

A. C. D.
LIBRARY

903 UTILITY
SOFTWARE

Book No. 120

Copy No. 23

Amendment No. 0

© The Copyright in this document is the property of Elliott Flight Automation Limited. The document is supplied by Elliott Flight Automation Limited on the express terms that it is to be treated as confidential and that it may not be copied used or disclosed to others for any purpose except as authorised in writing by this Company.

AIRBORNE COMPUTING DIVISION
ELLIOTT FLIGHT AUTOMATION LIMITED

PREFACE.

This book describes the following tapes:-

BINCOP	20/7/70	Binary Mode 3
900 TELEOP	20/7/70	Binary Mode 3
DO-ALL 4	3/8/70	Binary Mode 3
BOWDLER	20/7/70	Binary Mode 3
GENSORT	8/3/71	Binary Mode 3
TATDOC	27/7/70	Binary Mode 3
DUMP	15/4/70	Binary Mode 3
Clear Store	(Mylar)	
900 VERIFY	2/5/70	Binary Mode 3
Autostart	(Mylar)	
900 AMEND	20/7/70	900-Series Telecode
900 T5-CUM-QIN	20/7/70	900-Series Telecode
900 T6-CUM-QOUT	22/10/68	900-Series Telecode
900 STORE PRINT	3/8/70	900-Series Telecode
MONITOR	12/3/71	900-Series Telecode

The above tapes are all utility programs, diagnostic routines or diagnostic subroutines, and all may be run on any 903, 905, or 920 computer having tape-reader mode 3, except MONITOR which will not work on a 920A.

Where a Telecode is involved, the above tapes will all operate in 900-Series Telecode, and some are dual-code programs which will operate in 920 code also. This book does not include tapes which operate in 920 code only.

The tape reader mode and Telecodes referred to above and within this book, and the A.C.D. Binary tape format referred to in the book, are all described in Book 106 '903/905/920 USEFUL NOTES'.

BINCOP 30/7/70 Binary Mode 3

1. FUNCTION

BINCOP is a 903/905/920 program for making fully-checked copies of a tape. It may be used to copy the following types of tape:-

- a) Tapes containing only even parity characters, up to 171 feet long. (Telecode tapes and Mode 1 Binary tapes have even parity).
- b) Tapes containing no holes in track 8, up to 171 feet long. (Mode 3 tapes in "A.C.D. 900-SERIES 18-BIT BINARY TAPE FORMAT 1/4/70" have no holes in track 8).
- c) Any other tape up to 150 feet long.

When the tape to be copied is read in, leading blanks are ignored, but blanks occurring after this are copied exactly.

2. HARDWARE

BINCOP is written for use on an 8192-word 903, 905 or 920, with a high-speed punch and reader. "Mode 3" tape reader input is required. There are certain limitations if non-stop-on-character readers are used, (see para. 5).

3. METHOD OF USE

- 3.1. The following operations are performed using reader mode 3.
- 3.2. Read in "BINCOP 20/7/70 Binary Mode 3" by triggering at initial instructions.
- 3.3. Read in the tape to be copied, by triggering at 8 if the tape to be copied contains only even parity, 12 if it contains no holes in track 8, or 11 otherwise. BINCOP will now read in and store the tape until it finds 2 inches of blanks.

- 3.4. If the whole tape to be copied has not yet been read in, re-trigger at 9, WITHOUT MOVING THE TAPE IN THE READER. BINCOP will read in to the next 2 inches of blanks. Repeat this process until the whole tape, (except trailing blanks) has been read in.
- 3.5. Read in the tape to be copied again by triggering at 10. BINCOP will now check the tape against the contents of the store. It will not stop, even on finding 2 inches of blanks, until the whole tape has been read in. It will then punch a copy of the tape, preceded and followed by 180 blanks.
- 3.6. Read in the tape just punched by triggering at 10. BINCOP will now check the new tape against the contents of the store, and will then start to punch out another copy. If this copy is not required, stop the computer. Otherwise, when the copy has been punched, repeat this step until the required number of copies have been punched AND CHECKED.
- 3.7. Read in the original tape again by triggering at 10. BINCOP will now check the tape against the contents of the store, and will then start to punch out another copy. When this happens stop the computer. (This guards against store corruption)

4. ERROR INDICATIONS

4.1. The loader of BINCOPI performs a suncheck;
If, when reading in BINCOPI (as in para. 3.2.),
continuous punching occurs, it has mis-read.

4.2. If BINCOPI is used to copy even parity
tapes it performs a parity check on the tapes
when they are read in. If this fails, BINCOPI
will punch the character,

00000.001

until stopped.

4.3. If BINCOPI is used to copy tapes having
no holes in track 8, it checks that track
8 is, in fact, blank. If this check fails,
BINCOPI will punch the character

10000.000

until stopped.

4.4. If the tape being copied is too long to be
stored (see para 1.) BINCOPI will punch the
character

11111.111

until stopped.

4.5. When BINCOPI is triggered at 10, if the tape
being read does not agree with the contents
of the store, BINCOPI will punch the character

00000.010

until stopped.

5. LIMITATIONS ON READER

- 5.1. Stop-on-character readers may be used, irrespective of speed.
- 5.2. Non-stop-on-character readers of up to 250 c/s may be used, but if the tape to be copied contains 2 or more inches of blanks, characters may be missed when BINCOP is re-triggered at 9. However, since BINCOP reads the whole tape without stopping when triggered at 10, the absence of characters missed in this way will be detected and give an error indication.

6. CONVERTING MODE 1 BINARY TAPES (e.g. old 920A tapes) TO MODE 3 BINARY TAPES (for use on 903/905)

This conversion may be performed, if reader MODE 1 is available, by using BINCOP in the manner described in Paragraph 3; provided:-

- a) Entry 12 is used in Step 3.3.
- b) Use reader mode 1 for Steps 3.3, 3.4, 3.5, and 3.7, (but reader mode 3 still for steps 3.2 and 3.6)

This method will work irrespectively of whether track 5 of the tape is correctly or incorrectly punched. (2-PASS SIR 19/7/66, now obsolete, punched an incorrect track 5).

900 TELECOM 20/7/70 Binary Mode 3

1. FUNCTION

TELECOP is a 903/905/920 program for joining and making fully checked copies of tapes punched in 900-Series Telecode.

It may be used to copy and join tapes to a total of 171 feet long, excluding blanks and erases.

When the tape to be copied is read in, blanks, erases, and car. ret. are ignored. When the copy is punched out, each linefeed is preceded by a car. ret., and may be followed by 4 blanks if required.

Each of the tapes to be copied or joined must end with a $\textcircled{\text{H}}$. When the copy is punched out, it will also end with a $\textcircled{\text{H}}$. Where TELECOD is used to join tapes therefore only the $\textcircled{\text{H}}$ on the last tape will be reproduced.

2. HARDWARE

TELECOP is written for use on an 8192-word 903, 905, or 920, with a high speed punch and reader. "Mode 3" tape reader input is required.

Stop-on-character readers of any speed, and non-stop-on-character readers of up to 250 c/s may be used.

3. METHOD OF USE

3.1. The following operations are performed using reader mode 3.

3.2. Read in "900 TELECOD 20/7/70 Binary Mode 3" by triggering at initial instructions.

3.3. If copying a tape, read it in by triggering at 8. If joining several tapes, read the first one in at 8 and the rest at 9. TELECOD will read in and store them.

8 Read and Store
9

3.4. If copying a tape, read it in by triggering at 10 or 11. If joining several tapes, read the first one in at 10 or 11, and the rest at 9. TELECOD will now check these tapes against the contents of the store. On reaching the end of

10 Read, check and punch down
11 Read check and punch under

the last tape it will punch a copy, preceded and followed by 180 blanks. The copy will have 4 blanks after every (N) if entry 10 was used, and no such blanks if entry 11 was used.

- 3.5. Read in the tape just punched by triggering at 10 or 11. TELECOP will now check the new tape against the contents of the store, and will then start to punch out another copy.

As in para. 3.4, this copy will have blanks after every (N) if entry 10 is used. If this copy is not required, stop the computer. Otherwise, when the copy has been punched, repeat this step until the required number of copies have been punched AND CHECKED.

- 3.6. If copying a tape, read the original in again by triggering at 10 or 11. If joining several tapes, read the first original in at 10 or 11 and the rest at 9. TELECOP will now check these tapes against the contents of the store, and will then start to punch out another copy. When this happens, stop the computer. (This guards against store corruption).

4. ERROR INDICATIONS

4.1. The loader of TELECOP performs a sumcheck;
If, when reading in TELECOP (as in para 3.2.)
continuous punching occurs, it has mis-read.

