bytes. The RAD handler Data Chains an additional 4 bytes onto the block being written or read from the Disc. These 4 bytes contain the SEEK address of the record and are verified each time the RAD is read from.

STK1, STK2, ETK1, and ETK2 are the RAD maps (starting and ending tracks and sectors) for RAD1 and RAD2, respectively. They are defaulted to start at track 0 and end on a quarter file - Track 127 Sector 11. STK1 will be changed to allow for sufficient Foreground Diagnostic storage if the Foreground is called for with the U and Q directives. The following table contains the values pertinent to the three models of RAD's currently produced:

TYPE	MODEL	SECTOR SIZE HEX BYTES	ETK1 OR ETK2 ¼ File	½ File	Full File
Medium speed	7201,2,3 & 4	168	7FB	FFB	1FFB
Extended capacity	7231 & 32	400	7FB	FFB	1FFB
High speed	7211 & 12	400	7D1	FD1	1FD1
Disc pack	7240,42 & 46	400	N/A	N/A	CA1301

Even though there is only a four digit space allocated for STK and ETK, a larger group of characters may be input. (e.g. The ending track address necessary to run a full disc pack is CA1301, and may be entered in the following manner:

```
SCT1   SCT2   STK1   ETK1    STK2   ETK2...
0004   0004   0000   07FB    0000   07FB...
,      ,      ,      CA1301.,       1FFB.
```

The printout appears to be shifted, but this has no effect on the program).

NOTE: On subsequent type outs, only the bottom 4 hex digits will be typed out. (e.g. ETK1)
(1301)

SWAP has no provisions for handling flaw marks on the packs. If a flawed pack is to be run, set the maps to test only unflawed areas or insert an unflawed pack while running SWAP. Also sourcing of packs takes almost four minutes per pack and unless a specific problem calls for running the whole pack only the last 30 or so cylinders need be checked.

TEST STRATEGY (TITLE TWO)

TEST STRATEGY

HERE	R/W	STRT	TGLE	CONT	SORC	SEED
0000	0000	0000	0000	0000	0000	0001

The TEST STRATEGY parameters control the complete testing of the system within the limits set by the DEVICE ADDRESSES, SYSTEM PARAMETERS, and DEVICE PARAMETERS. The Test Strategy defines which devices are to be tested, the method used (read or write or toggle), the data type to be used and finally the mode of operation (Manual or Automatic).

The 16 bit mask for each parameter has the following format:

For HERE, R/W, STRT, TGLE, and SORC

Bit 0 = RAD1
Bit 1 = RAD2
Bit 2 = MT1
Bit 3 = MT2
Bit 4 = LP1
Bit 5 = TTY
Bit 6 = CR
Bit 7 = CP
Bit 9 = LP2
Bit 11 = MSG

Ex. If the user desired to run RAD1, RAD2, MT1, LP1, TTY, CR, and CP in the Automatic Mode he would enter EF00. under HERE. No entry would be necessary for R/W, STRT, TGLE, or SORC in Auto.

NOTE: For SORC only RAD1, RAD2, MT1, or MT2 may be selected as Source devices.

The 16 bit mask for CONT is as follows:

Bit 0 = HI DENSITY RAD1 (an Extended Capacity RAD Model 7231/32 is addressed as RAD1).

Bit 1 = HI DENSITY RAD2

Bit 2 = HI SPEED RAD1 (Device Addressed as RAD1 is a Model 7211/12 High Speed RAD).

Bit 3 = HI SPEED RAD2

NOTE: No bits would be entered for RAD1 or RAD 2 if they are Medium Speed Models 7201/02/03/04

Bit 4 - DISC PACK RAD1 (Device addressed as RAD1 is a Model 7240/42/46 Disc Pack).

Bit 5 = DISC PACK RAD2

Bit 6 = DUAL CONTROLLER DISC PACK RAD1

Bit 7 = DUAL CONTROLLER DISC PACK RAD2

Bits 6 & 7 are used when the Pooling Feature (7243 Option) is installed which allows two controllers to access one pack. If the option is installed and to be tested assign Controller 1 address to RAD1 and Controller 2 address to RAD2 under device address and set bits 4 thru 7.

