SDS RAD/TAPE SYSTEM GENERATION
FOR
940 COMPUTER

TECHNICAL NOTES

90 11 54A

April 1967

SDS

SCIENTIFIC DATA SYSTEMS/1649 Seventeenth Street/Santa Monica, California

Printed in U.S.A.

TECHNICAL NOTES

This document describes the method for generating a new RAD/TAPE system for the SDS 940 computer.

First, mount a self-fill version of the system on tape unit 0, making sure that the Write ring has been removed. Next, mount a file tape on unit 1. There should be a Write ring on this tape so that information on the tape can be modified. The tape on unit 1 contains the source files for the entire Monitor and Executive, some of which may have to be updated before making a new system. This update may occur for any one of the following reasons.

An individual installation may want to make corrections or additions to the system.

Changes might be issued by SDS. These changes would be made using the 940 QED program.

SDS might issue a major revision of some parts of the system. In this case, a user would be supplied with a tape containing the new symbolic file(s) necessary to implement the change.

In general, the correction files will be identified by the user name $S^cY^cS^c$ [t]. The files on this tape would be described in a supplementary information sheet accompanying the tape. To use these files it is necessary to do a "1ENTER $S^cY^cS^c$" sequence and then copy them onto the RAD. When this is done the regular file tape can be mounted on unit 1. These files could then be written onto the master file tape under the appropriate user name(s).

To describe this procedure in greater detail, assume that two packages in the Monitor were completely revised. These packages are SMDBG and SPAC. The following sequence will put these revised files onto the regular file tape.

Note: The sequence of commands given is exactly as seen on the teletype listing. However, the user need not type all the characters, since command and file name completion, as well as the typing of the @ sign, is done by the Executive.

Mount the tape containing revised packages on unit 1.

Obtain appropriate file directory with the following instruction:

@1ENTER $S^cY^cS^c$.

Put the two files onto the RAD with the following instructions:

@COPY 'SMDBG' TO /SMDGB/.

@COPY 'SPAC' TO /SPAC/.

---

[t] Throughout this document, the superscript c designates a non-printing control character obtained by depressing the CONTROL key and then the indicated character key.

Mount the master file tape on unit 1 and 1ENTER for the appropriate user.

Put the files on the master tape with the following instructions:

@COPY /SMDBG/ TO 'SMDBG' OLD FILE.

@COPY /SPAC/ TO 'SPAC' OLD FILE.

If an individual installation had some patches or additions of their own that they wanted to make in these packages, the following alternative sequence could have been used.

Mount the tape containing revised packages on unit 1.

Transfer the two files to the RAD with the following instructions:

@COPY 'SMDBG' TO /SMDBG/.

@COPY 'SPAC' TO /SPAC/.

Mount the master file tape on unit 1.

@QED.

When QED is ready to accept a command an asterisk is typed.

Type the following instructions:

```
*READ FROM /SMDBG/.
   :                          } Edit of SMDBG
*WRITE ON 'SMDBG' OLD FILE    }
*1,$DELETE.
*READ FROM /SPAC/.
   :                          } Edit of SPAC (R = RUBOUT)
*WRITE ON 'SPAC' OLD FILE.    }
* ®
* ®
@
```

After all corrections are made, the system is ready to be assembled. Assuming that major changes are being made to both the Monitor and Executive, the next procedure is to copy three files from tape to RAD. These files are under user name $S^CY^CS^C$.

The first file is called 'CFSIM' which is a save file containing a program that executes commands as if they were coming from another teletype. This program requires the use of a data file that contains the commands necessary to perform the task. As far as the Executive is concerned, the commands are being typed by a user at another teletype. The utility of this program is that it can perform a well-defined sequence of operations without requiring each one to be typed. In essence, it acts like a background job. This is a very useful and powerful tool for generating a system or subsystem.

The second file, called 'AMON', has the commands necessary to assemble the Monitor. The last file is also under user name $S^CY^CS^C$ and is called 'AEXEC'. This file contains the commands necessary to assemble the Executive when fed through CFSIM.

The following sequence of instructions will call these three files:

@!ENTER $S^CY^CS^C$.

@COPY 'CFSIM' TO /CFSIM/.

@COPY 'AEXEC' TO /AEXEC/.

@COPY 'AMON' TO /AMON/.

To load the simulator, the following sequence should be used:

@SET EXECUTIVE -1.

This instruction turns on the executive status. This is necessary because CFSIM uses certain privileged instructions which would be considered illegal by the system if the executive status is turned off.