4.2. TELECOP performs a parity check on the tapes
when they are read in. If this fails, TELECOP
will punch the character

00000.001

until stopped.

4.3. If the tapes being copied are too long to be
stored (see para 1) TELECOP will punch
the character

11111.111

until stopped.

4.4. When TELECOP is triggered at 10 or 11, if
the tape being read does not agree with the
contents of the store, TELECOP will punch
the character

00000.010

until stopped.

DO-ALL 4, 3/8/70, Binary Mode 3

I. FUNCTION.

DO-ALL 4, 3/8/70, Binary Mode 3, combines several utility programs into one tape.

It may be used to perform the following functions on tapes punched in 903/ISO or 920 Telecode :-

- (A) Remove blanks & erases
- (B) Insert 4 blanks after every newline
- (C) Join several tapes into one
- (D) Copy tapes of any length.
- (E) Convert to Legible tape.
- (F) Convert from the one Telecode to the other.

It may be used to perform the following functions on any 8-hole tape :-

- (G) Parity check
- (H) Check that tapes are the same
- (J) Copy tapes of any length.

Facility "J" is called QCOPY. It imposes a considerable strain on the tape reader, and therefore should not be used for copying Telecode tapes (which can be copied by facility "D" or by "TELECOPY") or Binary Tapes short enough to be copied by "SINCOPY".

2. HARDWARE

This program is written for use on an 8192-word 903, 905, or 920 with a punch and reader. Tape reader input Mode 3 is required.

For the QCOPY facility, a stop-on-character reader is essential. Stop-on-character readers are not required for the other facilities, but if they are not used, all Telecode source tapes should have some blanks after each newline.

3. TO CONVERT A TELECODE TAPE TO LEGIBLE TAPE.

- 3.1. The following operations use tape reader Mode 3.
- 3.2. Load "DO-ALL 4" into store by initial instructions (8181).
- 3.3. Read the Telecode tape in at 8. It will be converted, line-by-line, to Legible Tape.
- 3.4. The Telecode tape must start with a Newline, & end with a haltcode. Blanks, erases, & carriage return will be ignored, Newline and Tab will be treated as a space.

4. TO COPY &/OR JOIN TELECODE TAPES, &/OR
TO CONVERT FROM ONE CODE TO ANOTHER.

4.1. The following operations all use tape reader Mode 3.

4.2. Load "DO-ALL 4" into store by initial instructions (8181).

4.3. Specify the Telecode in which output is required by entering DO-ALL at

9 for 920 Telecode

10 for 903 Telecode

(Entry points 9 & 10 simply set a flag; they do NOT cause tape to be read)

4.4. Specify whether or not 4 blanks are required on the output tape after every newline or linefeed by entering DO-ALL at

11 for BLANKS

12 for no blanks

(Entry points 11 & 12 simply set a flag; they do NOT cause tape to be read)

4.5. Having performed the operations described in 4.2 and 4.3, read in the first tape to be copied by entering DO-ALL at 16. Read in any tapes to be joined to the first tape at 17.

After reading in the last tape, enter DO-ALL at 18 to obtain the final halt-code. The tapes will be processed line-by-line.

4.6. Repeat step 4.5, thereby obtaining a second copy of the new tape, and compare the two tapes by reading the first in at 20 and the second at 23. The tapes are not correct unless the second stops on reading the haltcode and the message "QCOPY OK" is punched.

4.7. The Telecode tapes must start with a Newline & end with a haltcode. Blanks, erases, & carriage return will be ignored on input. If output is in 903 Telecode, Newline will be punched as carriage return + linedfeed, Tab will be punched as Space + Tab.

4.8. If code conversion is performed the following will be considered equivalent

/ (acute)	and	⌘	(Vert. bar followed by <)
\ (grave)	"	⌘	(" " " " >)
⌘	"	⌘	(" " " " S)
"	"	~	.

5. TO PARITY-CHECK ANY TAPE.

5.1. The following operations use tape reader mode 3.

5.2. Load "DO-ALL 4" into store by initial instructions. (8181)

5.3. Read the tape in at 20. It will be checked for EVEN parity until 20 consecutive blanks are found. If the whole tape has not yet been read, re-trigger at 17 or 20 until the whole tape has been read in (except trailing blanks.) If the tape is correct, no output will be given.

6. TO CHECK THAT TAPES ARE THE SAME.

- 6.1. The following operations use tape reader mode 3.
- 6.2. Load "DO-ALL 4" into store by initial instructions. (8181).
- 6.3. Read the 1st tape in at 20, if it is even-parity, and at 21, otherwise. It will be read until 20 consecutive blanks are found, and a sum-check and blank-count will be formed. If the whole tape has not yet been read, re-trigger at 17, WITHOUT MOVING THE TAPE IN THE READER, until the whole tape has been read in (except trailing blanks).
- 6.4. Read the 2nd tape in at 23. The tapes are not the same unless this tape stops on reading the last non-blank character and the message "QCOPY OK" is punched. Repeat for any other tapes to be compared.

7. TO COPY ANY 8-HOLE TAPE OF ANY LENGTH ("QCOPY")

- 7.1. If the tape can be copied by RMCOP, TELEOP, or the method of Paragraph 4, do so.
- 7.2. The following operations use tape reader mode 3.
- 7.3. Load "DO-ALL 4" into store by initial instructions. (8181).

- 7.4. Read the tape to be copied in at 20, if it is even-parity, and 21, otherwise. It will be read until 20 consecutive blanks are found, and a sum-check and blank-count will be formed. If the whole tape has not yet been read, re-trigger at 17, WITHOUT MOVING THE TAPE IN THE READER, until the whole tape has been read in (except trailing blanks.)
- 7.5. Read the tape to be copied in again at 22. It will be copied, character by character, until the last non-blank character is reached OR until the tape reader explodes. The message "COPY OK" should be punched. If it is not, or if the tape does not stop on reading the last non-blank character, destroy the copy just made, & return to 7.4.
- 7.6. Read the tape just produced in at 22, if another copy is required, or 23, otherwise. Again the message "COPY OK" should be punched. If it is not, or if the tape does not stop on reading the last non-blank character, destroy the copy JUST READ IN, and the copy just made if entry was at 22; & return to 7.5. Repeat until the required number of tapes have been made & checked.
- 7.7. Check that you now have one more "COPY OK" message in the rubbish bin than the number of new tapes made.
- 7.8. If, whilst copying a tape at 22, the tape is not passing with a character exactly in the reader light whilst each character is punched, this method of copying tapes is unlikely to work.

8. ERROR INDICATIONS.

8.1. When DO-ALL is being loaded, the loader on the binary tape performs a sum-check. If this fails continuous output is given on the punch.

8.2. DO-ALL can give the following error indications in legible tape form:-

CH I/O ERROR 1. } Parity error on tape being read.
QCOPY ERROR 1. }

CH I/O ERROR 2. First character on Telecode tape is not a newline, linefeed, or carriage return symbol.

CH I/O ERROR 3. When reading or punching in 920 Telecode; a character cannot be converted to or from internal code.

CH I/O ERROR 4. A line of Telecode text contains over 120 characters.

QCOPY ERROR 2. The tape just read in at 22 or 23 does not agree with the tape previously read in at 20 or 21.

BOWDLER 90/7/70 Binary Mode 3

1. FUNCTION.

"BOWDLER 20/7/70 Binary Mode 3" is a program for editing Telecode tapes of any length, using a 900-series 18-bit computer.

The user punches an edit tape, or steering tape, which defines the amendments to be made to the main tape. The edit tape is read into the computer and stored. The main tape is then read, edited, and punched in accordance with the instructions on the edit tape.

BOWDLER will accept edit tapes and main tapes punched in either 920, 903 or ISO telecodes. The ISO telecode is similar to the 903 telecode and it is the telecode used by the Elliott 502 computers. The edit tape need not be punched in the same telecode as the main tape, but if the telecodes of the two tapes are different then all the characters used in the edit tape must have an equivalent character in the 920 telecode. Note that BOWDLER treats the 920 non-escaping vertical bar as a character in its own right, and thus does not recognise \bar{k} , \bar{x} , $\bar{\$}$ or $\bar{\$}$ as equivalents for $\bar{/}$, $\bar{\backslash}$, $\bar{\$}$ or $\bar{1/2}$ in 903 code.

2. HARDWARE.

This program is written for use on any 900-series 18-bit machine having an 8192-word store and a high-speed punch and reader. Tape reader input "Mode 3" is required.

Stop-on-character readers are not required.

3. METHOD OF USE.

All operations are performed using tape reader Mode 3.

"BOWDLER, 20/7/70 Binary Mode 3" is entered by initial instructions, (8181).