Bit 8 = 4BYTE EP RAD ON MIOP, RAD1

Bit 9 = 4BYTE EP RAD ON MIOP, RAD2

Bits 8 & 9 refer to the Model 7231 EP RAD being placed on a 4 BYTE IOP. This combination does not work on byte boundaries and with either of the bits set SWAP will run all RAD data transfers on word boundaries. If the EP RAD is a Model 7231-3 (identifiable by a Maintenance Panel on the Controller) the bits should not be set as the 7231-3 allows data transfers from byte boundaries.

Bit 10 = Not Used

Bit 11 = Not Used

Bit 12 = CONTROLLED DATA MODE

In this mode the SEED parameter provides the pattern and increment.

Bit 13 = RANDOM DATA MODE

INCREMENTAL DATA MODE is implied if bits 12 and 13 are zero. Refer to DATA BUFFERS and MODES in GENERAL INFORMATION.

Bit 14 = RUN THE BOMB

Bomb is a memory test routine run on the data buffers in conjunction with the Data Checker.

Bit 15 = AUTOMATIC OPERATION

The Auto counters control the pass. Bit 15 reset implies MANUAL OPERATION.

PARAMETER DEFINITION

HERE - All devices selected by a one will be run. All devices selected by a zero will be issued an HIO.

R/W - All devices selected by a one will be written. Those selected by a zero will be read. These bits are controlled by the program in Auto Mode. In Manual Mode the user must define which operation is to take place on each device. A bi-directional device must have been previously written before it can be read.

STRT - Devices slected by a one will be started in Manual Mode. No entry is necessary in Auto Mode.

TGLE - Devices selected by a one will be toggled. If only one device is selected it will be toggled after each buffer is transferred i.e. Read one buffer write one buffer. If more than one device is selected the rate of toggling for any device will be dependent on the total number of devices, block length, speed of toggling device, etc.

CONT - Defines Data and Operation Modes also unique device options, and SEEK/SENSE FORMATS and Data boundaries for rotating memories.

SORC - Defines which of four devices is to be used as the Source device. The Source device is initially filled with data blocks which then become the Source of all the buffers for the pass. The first device (RAD1, RAD2, MT1, or MT2) entered under HERE will be the SORC in Auto Mode unless otherwise specified by the user. In Manual Mode the user must specify which device is the SORC. More than one device may be selected under SORC in which case all devices selected must finish sourcing before SWAP begins writing to the other devices.

SEED - Defines the pattern and increment for Controlled Data Mode. First byte is the pattern and second byte is the increment. The pattern will be incremented before generation of each new buffer spread to the source. If no input is provided the SEED is defaulted to 0001.

SECTION 11
FOREGROUND CONTROL

In general the control and operation of foreground programs are the same as if they were being used in the stand-alone mode. However, there are some exceptions:

1. The D.C.P. Programs have an extra command (EXIT) in its syntax. This command allows a slice after a command list.
2. Internal CPU diagnostics (auto, suffix, etc.) can not be interrupted.
3. All output and input operations are preceeded by a program designator (80000000). This tells you which Foreground Program is currently running.
4. All Foreground Programs inputs are preceeded by a request (Foreground Input). This indicates that you are talking to the Foreground.
5. An extra carriage return is required to terminate an input to either a D.P.M. or D.C.P. Program.

READER SURVEY

XDS
Xerox Data Systems

Use this postpaid form to communicate any constructive comments you may have regarding this publication. Please be specific.

PUBLICATION

NUMBER	DATE OF PUBLICATION	TITLE	REV.

HOW DID YOU USE THIS PUBLICATION

☐ LEARNING ☐ INSTALLING ☐ MAINTAINING ☐ OPERATING ☐ SALES

☐ OTHER _____

COMMENTS

YOUR NAME AND RETURN ADDRESS

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 229
EL SEGUNDO, CALIF.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

Xerox Data Systems
701 South Aviation Blvd.
El Segundo, California 90245

ATTN: FIELD ENGINEERING PUBLICATIONS

FOLD

CUT ALONG LINE