@GO TO /CFSIM/.

This instruction places the contents of the save file, /CFSIM/ in core and goes to the starting address given when the file was created.

COMMANDS FROM /AMON/TELETYPE NO 7.

This instruction gives the name of the file where the commands are to be found, as well as the teletype on which they should be simulated.

Note: It is important that no one actually be using the teletype associated with the number given to CFSIM, since CFSIM will also connect to that teletype and an incorrect result will occur.

After the confirming period following the input of the teletype number, the program will continue, then type an @ sign, and stop. This is the user's signal to type a tape number. If a 0 is typed, it indicates that the file tape is on unit 0. If a 1 is typed, the file tape is on unit 1. In our example, the file tape was mounted on unit 1. The system will respond with ENTER and wait for a proceed signal; that is, the user must enter an L^{CS} by simultaneously depressing the CONTROL, SHIFT, and L keys. This special character tells the command simulator to proceed. From then on the entire process of assembling the Monitor is automatic, and a dialogue is printed showing the assemblies. In this example it should be noted that the system only says ARPAS once: when assembling SMDBG. This is an important package to the Monitor which contains many system parameters. Each succeeding package is assembled with an @CONTINUE ARPAS, the reason being that certain parameters are "frozen" in the system and will be used in assembling the rest of the packages.

Assembling the Monitor should take approximately half an hour on a system that is not heavily used.

The system will finally type EXIT and control will go back to the observer's teletype. The Executive can now be assembled using essentially the same sequence of commands, except this time /AEXEC/ is typed when the system asks where the commands are from. Assembling the Executive will take about 15 minutes on a machine that is not heavily used.

The first of the Executive packages is SGSUBR. This package is as important to the Executive as SMDBG is to the Monitor. Each of the succeeding packages, namely SCMNDS and SIT, will be assembled by an @CONTINUE ARPAS. When the Executive is completely assembled CFSIM will EXIT again, and the assembly of both the Monitor and the Executive is completed.

Before describing how the system is loaded, it is important to note that in case only a few packages in either the Executive or Monitor have been modified, it is not necessary to go through this entire process shown above. For example, assume that nothing is changed in the Monitor, and only package SIT in the Executive has been modified (this is where the user name directory is found).

The following sequence modifies the Executive:

```
@ARPAS.
BINARY: NOTHING, INPUT 'SGSUBR'.
@COPY 'SIT' TO /SIT/.
@CONTINUE ARPAS.
BINARY: 'BIT', INPUT: /SIT/.
```

The first step initialized the assembler using 'SGSUBR', as explained earlier. Note that in this situation the binary output of the assembler goes to the file named NOTHING because no changes were made to 'SGSUBR' and consequently there is no need to update the binary image. Next, file SIT is copied from tape to RAD. This file is put on the RAD because if one tried to do an assembly having the input and output on the same tape a TAPE WAIT (TW-) would occur, meaning the tape is already busy. This arises because the input file that ARPAS is given is still open when ARPAS tries to open the output file. The system's TAPE WAIT indication says, in effect, "you can't use that tape, it's already been given to some user". In this case, therefore, input must be put on a temporary (or RAD) file.

Finally 'SIT' is assembled using an @CONTINUE ARPAS. When this is complete the binary file, 'BIT', has been updated.

Instead of the above, suppose that it was only necessary to update one package in the Monitor. In this case, the sequence would be almost identical, starting with a call of ARPAS (only this time the input file is SMDBG). The rest of the sequence is basically the same. If there were two packages in the Monitor to reassemble, then the second step would be done twice. This would be considerably faster than going through the automatic assembly of all the packages.

If there were any changes in SMDBG (Monitor) or SGSUBR (Exec), all the packages in the respective system would have to be reassembled, since these parameter changes may affect many or all packages.

When the assembly of each package is completed, it will be noticed that two different tables are typed out. One is the list of all the null symbols and the second is the list of all the external symbols used in that package. Modification is currently being made to ARPAS so that the printing of these two tables will be an optional feature.

The only disadvantage in printing these tables is that it takes a little more time to complete the assemblies, but otherwise it does not affect the process of making a new system.

The next step is loading the system. First, copy a file known as 'LST' to a temporary RAD file. LST is a program that will automatically load and initialize the new system using the binary images. LST is loaded in the same way as AMON and AEXEC.