An edit tape of the edit instructions (section 4.) should then be input at 8 or 13. The number of instructions or any errors (section 6) in the edit instructions is then output in 920 (entry 8) or 903 (entry 13) code.

The tape to be edited should be input at 9 when the edited tape will be output. If the editing process is correctly performed and no 'wait' instructions have been encountered the output tape will be terminated by the standard trailer of 120 blank, 10 erase and 120 blank characters but if any instruction cannot be obeyed then the output tape will be terminated by the error type and number of that instruction. In the latter case, the output tape will be an edited version of the original up to that instruction.

If the tape to be edited is too long to be edited in one pass (i. e. more than 72 feet), then BOWDLER stores as much as it can and edits to the last string of newline and 4 blank characters. If no such string is held in store, then an error is output (section 6). To continue editing the tape should be reinput at 10 with these blanks in the reader. If the store becomes full while reading blanks, the tape will be read until the first non-blank character has been read. If this character is a stop code then editing will continue as though the whole tape had been read normally. When the edit tape is obeyed this character (unless a stop code) will not be edited. Apart from a string of newline and 4 blank characters being required on the main tape no additional instructions are required on the edit tape to enable BOWDLER to handle tapes more than 72 feet.

If a 'wait' instruction occurs during the course of editing, editing must be resumed by entering the next main tape at 10.

If the editing process appears to go wrong the number of the instruction being obeyed can be obtained by entering at 11.

When the editing process has been performed without an error exit, the edited tape may be sumchecked by entering BOWDLER at 12, with this tape in the reader. This should read up to but excluding the first of the 10 erase characters. If this sumcheck is correct, there will be no error message output.

If the above entry points are used in the following sequence any reader and punch errors and any store corruptions should be detected:-

1. Load BOWDLER at 8181
2. Load the steering tape at 8 or 13. Check that the instruction count is given, (but no error indication), but do not tear this output off the punch.
3. Load the tape(s) to be edited at 9, retriggering at 10 when necessary. Check that the final 10 erases are punched when the editing is complete, but do not tear this output off the punch.
4. Repeat steps 1, 2, & 3.
5. Now tear off the output from the punch, rewind it & place it in the reader.
6. Pull the 1st instruction count through the reader by hand, tear it off & throw it away.
7. Check the 1st tape produced; by triggering at 12; the tape should stop with the 1st erase in the reader light. Tear this tape off & rewind it.
8. Pull the erases and the 2nd instruction count through the reader by hand, tear them off & throw them away.
9. Check the 2nd tape produced; by triggering at 12; the tape should again stop on the 1st erase. Tear this tape off & rewind it.
10. Throw away the second set of erases.

Note that both tapes are checked against the sumcheck formed in step 4. This proves they have the same sum, and thus punch faults, reader faults, and store corruptions should all be detected.

Further the above method gives 2 copies; and prevents them being confused with one another until they have both been checked.

4. EDIT INSTRUCTIONS.

A summary of instructions is given in Appendix 1 and a simple example in Appendix 2.

Facilities exist for copying and deleting the main tape and inserting new tape. There are also instructions for performing a group of editing instructions repeatedly without repeating the group of instructions in the steering tape.

Each editing instruction is specified by a special instruction (or group of instructions in the case of the repetition facility). These instructions constitute the edit tape and they appear in the order that the alterations occur in the main tape. The repetition instructions become operative as soon as they are declared in the editing process as distinct from the steering tape phase. The repetition instructions are declared as soon as all the previous instructions, except other repetition instructions, have been obeyed. All the instructions, except the repetition instructions, may be used consecutively. All instructions, except the insert and the replace instructions, are punched on one line.

The first non blank character of the edit tape should be a newline character so the telecode of the edit tape may be partially decoded. This character distinguishes the 920 telecode from the 903 and ISO telecodes but not the 903 and ISO telecodes. This should be followed by two telecode instructions, the first specifies the telecode of the edit tape, the second the telecode of the main tape and followed by the edit instructions. These are each of three letters (upper or lower case), independent of layout characters between the previous newline and the third letter and must be terminated by a stopcode immediately following a newline. Each complete edit instruction should be terminated by a newline. In 903 code each newline is preceded by a carriage return on output. Any adjacent carriage returns are then deleted. It is not necessary to blank the edit tape since Bowdler reads the whole tape at full reader speed.

4. 1 Telecode Instructions

These are '920', '903', or 'ISO' (upper or lower case) for the 920, 903 or ISO telecodes respectively.

4. 2 Preserve Erase Characters

Normally erase characters are treated as layout characters and edited out, but when 'SPE' is used all subsequent erase characters are copied. This facility once invoked cannot be cancelled.

4. 3 Insert Blank Characters

The instruction designator 'INB' ensures that there are only 4 blank characters following each newline character output. Once used, this facility cannot be revoked. If this facility is not invoked, no blanks will be output within the output tape.

4. 4 String Search Instructions

The instruction designator should be followed by a string of characters terminated by a newline character. The instruction designators are:-

- 'CLE' - copy up to and including the newline character before the specified line.
- 'DLE' - delete up to and including the newline character before the specified line.
- 'CLI' - copy up to and including the newline character after the specified line.
- 'DLI' - delete up to and including the newline character after the specified line.
- 'CSE' - copy up to but excluding the specified string (layout characters before the string are copied)
- 'DSE' - delete up to but excluding the specified string (layout characters before the string are deleted)
- 'CSI' - copy up to and including the specified string
- 'DSI' - delete up to and including the specified string

The string of the instruction on the edit tape is independent of any layout characters and string searches are independent of any layout characters on the main tape. The maximum number of characters (including layout) allowed on the main tape between the first and last character of the string is 253 characters. In the case of a line search the first

character is considered to be a newline character immediately followed by a string, and the search is then conducted only at the beginning of each line. If too many characters are specified then a Buffer Overflow error is output.

4. 5 Insertion Instruction

This function may be used to insert a string of indefinite length at any point in the main tape. The length of the string to be inserted is limited only by whether there is sufficient room in the store module to store the characters. If this Buffer overflows, then a warning message is output and a 'copy' instruction substituted for the insert instruction and no further instructions will be compiled.

The designator for this instruction is 'INS' and should be immediately followed by a newline character and the string of characters to be inserted, terminated by ' $\uparrow\%$ ' and newline character. Any characters between the 'S' of 'INS' and the next newline character will result in a warning message being output and any such characters will be ignored when the edit tape is executed. The ' $\uparrow\%$ ' string must contain no layout blank or erase characters within the string. This string should be preceded by a newline character if the next part of the main a tape is to be preceded by a newline character. The sequence of characters to be inserted may be any sequence of any printable or layout characters (with the exception of ' $\uparrow\%$ ' and erases which are normally edited out). It is possible to insert ' $\uparrow\%$ ' by inserting one character at a time.

4. 6 Replace Instruction

The instruction designator is 'RTS' when the string is searched for anywhere on the line or 'RTL' when the string is searched for only at the beginning of each line. The string or line to be replaced follows the instruction designator on the same line and is similar to the string search string described in section 4. 4. The string to be inserted should follow on the next line; is similar to the insert string described in section 4. 5 and is terminated in the same manner. When the 'replace' instruction is compiled, it is counted as three instructions for the total number of instructions compiled, or for the output of the number of the instruction containing an error.

Note that the following are equivalent:-

$\left. \begin{array}{l} \text{RTS A} \\ \text{B}\uparrow\% \end{array} \right\} \text{ is equivalent to } \left\{ \begin{array}{l} \text{CSE A} \\ \text{DSI A} \\ \text{INS} \\ \text{B}\uparrow\% \end{array} \right.$

$\left. \begin{array}{l} \text{RTL A} \\ \text{B} \\ \text{B}\uparrow\% \end{array} \right\} \text{ is equivalent to } \left\{ \begin{array}{l} \text{CLE A} \\ \text{DLI A} \\ \text{INS} \\ \text{B} \\ \text{B}\uparrow\% \end{array} \right.$

4. 7 Repetitive Strings

The repetitive instructions must begin with 'REP' followed by a digit from 0-9 (to define this particular repetitive sequence for the cancellation facility). This should be followed by the instructions to be repetitively obeyed and terminated by 'END', preceded and followed by a newline character.

Since the repetitive instructions operate concurrently with the main edit instructions, it is necessary to have a main edit instruction before cancelling the repetitive sequence. The instruction designator CAN followed by the digit used to define the repetitive sequence is used to cancel the repetitive instructions.

An inclusive search instruction must follow after an exclusive search and insert instructions otherwise the string will be in store and it will be continuously found and the insert string continuously output. It is not permissible to nest repetitive instructions (i. e. a repetitive sequence of instructions within a repetitive sequence of instructions but it is permissible for repetitive sequences to follow repetitive sequences even if the previous repetitive sequence has not been cancelled. In the latter case, if no cancellation has taken place, the first string to be found takes precedence if both are found on the same line, e. g

```
REP2
RTS ON SET
FRED ↑ %,
END
REP1
RTSCOLON
FREDA ↑ %,
END
```

If the line on the main tape is
PROTOCOL ONSET

then COLON is found first and REP1 obeyed.

When more than one string is identical except for extra characters at the end of one of the strings of the pair and all are found on the same line, the order of precedence is the main edit string followed by the repetitive sequence with the higher digit labels taking precedence over the sequence with the lower digit label. This facility is suppressed when a 'copy' instruction (4. 8) is obeyed. On restarting at 10 for further tapes, the repetitive instructions will still be carried out.

4. 8 Copy Instructions

The Instruction designator 'COP' will copy the main tape up to the first stopcode immediately following a newline character. On restarting at 10, if there are further edit instructions, 120 blanks are output.

4. 9 Wait Instruction

There are two types of 'wait' instruction. The first 'WAI' causes editing to cease until restarted at 10 and the remainder of the main tape held in store is deleted.

The second 'WTS' causes the editing to 'wait' whenever a stopcode is found and the stopcode deleted. Once this facility has been invoked it can not be cancelled but it is suppressed whenever the 'copy' instruction is used.

4. 10 Insertion of Layout Characters

The following instruction designators are used to insert one layout character.

- 'PNL' - punch newline character *or carriage return + linefeed*
- 'PSP' - punch space character
- 'PHT' - punch horizontal tab character
- 'PSC' - punch stopcode character
- 'PVT' - punch paper throw character
- 'PBE' - punch bell character if maintape *in 903 or ISO code*

4. 11 Deletion of Layout Characters

The following instructions delete the first occurrence of a layout character and copies all preceding layout characters. If there is no occurrence of the layout character before the first printable character, all layout characters are copied and the deletion instruction ignored.

- 'DNL' - delete newline character *or linefeed*
- 'DSP' - delete space character
- 'DHT' - delete horizontal tab character
- 'DVT' - delete paper throw character
- 'DBE' - delete bell character if maintape *in 903 or ISO code*

5. LIMITATIONS.

The amount of workspace reserved in BOWDLER for characters, etc. is limited; thus there are limitations on the main tape and edit tape.

These are:-

- (i) If the main tape is longer than 72 feet there must be 4 blanks after a newline at intervals of less than 72 feet throughout the tape.
- (ii) The edit tape must not be longer than 72 feet or about 8640 characters, INCLUDING blanks and erases.
- (iii) The edit tape must not contain more than 119 instructions. The first two 'telecode' instructions do not count, but 'RTL' and 'RTS' count as 3 each.
- (iv) The total number of characters in the strings of 'copy' 'delete' 'insert' and 'replace' instructions on the edit tape must not be more than 2984, INCLUDING SOME of the blanks and erases.
- (v) The number of characters in any one string of a 'copy' or 'delete' instruction or of the first string of a 'replace' instruction, on the edit tape, must not be more than 64, EXCLUDING blanks and erases.
- (vi) The number of characters on the main tape between the first and last character of a string must not exceed 253 characters INCLUDING blanks and erases.
- (vii) Only 10 repetitive groups of instructions can be operative at any one instant.

6. ERROR INDICATIONS & WARNINGS.

There are 3 types of error indications associated with BOWDLER:

- (i) Loader Sum Check.
- (ii) Error Messages.
- (iii) Warning Messages.

These are described in turn below.

6.1. Loader Sum Check

The loader at the start of the BOWDLER binary tape performs a sum-check. If continuous punching occurs when BOWDLER is loaded, this sum-check has failed.

6.2 Error Messages

The format of the error messages is as follows:

E "M"
I "N"

where "M" is the error number and "N" is the number of the instruction, being compiled in the steering tape phase or being obeyed in the editing phase. There are 10 blanks between E "M" and I "N". The edit tape phase ceases after an error message.

E0

This indicates that one of the two telecode words is incorrect or missing, e. g. 921 instead of 920.

E1

This indicates that an edit instruction designator is incomplete or incorrect, e. g. WTT instead of WTS.

E2

This error message occurs when there is no string after a string search or replace instruction.

E3

This error indicates that there is no digit or an illegal character after a REP or a CAN instruction designator, e. g. REP A instead of REP 1.

E4

Two REP or END instruction designators are used consecutively if this error message occurs, e. g.

REP 5
RTS fred
mike ↑ %,
REP 6

E5

This error message denotes that either a REP instruction designator has the same label string as another REP instruction which has not yet been cancelled or a CAN instruction designator has been used for a non-existent repetitive group.

E6

This error message shows that a REP instruction designator has not been followed by a copy search or replace instruction.

E7

This error denotes that a character in a search string or insert string cannot be converted to or from the 920 telecode, e. g. the character ll in the 920 telecode cannot be converted to 903 or ISO telecode.

E8

This error denotes that there are more than 253 characters between the first character of the string or preceding newline character and the last character of the string for a string or line search respectively, during the editing phase.

E9

This error denotes that a string search or replace instruction cannot be obeyed, i. e. the final stop-code has been reached. The edited tape is output before the error message. Alternatively it denotes that the final stop-code of the steering tape occurs in the middle of an insert instruction.

E10

This error message shows that the sumcheck has failed. This is the only error message which has no instruction number after it.

E11

This error message denotes that the first character read on a tape on the first or subsequent pass was not a blank (see para. 4.2).

E12

This error message shows that a tape has been partially read but there is no stretch of two or more blanks after a newline character on the part of the tape that has been read.

E13

This error message denotes that the first character on the edit tape is not a 920, 903 or ISO newline character.

E14

This error indicates a parity error on the character just read. Parity errors found on the edit tape are always followed by I 0.

The format of the warning messages is as follows:-

W "M"
I "N"

where "M" is the warning number and "N" is the number of the instruction being compiled. There are ten blanks before the W and the I. The edit tape phase continues but the edit tape should be checked to see whether the warning can be ignored.

W0

The warning message indicates that the end of the steering tape occurred in the middle of a repetitive set of instructions. All instructions after the last REP are ignored and a COP instruction is substituted for the REP instruction designator.

W1

This warning message shows that there are characters are an instruction designator but before a newline character. The characters are ignored, e. g. WTSABC is treated as WTS.

W2

This warning message denotes that a non-cancel instruction has been used in a repetitive set of instructions. The instruction is removed and ignored by the instruction count.

W3

This warning message shows that there is no room in the instruction list or character list. The instruction which cannot be compiled is replaced by a COP instruction and no more instructions are compiled.

W4

This warning message shows that there are more than 64 characters in a search string or replace string. All extra characters are ignored.

W5

This warning message denotes that non-cancel instruction has been used twice. The second use of the instruction is removed and not included in the instruction count.

APPENDIX 1.

This is a list of the instruction designators each of which can be punched in upper case or lower case letters

920	tapes in 920 telecode.	} Specify code of
903	tapes in 903 telecode.	
ISO	tapes in 502 ISO - telecode.	
		1. Edit tape
		2. Tape being edited.
CSE	<u>string</u>) (
CSI	") (C means copy
DSE	") (D means delete
DSI	") (L means until <u>string</u> is found at start of a line
CLE	") (S means until <u>string</u> is found anywhere on a line
CLI	") (I means inclusive of <u>string</u>
DLE	") (E means exclusive of <u>string</u>
DLI	") (
INS	Insert <u>string</u>	
<u>string</u> ↑ %		
RTS	<u>string</u>)	
<u>string</u> ↑ %)	Replace specified string or line
RTL	<u>string</u>)	
<u>string</u> ↑ %)	
SPE	Copy erases	
INB	Insert four blanks after each newline	
WTS	Wait on each stop-code	
WAI	Wait	
COP	Copy main upto and including next stop-code on a newline	
REP	<u>label</u>)	The instructions between REP <u>label</u> and END
END)	are obeyed whenever possible until the instruction
CAN	<u>label</u>)	CAN <u>label</u> is obeyed.
PNL)	{ newline (920); <i>newline & carriage return (903/150)</i>
PSP)	{ space
PHT)	{ tab
PSC)	Insert one (stop-code
PVT)	{ paper throw
PBE)	{ bell character (903/150)
DNL)	{ newline
DSP)	{ space
DHT)	{ tab
DVT)	Delete one (paper throw
DBE)	{ bell character (903/150)

APPENDIX 2.