It should be noted that when the first proceed signal, L^{CS}, is given during the execution of CFSIM, the system will respond with a name. This name is a user name and its purpose is to determine which file directory on the tape is to be used. It is necessary to have these user names because the file directory for each contains the names of the files required by the sequence being run through CFSIM. These files correspond to the source and binary files for either the Executive (for assembling the Executive), the Monitor (for assembling the Monitor), or the binary files for both Executive and Monitor (for loading the system). In the SDS system, these user names are USER2 for assembling the Monitor, USER1 for assembling the Executive, and USER4 for loading the system.

The ordering of the user names is as follows for the SDS system:

first user	–	$S^CY^CS^C$	(for all systems)
second user	–	USER1	(assemble Executive)
third user	–	USER2	(assemble Monitor)
fifth user	–	USER4	(loading the system)

When the user inserts names for his individual system in the User Directory, his second name will correspond to USER1, the third name to USER2, and the fifth name to USER4.

After LST has given the initial "proceed", the loading process will produce a dialogue of the operations being performed. However, it will be very one-sided, in that the computer will be giving a great deal of information, whereas the observer will only have to make an occasional decision on whether or not to continue.

The first thing the computer will ask is whether there is enough drum space available to load the system. If there is not, it means the system is quite busy and, therefore, loading the system should not be attempted at this point. However, if the necessary criteria is met, the proceed signal (L^{CS}) should be given.

The first thing loaded is the Executive. The loading process consists of reading in some files from tape via DDT, making small patches, and finally initializing the code. At various points in the process you will notice a ;U is typed. ;U is a DDT command that will print out all undefined symbols (see SDS 940 DDT Reference Manual, 90 11 13). If there are no undefined symbols, nothing will be typed.

As previously mentioned, the system does some initialization. The reason for this is that many tables have to be set up which have been put in an initial form by the assembler, but not the form that the system requires. In the process of loading, therefore, certain pieces of code are executed by saying 'address';G, which means transfer control to location 'address' in the program. When this piece of code is finished, control is returned to CFSIM, which continues. As the system is loaded, some information is saved on RAD files, using commands in DDT, e.g., the DDT symbol table containing the symbols for the Executive or the Monitor is saved. At other points CFSIM will do a SAVE CORE onto a RAD file.

It is worth mentioning that in the process of loading a system, there will be many points where the program will tell the user that if there are any corrections to be made, insert them and then type L^{CS}. If there were any undefined symbols or other minor errors, DDT should be used to make the necessary revisions. When the corrections are complete, the proceed signal can be given.

After CFSIM has finished loading the Executive, it loads the Monitor. This process is similar to the loading of the Executive. Finally, CFSIM will get to a point where it says, "The loading of the system is complete and the following files must be saved if you wish to save this system...". This is very important because these RAD files are the only copy of the new system that exists. Therefore, they must be saved on magnetic tape, otherwise all this work will be lost when the system exits because RAD files are only temporary. To do this is almost the same as when one is normally using the system and it is desired to copy a RAD file to tape. However, the user must be very careful not to type RUBOUT, even if a mistake is made. The reason is that at this point one is talking to the Executive from inside the command simulator. If RUBOUT is typed, control will leave the command simulator and all RAD files will be lost. If one makes a mistake, therefore, a CR should be given, question marks (?) will be returned and the user may proceed.

When all copying is done, the user should type an L^{CS}. The program will then exit.

System generation is now complete, but there is not yet an executable system. First, the system files must be read from tape and put on the RAD in places where the system expects that the code will be. For example, RAD band 19 contains the Executive Symbol Table, and band 18 contains the Monitor Symbol Table. The Executive resides on band 9, page 0 to page 2. To transfer the new system to the RAD the following sequence of operations must be performed:

Get proper file directory

@1S^{c}Y^{c}S^{c}.

Load in DDT

@DDT.

Read in the Executive Symbol Table which was saved on the file 'EST'

;T 'EST'.

The system will respond by typing:

IDENT GSUBR

IDENT CMNDS

IDENT IT

IDENT NIO 240 Note: There is a slight delay before this number is typed.

Go back to the Executive

Ⓡ Ⓡ

Save the Executive Symbol Table

@S^CY^CS^CSAVE 19.

Puts DDT Symbol Table on band 19 of the RAD

Reset the system for this user.

@RESET

To save the Monitor Symbol Table, start again at Load DDT, except this time substitute 'SST' for 'EST', since this time we are reading the Monitor Symbol Table. The names that follow IDENT in this step will be different from those in the Executive (i.e., MDBG, W, IO, TTY, etc.).

These names refer to the titles given the packages comprising the Executive and Monitor. In the step that saves the Executive Symbol Table the number 18 is used instead of 19.