This appendix is a simple illustration of the use of Bowdler. The original tape for the example is as follows:-

((EXAMPLE))

[CHECK]

```
CHECK      >1
           0      +0
           3      COUNT
           5      WS
REPEAT     5      WSP
           7      EXIT
           1      -1
           6      WSP
           10     COUNT
           8REPEAT
EXIT       4      COUNT
           0      CHE CK
           /8     1
WS         >1
WSP       >1
COUNT    >1
%
```

The edit tape is as follows:-

```
n/1
920
920
INB
CSE      0      +0
INS      4      +0
```

```
↑%,
REP 2
RTS COUNT
SUM↑%,
END
CLI REPEAT
DLI 7 EXIT
INS
9 REPEAT
↑%,
CAN2
CSE REPEAT
PTB
```

CLI
DSE
INS

CSI
DSP
COP
s/c

EXIT
CHECK

7 ↑%
CHE

The edited tape is as follows:-
(EXAMPLE)

```
[CHECK ]  
CHECK            >1  
                 4        +0  
                 0        +0  
                 3        SUM  
                 5        WS  
REPEAT           5        WSP  
                 9        REPEAT  
                 1        -1  
                 6        WSP  
                 10       COUNT  
                 8        REPEAT  
EXIT             4        COUNT  
                 7        CHECK  
                 /8.      1  
WS               >1  
WSP              >1  
COUNT          >1  
%
```

GENSORT, 8/3/71, Binary Mode 3.

1. FUNCTION.

"GENSORT 8/3/71 Binary Mode 3" is a program for sorting lines of text into Alphabetical (or Numerical) order, using a 900-Series 18-bit computer.

The user defines "alphabetical order" by means of a short Alphabet tape. The tape or tapes containing the text to be sorted may be of any length. The tapes may be in 900-Series or 920 Telecode; other codes may also be acceptable.

2. HARDWARE.

GENSORT is written for use on any 900-Series 18-bit machine having an 8192-word store and a high-speed punch and reader. Tape reader input "Mode 3" is required. Stop-on-character readers are not required.

3. TELECODES

The alphabet tape and the tapes to be sorted must be in the SAME Telecode; the sorted output will be in the same code.

This Telecode may be either 900-Series Telecode (or 903, ISO, or ASCII codes with even parity) or in 920 Telecode, or any OTHER Telecode using the same characters for blank, erase, carriage return & linefeed (or newline) and haltcode as either of these codes.

4. ALPHABET TAPE

The user should define "Alphabetical Order" by punching a tape comprising :-

Newline or Carriage Return + Linfeed
The characters used on the tapes to be sorted, excluding Newline & Haltcode, but including Space & Tab if required, in the desired alphabetical order to a maximum of 63 characters
Newline or Carriage Return + Linfeed
Haltcode

The 63 characters will usually include the letters A-Z in order and digits 0-9 in order; if letters of both cases are required it is suggested that they are punched in the order Aa Bb Cc Dd Yy Zz.

Blanks, Erases, and Carriage Return will be ignored on the Alphabet tape.

5. TAPE TO BE SORTED

The tape to be sorted need not start with a newline but must end with a haltcode, preferably on a new line. There is no need for blanks after each new line.

Blanks, Erases, and Carriage Return will be ignored; apart from these the tapes to be sorted must not contain any characters not on the alphabet tape.

The text to be sorted may be on several separate tapes. There is no limit to the number of characters on a line, or to the total number of lines to be sorted, but each tape should not exceed about 20,000 characters.

6. METHOD OF USE.

- 6.1. The following operations all use reader mode 3.
- 6.2. Read GENSORT into the computer by initial instructions (8181) and define the Telecode to be used and the Alphabetical Order by reading the Alphabet tape in at 8.
- 6.3. To SORT a tape or tapes totalling less than about 20,000 characters, read one of the tapes in at 9, and the others, if any, at 10. Then trigger at 12 to obtain the sorted output. (Then do step 6.6)
- 6.4. To MERGE a tape or tapes totalling less than about 20,000 characters INTO an already sorted tape or tapes of unlimited length, read one of the unsorted tapes in at 9, and the others, if any, at 10. Read the first of the sorted tapes in at 11, and the others, if any, at 10 or 11. The sorted tapes will be copied line-by-line, the store contents being merged into the output where appropriate. Finally trigger at 12 to obtain the remainder of the sorted output.
- 6.5. To SORT tapes totalling over 20,000 characters, arrange the tapes in groups each totalling less than 20,000 characters. Sort the first group as in 6.3, merge the next group into the output just obtained as in 6.4, and repeat step 6.4 as often as necessary until all groups have been merged into one tape.
- 6.6. Having sorted the tape once, repeat the whole process and compare the two sorted tapes with a suitable program, e.g. DO-ALL.

7. EXAMPLE

Alphabet tape :-

ⓃABCDEFGHIJKLMN OPQRSTUVWXYZ.,0123456789 Ⓢ*(%) ⓃⓂ

Tape to be sorted :-

(BRACKETS)
SMITH, D. (SENIOR)
4 DIGIT
2 DIGIT
2 SPACES
INDENT

ZULUARAB
SING 64%

*THESE LINES WILL
**STAY IN
***THIS ORDER

ARABZULU
ZULU
SING 36%
.POINT
SMITH, D. (JUNIOR)
SMITH, A.B.C.
JONES, D.E.F.

Sorted output :-

ARABZULU
JONES, D.E.F.
SING 36%
SING 64%
SMITH, A.B.C.
SMITH, D. (JUNIOR)
SMITH, D. (SENIOR)
ZULU
ZULUARAB
.POINT
2 DIGIT
2 SPACES
4 DIGIT
INDENT
*THESE LINES WILL
**STAY IN
***THIS ORDER
(BRACKETS)

8. ERROR INDICATIONS.

The loader at the start of the GENSORT binary tape performs a sum-check. If continuous punching occurs when GENSORT is loaded, this sumcheck has failed.

Errors given by GENSORT itself take the form of 18" of a certain character, (preceded & followed by a length of blank) the character defining the error:-

- | | |
|-----------|--|
| 00000.001 | 1st character of Alphabet tape is not newline, carriage return or linefeed. |
| 00000.010 | Character appears on tape to be sorted or merged but was not on Alphabet tape. This error will be given if there is a parity error on the tape to be sorted or merged OR ON THE ALPHABET tape. |
| 00000.011 | Character appears twice on Alphabet tape. |
| 00000.100 | Over 63 characters in Alphabet |
| 00000.101 | Haltcode appears within the Alphabet, or Alphabet punched on more than one line. |
| 00000.110 | Character store full. |

9. STORE USED.

GENSORT occupies locations 8-610, including a 256 location array for holding tables for converting Telecode to & from Alphabetic values.

Locations 680-8179 are used for the tapes to be sorted: i.e. exactly 7500 locations. 3 characters are stored per location; each line has to be separated from the next by a newline character and each line takes an integral number of locations.

Thus a line of 6, 7, or 8 characters (excluding any newlines) will occupy 3 locations; one of 9, 10, or 11 will occupy 4, etc.

10. TIME TAKEN.

The sorting takes about $2L^2C$ instructions (there being no multiplies, divides, or long shifts), where

L is the number of lines of text

C is the average number of characters per line.

Whether or not the data is initially almost in order is irrelevant; but if many lines start with the same characters sorting will be slower than above.

Thus a tape of 20,000 characters will take about $320/C$ minutes to sort on a 903/920B. (plus peripheral time)

11. OTHER NOTES.

- 11.1. GENSORT performs NO PARITY CHECKS, although a parity error on the Alphabet tape, or on the tape to be sorted or merged will give error "0000.010" when the tape to be sorted or merged is read.
- 11.2. GENSORT does NOT check that tapes being merged are in fact already in alphabetical order. Tapes being merged should have some blanks after each newline: if previously obtained from GENSORT they will have these.
- 11.3. Lines are deleted from store as they are punched; thus to obtain a second copy of a sorted tape the unsorted tapes must be re-loaded.
- 11.4. If 2 lines are of different length but are the same up to the end of the shorter, the shorter is output first; in effect 'newline' precedes all the other characters of the alphabet.
- 11.5. Empty lines are ignored.
- 11.6. If a tape which is too long to be sorted is to be broken into sections by hand-punching haltcodes onto it, and the positions for these haltcodes are to be found by reading the tape in until error "00000.110" is given, these haltcodes should be punched several lines before the line giving the error. This is to allow space in store for the whole tape section PLUS the longest line of the tape it will be merged into.

TATDOC, 27/7/70, Binary Mode 3.

1. FUNCTION.

"TATDOC, 27/7/70, Binary Mode 3" is a program for use on 903, 905, and 920 computers, and is used to document Telecode tapes, i.e. it copies the tapes, but adds extra newline symbols at regular intervals, enabling a continuous print-up of the tape produced to be cut into pages. In addition a page number is inserted at the top of each page, and, if the tape is a SIR program or label list, or an ALGOL program, the program title will appear at the top of each page. The text to be documented may be on more than one tape (except label lists, which are always on one tape).

50 lines of source text will appear on each page of the output tape. The format of the output tape is such that (assuming 6 lines to the inch) the print-up may be cut into pages of between 10" & 13", and is thus suitable for FOOLSCAP, AMERICAN QUARTO, A4, and QUARTO paper sizes.

2. HARDWARE.

This program is written for use on an 8192-word 903, 905, or 920 computer with a punch and reader. Tape reader Mode 3 is required.

Stop-on-character readers are not required, but if they are not used, the source tapes should have several blanks after each (N).

3. RESTRICTIONS ON SOURCE TAPES.

- 3.1. They must be in 903 or 920 Telecode, each tape must start with a (N) and end with a (H).
- 3.2. No line may exceed 74 characters, excluding the (N) and any blanks or erases. (This restriction ensures that the print-up will fit onto the paper sizes mentioned in Paragraph 1, unless the (T) symbol is used, and allows a margin for binding or filing the print-up).

4. OUTPUT FORMAT.

- 4.1. The output tape will contain no blanks or erases. In particular there will be no blanks after each (N), (since it is not expected that these tapes will be read into any computer).
- 4.2. Each page of the output will have, in the top right-hand corner, one of the following:-

Programme Page N	if the source tape is a SIR program
Label List Page N	if the source tape is a SIR label list
Page N	if the source tape is an ALGOL program or general text

where N is an integer starting at 1.

- 4.3. Except if the source tape is general text, each page of the output will be titled with the title of the source tape, underlined. The brackets around a SIR title, or the "," at the end of an ALGOL title, will not be copied. (The title will still appear as the first line of text on the first page in its normal form).
- 4.4. If an empty line (i.e. 2 adjacent (N)s) on the source tape coincides with the top of a page on the output tape, it will be ignored.
- 4.5. As a consequence of 4.2 & 4.3 above, the output tape will not be acceptable to SIR or ALGOL. As a consequence of 4.4 it is not possible to write a program to convert the output tape exactly back to its original form.

5. ENTRY POINTS & METHOD OF USE.

- 5.1. The following operations all use tape reader Mode 3.
- 5.2. "TATDOC 27/7/70 Binary Mode 3" is loaded by initial instructions (8181).
- 5.3. Select the output Telecode by triggering at:-
15 for 903/ISO code output
16 for 920 code output.
(These entry points merely set a flag; they do not cause tape to be read. If neither is used, 903/ISO code is presumed).

- 5.4. To document a SIR or ALGOL program, or general text, enter the first tape at 8, and, if the source text is on more than 1 tape, enter the remaining tapes at 9. These tapes will be processed line-by-line. After the last tape is completed, enter at 10. (This causes the end of the last page and a final (H) to be punched).
- 5.5. To document a SIR label list, enter it at 11. (No subsequent entries are required).
- 5.6. Having obtained a documented version of the source tape, DO NOT now destroy the source tape. (See Paragraph 4.5). It is recommended, however, that once the documented version has been printed up, it is destroyed, since it is of no further use and could be confused with the source tape.

6. TOP-AND-TAIL FACILITY (whence T.A.T.-DOC).

- 6.1. TATDOC contains a facility, which must be distinguished from that of Paragraph 4.3, whereby a character string, of up to 50 characters, can be printed near the centre of the top & bottom of each page, e.g.:-

"UNCLASSIFIED" or
"SUPERIOR SOFTWARE LTD."

- 6.2. To use this facility, punch the string on a short tape, preceded by a (N) and followed by a (H). It should not start or end with a (S), or contain (N) or (H). I.e. for the first of the above examples, punch:-

(N)UNCLASSIFIED(H)

- 6.3. Read the string into store (after loading TATDOC) by entering at 12.
- 6.4. To document tapes using the stored heading, start at 13 (instead of 8) for programs or general text, and at 14 (instead of 11) for label lists. (Entries 9 & 10 remain unchanged).
- 6.5. Note that when this facility is used the string at the bottom of each page will not fit onto Quarto paper.

6. ERROR INDICATIONS.

- 6.1. The loader of TATDOC performs a sum-check. If, when loading TATDOC (as in Paragraph 5.2) continuous punching occurs, it has mis-read.
- 6.2. TATDOC can give the following error indications in legible tape form:-

CH I/O ERROR 1. Parity error on tape being read.

CH I/O ERROR 2. First character on tape being read is not a newline, linefeed, or carriage return symbol.

CH I/O ERROR 3. When reading or punching in 920 Telecode; a character cannot be converted to or from internal code.

DOCUMENTATION ERROR 1. Entry 9 or 10 has been used before entry 8, 11, 13, or 14; or entry 13 or 14 has been used before entry 12.

DOCUMENTATION ERROR 2. Character string being read in at 12 is too long or contains an illegal separator.

DOCUMENTATION ERROR 3. Title contains lower-case letters or an illegal separator, or a tape entered at 11 or 14 has no title.

DOCUMENTATION ERROR 4. A line of source text is too wide. (See Paragraph 3.2).

DOCUMENTATION ERROR 5. Indicates that a haltcode has been found in the wrong position; probably caused by loading a program at 11 or a label list at 8.

DUMP 15/4/70 Binang Mote 3

1. FUNCTION

'DUMP 15/4/70' is a program for producing a store-dump of all non-zero locations from 2 to 8000 inclusive.

The tape produced is Mode 3 Binary, suitable for input at initial instructions, with a clearest at the front of the tape; and is in "A.C.D. 900-Series 18-bit Binary Tape Format 1/4/70".

2. HARDWARE

The program is written for use on an 8192-word 903, 905, or 910 computer with a punch and reader. Mode 3 (or 2) input is required.

Stop-on-character readers of any speed and non-stop-on-character readers (of up to at least 4000 c/s) may be used.

3. METHOD OF USE

3.1. 'DUMP 15/4/70 Binary Mode 3' is read in at initial instructions in Mode 3. (or 2)

It has a self-triggering loader, so the dumped tape is produced immediately.

3.2. To produce another dump, repeat step 3.1.

Alternatively 'DUMP' may be retriggered at 8001, (&17501).

3.3. To read dumped tapes back into the computer, they should be loaded under initial instructions, in Mode 3 (or 2)

4. ERROR INDICATIONS.

Both the loader of 'DUMP' and the loader in the binary tape it produces perform a sum-check.

If this fails continuous output occurs on the punch.

5. TYPICAL USES

Two typical uses of 'DUMP' are :-

- 5.1. User 'A' is running a long calculation, and user 'B' wishes to perform a shorter, more urgent, calculation, so user 'A' stops the computer, records the contents of the A & Q registers & the top-level B-register (loc. 1) & SCR (loc. 0) then dumps the store. User 'B' then uses the computer. Finally 'A' reloads his dumped programme, resets the registers & continues.
- 5.2. To produce a binary tape of a 'composite' programme.
Examples :-
 1. If a programme is written in 'Algol'; translated by the Algol translator; and loaded by the Algol interpreter; then DUMP may be used to obtain a single tape of the Algol interpreter and the programme. Thus the operating instructions of this single tape need make no reference to Algol.
 2. If a programme is written to perform a given function, on a machine other than a 903/905/920, a tape may be produced to perform the same function on the 903/905/920, if a suitable 'simulation' programme is available, by loading the simulator, loading the existing programme into the simulator & dumping the result.

6. CAUTIONS

- 6.1. For several reasons it is far better to make Binary tapes of SIR Telecode tapes using 2-PASS SIR, rather than 1-PASS SIR and DUMP.
- 6.2. Since DUMP punches all non-zero locations it is advisable to clear the store before loading any tapes which are going to be dumped.

Clear Stone

FUNCTION

To set locations 2 to 8177 of store to +0. (Note that the locations NOT cleaned are used when the next tape is read in, anyway). The main use of this programme is in conjunction with the 'DUMP' programme.

HARDWARE

"Clear Store" is written for use on an 8192-word 903, 905, or 920.
"Mode 3" input is required.

METHOD OF USE

Set the mode switch, where applicable, to Mode 3.

Read in "clear store" by triggering at initial instructions.

When the reader stops, the store will be cleaned in under one second.

900 VERIFY 2/5/70 Binary Mode 3.

1. FUNCTION.

'900 VERIFY' compares a binary tape with the contents of store.