Each of the four Save files that make up the Executive and Monitor have some code that causes the package to be placed in the appropriate place on the RAD. This code is inserted during the loading of the System (LST). By saying GO TO each of the save files, that package will automatically place itself on the RAD and return control to the Executive when completed.

Note: It is very important when executing the following sequence, that no one except the user executing these commands be on the system, because vital pieces of Monitor and Executive are being replaced. This code may not be consistent with the current system. Thus, if someone causes the Executive to be swapped in during this sequence, strange results may occur, and, in fact, the system may crash.

Put main Executive on RAD

@GO TO 'EXEC'.

@RESET.

Put Executive overlay on RAD

@GO TO 'XM2'.

@RESET.

Put main Monitor on RAD

@GO TO 'SYSSAVE'.

@RESET.

Put nonresident Monitor on RAD

@GO TO 'MM0'.

At this point, the new system is completely on the RAD, and it is now possible to run WDT and dump the RAD out onto tape.

To run WDT, mount a tape on unit 7. Place the paper tape for WDT in the reader and do a self-fill on paper tape. When the system is completely dumped, unit 7 will rewind. It is advisable to remove the Write ring immediately so that the tape cannot be inadvertently destroyed.

Note: In case a system has no paper tape reader, but does have a card reader, there is a self-filling deck for WDT.

The previous explanation assumes that the system being generated contained no errors. However, if one had reason to believe that there might be some bugs, it would be highly desirable to have a method of debugging the system itself. As can be seen from the attached RAD map, there is an area reserved for a debug version of each part of the system. Preparing a system for debugging involves more than just placing the code in the debug area of the RAD because the system expects to find the Executive, Executive Overlay, and Non-Resident Monitor in their fixed places on the RAD. Before the new system can be run, certain changes must be made to reflect the fact that these packages are in the debug area. The two types of operations that must be done are:

Making changes to the tables

Getting a system onto the debug area

The following instructions will achieve the desired results:

Save Executive symbol table (same as before)

@RESET.

Put new Executive in core

@PLACE 'EXEC'.

Save Executive

@S^CY^CS^C DUMP BLOCKS 2 TO 3 BAND 15.

@RESET.

Put Executive Overlay in core

@PLACE 'XM2'.

Save Executive Overlay

@S^CY^CS^C DUMP BLOCK 2 TO 2 BAND 15, PAGE 2.

RESET.

Save Monitor Symbol Table (same as before, but do not reset when complete).

Get main Monitor

@PLACE 'SYSSAVE'.

Return to DDT and fix up SMT entries

@CONTINUE DDT

Executive	SMT 6/	40104440	40107440
	SMT 7/	40104540	40107540
	SMT 10/	40104640	40107640
Non-Resident Monitor	MSMT/	40106040	40106240

The two octal digits denoted by the bracket represent the drum address, in 2K units, where a particular piece of code is found.

In the regular system, the Executive is found on Band 9, pages 0-2 (i.e., drum addresses 44_8, 45_8, 46_8) and the Monitor Overlay is on Band 12, page 0 (60_8). However, in the debugging system these numbers should be 74_8, 75_8, 76_8 for the Executive (Band 15, pages 0-2) and 62 for the Monitor Overlay (Band 12, page 2).

Ⓡ Ⓡ

Save revised Monitor

@S^CY^CS^C DUMP BLOCK 0 TO 4 BAND 13.

RESET.

Put Monitor overlay in MAP

Place /MM0/.

Save Monitor overlay

S^CY^CS^C DUMP BLOCK 6 TO 6 BAND 12, PAGE 2.

RESET.

At this point, the entire Monitor and Executive have been saved in the desired areas. If it is desired to try the new system, make sure no one else is on; then type

@SCYCSC GO TO 13.

This loads in the new Monitor from Band 13 and starts it up.

By typing Ⓡ a user should get a response.

If nothing happens, or if it starts up properly but later crashes, the core image can be saved and a fresh copy of the working system can be brought in. This revives the system and allows the user to look at the old (crashed) system, which was saved, to see what might have happened. The following sequence will accomplish this:

@RESET.

@SCYCSC DEBUG 18.

Start up DDT with Monitor symbol table in DDT map and live system in program map

Ⓡ

RESET program map

@KILL PROGRAM.

Put crashed Monitor system in program map as a user program

@SCYCSC LOAD BLOCKS 0 TO 4 BAND 16.

Inspect system

@CONTINUE DDT.

If a bug is found, it can be corrected in the fresh image on Band 13 using the following sequence:

Ⓡ Ⓡ

@KILL PROGRAM.

@SCYCSC LOAD BLOCKS 0 TO 4 BAND 13.

@CONTINUE DDT.

Make changes

Ⓡ Ⓡ

Put corrected image back on RAD

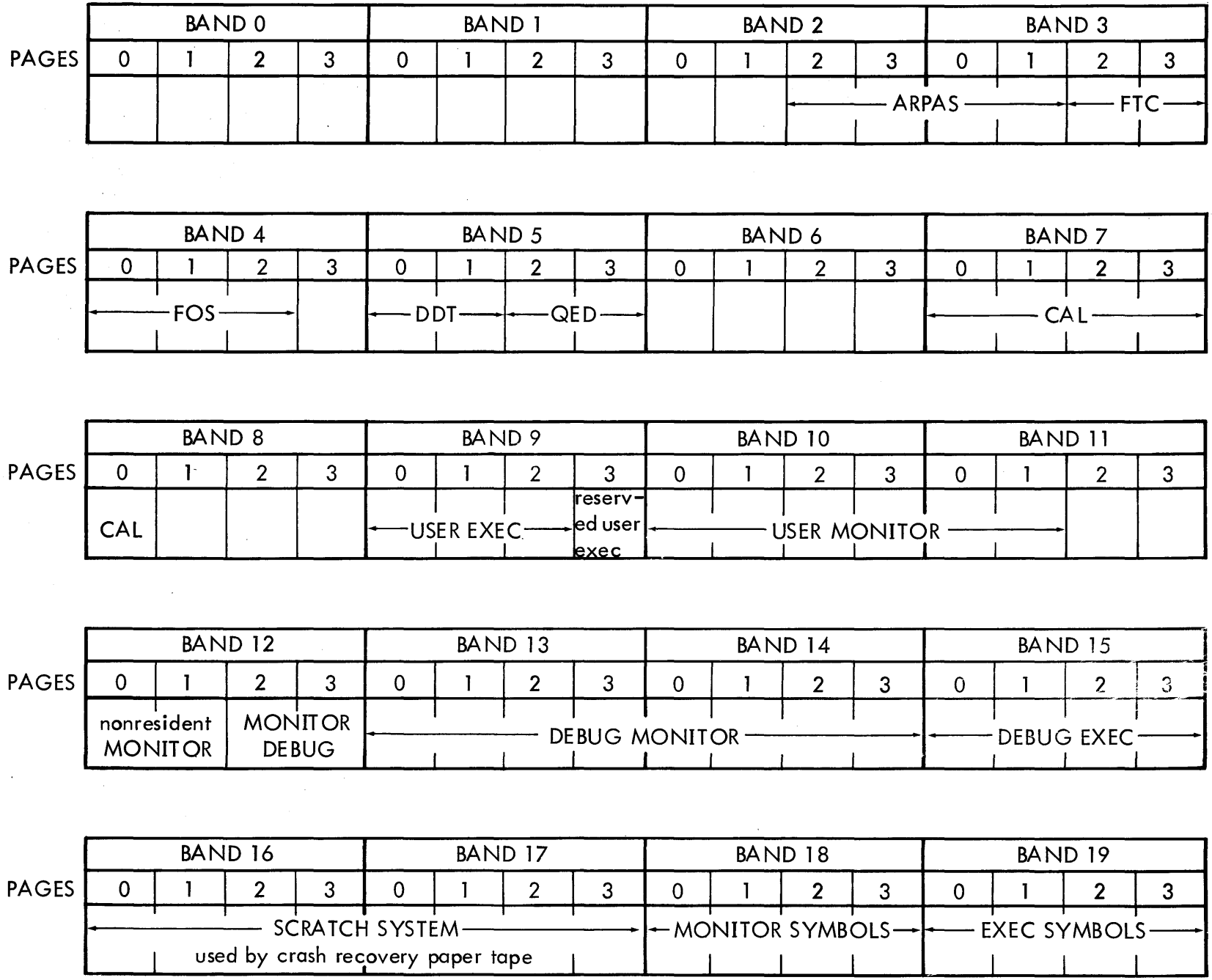
@SCYCSC DUMP BLOCKS 0 TO 4 BAND 13.

Try again

@SCYCSC GO TO 13.

Etc.

940 RAD MEMORY MAP (8K Bands) - System 1.85



BANDS 20-35 are used for swapping. BANDS 36-63 are used for files.