Thus it may be used :-

- (i) To check that a store-dump has been punched correctly,
- (ii) To check that a binary tape has been correctly read into store,
- (iii) To check for program corruption, as a consequence of faulty hardware or software.

The binary tapes used must be in "A.C.D. 900-Series 18-Bit Binary Tape Format 1/4-70" as described elsewhere.

Error indications are given in 900-Series Telecode.

2. HARDWARE.

'900 VERIFY' is written for use on an 8192-word (or more) 903, 905, or 920. computer, with a punch and reader. Reader mode 3 is required.

Stop-on-character readers are NOT essential, but, if they are not used, only the first error indication given will be meaningful.

3. METHOD OF USE

- 3.1. The following operations are all performed using reader mode 3.
- 3.2. Read '900 VERIFY' into store using initial instructions (8181).
- 3.3. Place the binary tape, to be compared with the store, in the reader. If it starts with a clear-store, pull the clear-store through the reader by hand.
- 3.4. Trigger VERIFY at 8001 (817501).
- 3.5. If the tape and the store agree, the tape will stop a few characters from the end, and no output will occur.
- 3.6. Further tapes may be checked by repeating steps 3.3. to 3.5.

4. ERROR INDICATIONS.

- 4.1. The loader of '900 VERIFY' performs a sum-check; If, when reading in VERIFY (as in Para. 3.2.), continuous punching occurs, it has mis-read.
- 4.2. VERIFY checks that the loader of the binary tape being checked has the same sum-check as that of a standard-format tape. If it does not,
"EL" (L for Loader)
is punched, and checking stops.

This error will be given, for example, if there is a clear-store on the tape which has not been pulled through the reader by hand.

- 4.3. If VERIFY finds that a word on the tape and a word in store disagree, then it will punch, on one line, in octal:-

The address of the word;
The word on the tape;
The word in the store.

Checking continues, but further errors may be given spuriously if stop-on-character readers are not used.

- 4.4. If there is a character missing or a character extra on the tape, or if a directive on the tape is mis-punched, or if the tape is not in standard format, then VERIFY may be unable to proceed and will punch

"EF". (F for Format)

This error may also be given spuriously, after an error of the type in Para. 4.3, if stop-on-character readers are not used.

- 4.5. VERIFY checks that the sum-check at the end of the tape being checked agrees with the words previously read in. If this is not so

"ES" (S for Sum-check)

will be punched.

4.6. VERIFY occupies locations 8001 to 8179. It may therefore over-write part of the store which the user is trying to check. This does not prevent the user checking the rest of the store.

If the tape being checked contains words which would load into the area computed by VERIFY, error indications of the form described in Para. 4.3. will NOT be given, but a warning message

"EA" (A for Address)

will be given on reaching the end of the tape, indicating that the user's program is no longer intact in store.

5. WORKSPACES.

If VERIFY is used to check for store corruption AFTER a program has been run, then the workspaces will NOT appear spuriously in this list PROVIDED that the binary tape was assembled by 2-PASS SIR, that the 'zeros for skips' option was not used, and that the workspaces are punched as skips, (not zeros,) on the telecode source tape.

Autostart

FUNCTION

To enable a program held in store to be started, at location 8, when the computer is switched on in the 'AUTO' mode or when no control unit is attached.

HARDWARE

"Autostart" is written for any 903/905/920 with mode 3 reader input.

METHOD OF USE

Set the mode switch, where applicable, to Mode 3.

Read in "Autostart" by initial instructions.

The computer will at once start to obey the program in store at location 8.

If the computer is now switched off, and set to 'AUTO' mode, or if the control unit is removed (and replaced, on 920B, by an autostart plug); the computer will again enter the program in store at 8 when subsequently switched on.

900 AMEND, 20/7/70, 900-Series Telecode

1. FUNCTION

To read a 900-Series teletype program correction tape directly into the store of a 900 series 18 bit computer.

This program is particularly useful for correcting programs which are too large to be corrected by using 1-PASS SIR.

It performs a similar function to 1-PASS SIR, but it is considerably smaller and provides correspondingly fewer facilities.

2. FORMAT OF CORRECTION TAPES

The correction tapes must be in 900-Series code and may make use of the following SIR facilities :-

Comments (including titles)
Patches of the form $\uparrow N$ } where N is an
Skips of the form $> N$ } unsigned integer
Integers in the range +131071 to -131071
Octal numbers of up to 6 digits preceded by 'Q'
Instructions with absolute addresses, e.g. /4 3000
Instructions with relative addresses, e.g. 8 j-5

The function and address of an instruction must be separated by one or more (S) or (T).

Any 2 of the above-listed items must be separated by one or more (N), (S), or (T).

Each tape must end with a (H) preceded by one or more (N), (S), or (T).

No blanks are required using 950 c/s readers.
Blanks and erases are ignored.

3. EXAMPLE OF CORRECTION TAPE

((FRIG TO AVOID S/N INPUT, 9/10/69))

↑3027 (SNIN+6)

8 6144

↑6144

4 2122 (ARRAY)

9 j+3

4 6151

8 j+2

4 6152

5 2123 (ARRAY+1)

8 3028

+0

&400000

4. METHOD OF USE

4.1. "900 AMEND, 20/7/70, 900-Series Telecode" is a SIR program. Since it will usually be used to modify large programs it should be assembled by 2-PASS SIR, either as part of the program which it will be used to correct; or independently but so as to occupy an area of store not used by the program being corrected.

4.2. To read in a correction tape enter at the address of the global label 'AMEND', using tape reader mode 3.

5. ERROR INDICATIONS

5.1. If a parity error is detected in a correction tape, the character

000.00001

will be punched continuously.

5.2. If any other error is detected in a correction tape, the character

000.00111

will be punched continuously.

6. WARNING.

'AMEND' permits program to be placed in ANY store location from 0 to 131071. It is the user's responsibility to consider the effects of placing program in locations 0 to 7 or within 'AMEND' itself.

7. STORE USED.

195 Consecutive locations and 29 literals

900 QIN1 20/7/70 900-Series Telecode

NOTE.

The "DIRECT" TS-CUM-QINI and T6-CUM-QOUTI described in this book should not be confused with the "INDIRECT" versions issued as one tape, "QS I/O", in the SIR subroutines package.

The "DIRECT" versions input and output via the punch and reader directly, in 900-Series Telecode. They are thus inflexible, but occupy little store. They are provided as a debugging aid for programs occupying most of the store; or for programs which are to be run a few times but not to be kept.

The "INDIRECT" versions input and output via character input and output subroutines. They should be used in any program requiring single-length number input and output which is to be kept. Whilst the indirect versions (plus a suitable character input and output subroutine) occupy more store in total than the direct versions, the resulting program can easily be modified at a later date to work in any Telecode via any peripheral.

1. FUNCTION

TS-CUM-QIN is a SIR subroutine punchlet in 900-Series code to read one integer or fraction from a 900-Series code data tape, and place it in the accumulator. It is suitable for use on any priority level.

2. HARDWARE

TS-CUM-QIN reads data tapes with "Mode 3" input. Stop-on-character readers of any speed and non-stop-on-character readers (of at least 250 c/s) may be used.

3. STORE USED

116	consecutive locations
13	literals

4. ACCURACY

Integers are stored exactly.

Fractions are rounded towards zero with a maximum error of 2^{-17} .

5. METHOD OF USE

T5-COM-QIN is entered by the following instructions:-

11 QIN1
8 QIN1+1

or :- 11 T5
8 T5+1

or :- 11 INPUTL
8 INPUTE

(The first of these 3 forms is recommended, the others are only provided for compatibility with old programs)

6. DATA TAPE

Integers should be punched

+ or -

a sequence of 1 to 6 digits

any non-digit (e.g. (N), (T), or (S))

Note that -131072 cannot be input.

Fractions should be punched

+ or - followed by .

a sequence of 1 to 6 digits

any non-digit (e.g. (N), (T) or (S))

Blanks & erases will be ignored everywhere, as will the characters preceding the + or -

when using non-stop-or-character readers it may be necessary to punch blanks after each item of data. (This will depend upon the reader speed, the type of computer, and the user's programme).

6A. EXAMPLE OF DATA TAPE

+3 SAMPLES, 12/10/68.

A	+678	B	+005	C	-999999
A	+713	B	+040	C	-995000
A	+82	B	+010	C	+000010

This example would be read by entering
TS-CUM-QIN 10 times.

7. ERROR INDICATIONS

TS-CUM-QIN will punch a character continuously if any of the following errors is found:

The first character after the + or - is not a digit or 0

The first character after a 0 is not a digit

More than 6 digits

Integer outside ± 131071 .

Parity error

900. TG-CUM-QOUT 22/10/68 900-Series Telecode

NOTE.

The "DIRECT" TS-CUM-QINI and TG-CUM-QOUT1 described in this book should not be confused with the "INDIRECT" versions issued as one tape, "QS I/O", in the SIR subroutines package.

The "DIRECT" versions input and output via the punch and reader directly, in 900-Series Telecode. They are thus inflexible, but occupy little store. They are provided as a debugging aid for programs occupying most of the store; or for programs which are to be run a few times but not to be kept.

The "INDIRECT" versions input and output via character input and output subroutines. They should be used in any program requiring single-length number input and output which is to be kept. Whilst the indirect versions (plus a suitable character input and output subroutine) occupy more store in total than the direct versions, the resulting program can easily be modified at a later date to work in any Telecode via any peripheral.

1. FUNCTION

T6-CUM-QOUT is a SIR subroutine punched in 900-Series code, to punch the contents of the accumulator as an integer, fraction, or octal word in 900-Series code, in a format indicated by a parameter word. It is suitable for use on any priority level.

2. STORE USED.

126 consecutive locations
17 literals.

3. METHOD OF USE

The entry instructions are as follows:-

To print the contents of the accumulator as:-

An integer	A fraction	An Octal word
11 QOUT1 8 QOUT1+1	11 QOUT1 8 QOUT1+2	11 QOUT1 8 QOUT1+3
or	or	or
11 T6 8 T6+1	11 T6 8 T6+2	11 T6 8 T6+3
or	or	or
11 PRTL 8 PRTEI	11 PRTL 8 PRTEF	11 PRTL 8 PRTEO

(The upper of these 3 forms is recommended, the others are only provided for compatibility with old programmes) T6-1

In all cases the 2 entry instructions must be followed by the parameter word (see below).

The subroutine exits to the location after the parameter word.

4. THE NUMBER -131072

The integer -131072, and FRACTION -1, will be punched in OCTAL, (irrespective of the entry instructions used) as &400000.

5. OUTPUT FORMAT & PARAMETER WORD

The parameter word is punched in the form of an absolute addressed instruction, e.g.

/0 0
or 2 3.

If the modifier bit '/' is present, the output will start with a (N). If this is not present, output will start with (S)(S).

Below, { F refers to the function digits of the parameter
 { N " " " address " " " " }

5.1. OCTALS will be punched in the usual format:-

$\left\{ \begin{array}{l} \textcircled{N} \text{ or } \textcircled{S} \textcircled{S} \\ \& \\ \text{exactly 6 digits} \end{array} \right.$

E.G. $\textcircled{N} \& 123456$.

F & N are ignored

5.2. FRACTIONS will effectively be MULTIPLIED by 10^N before printing, and the least significant F digits will be omitted, I.E. the following will be punched:-

$\left\{ \begin{array}{l} \textcircled{N} \text{ or } \textcircled{S} \textcircled{S} \\ + \text{ or } - \\ N \text{ digits} \quad (\text{with 'leading' zeros replaced by SPACES}) \\ \text{Decimal point} \quad (\text{unless followed by no digits}) \\ (6-N)-F \text{ digits} \end{array} \right.$

Thus for the usual fraction format, e.g. $\textcircled{N} + \cdot 123456$, use $F=N=0$.

5.3. INTEGERS will effectively be DIVIDED by 10^N before printing, and the least significant F digits will be omitted, I.E. the following will be punched:-

$\left\{ \begin{array}{l} \textcircled{N} \text{ or } \textcircled{S} \textcircled{S} \\ + \text{ or } - \\ (6-N) \text{ digits} \quad (\text{with 'leading' zeros OMITTED}) \\ \text{Decimal point} \quad (\text{unless followed by no digits}) \\ N-F \text{ digits} \end{array} \right.$

Thus for the usual integer format, e.g. $\textcircled{N} + 123456$, use $F=N=0$.

6. EXAMPLE OF USE OF PARAMETER

Given that the accumulator contains an angle in degrees scaled by 180° (so that $+0.25 = +45^\circ$) to print the angle on a new line, in degrees, to one decimal place :-

```
12 +018
11 QOUT
8  QOUT1+2
  /2 3
```

```
F=2
+0180000
  ↗
N=3
```

```
or
12 +18000
11 QOUT
8  QOUT1+1
  /1 2
```

```
F=1
+18000
  ↗
N=2
```

It is recommended that the parameter word is punched in the "address" column with F & N separated by one (5); so that it is easily recognised as a parameter.

7. WARNING

Although this subroutine is particularly suited to PRINTING quantities scaled by powers of 10, it is usually more accurate to STORE and OPERATE upon quantities scaled by powers of 2 or scaled by their maximum values, (as in the above example).

8. ERROR INDICATIONS

T6-CUM-ROUT will punch a character continuously if entered at the integer or fraction entry point and a meaningless parameter is used, i.e. if

$$N > 6$$

$$N < F \quad \text{for integers}$$

$$6 - N < F \quad \text{for fractions}$$

9. ACCURACY

Integers, printed with $F=0$, are printed exactly.

Fractions, " " $F=0$, " rounded

towards zero, & hence contain a maximum error of 10^{-6} .

When digits are omitted by setting $F > 0$, the printed digits are not changed, i.e. rounding is towards zero. For example $+49$ would be printed as $+4$ if $F=1$.

900 STORE PRINT, 3/8/70, 900-Series Teletype

FUNCTION

To assist in the checking of programmes by printing the content of any number of locations in the store in 900-Series Teletype. The print out can be obtained as :-

- (a) quasi-instruction form
- (b) fraction
- (c) integer
- (d) octal form

STORE USED

266 consecutive locations & 29 literals

DATA TAPE

A data tape is required and should be punched in 900-Series Teletype. It should consist of one line for each area of store to be printed. Each line should contain 2 unsigned integers and a letter (separated by spaces or tabs), the integers being the first and last locations to be punched, and the letter being as follows:-

- (a) an Q when quasi-instruction output is required
- (b) an F when fraction output is required
- (c) an I when integer output is required
- (d) a B when octal output form is required

The data tape should end with a halt-code. Blanks, erases, and car. ret. are ignored everywhere on the data tape.

For example :-

(N) 8 (S) 300 (S) Q
(N) 301 (S) 350 (S) I
(N) (H)

would cause locations 8 to 300 to be printed as instructions, and 301 to 350 as integers.

If a single location is to be printed, the second integer may be omitted.

If stop-on-character readers are not used, some blanks should be punched, either after each format letter or after each newline.

METHOD OF USE

900 STORE PRINT is a 900-Series Telecode tape. Load it into an unused area of store using SIR. Load the data tape by entering STORE PRINT at the label QCHECK or C2 (with the reader in Mode 3). STORE PRINT will read each entry on the data tape & punch the specified output.

When the output has finished the programme comes to a dynamic stop. It may be re-triggered at QCHECK or C2.

OUTPUT FORMAT

Each requested location is printed on one line preceded by its address in round brackets. A "patch" is punched at the start of the output and before each subsequent area of store, unless adjacent to the previous area; and the output ends with a haltcode.

The output is thus suitable for re-input by "AMEND" (unless fractional format has been used) or by 1-PASS or 2-PASS SIR, (provided blanks are inserted after each newline).

ERROR INDICATIONS

Continuous output will be given on the punch if:-

- (i) A parity error is found on the data tape
- (ii) The haltcode of the data tape appears inside or after either of the integers on a line (probably caused by a contextual error elsewhere on the data tape).

MONITOR, 12/3/71, 900-Series Telecode.

Chapter 1: INTRODUCTION

1.1 Purpose.

MONITOR is a program testing aid which gives the facility of a temporary hold-up at any specified point in the program under test while the contents of the accumulator, Q register, and any specified core locations are output for printing in any combination of integer, octal and instruction modes.

1.2 Form of Distribution.

The program is distributed as a tape for input by SIR.

1.3 Method of Use.

A parameter tape is read in by MONITOR to specify the points at which a hold-up is required and the locations and modes then to be output. The program under test is entered and output occurs whenever a monitor point is reached.

An alternative entry point to MONITOR cancels all monitor points previously set.

MONITOR may run at any program level and in any store module. Monitor points must be in the same store module as MONITOR.

1.4 Telecodes.

The program is distributed as a 900-Series Telecode tape. The parameter tape must be in 900-Series Telecode; the output of monitor points will also be in 900-Series Telecode.

1.5 Compatibility.

This program WILL NOT function properly on a 920A computer, (although it may be possible to use it in a limited manner.)

Chapter 2: FUNCTIONS

2.1 Addresses.

All addresses refer to a 64K store and where necessary are taken as modulo 64K. Addresses are always relative to the beginning of the store module in which MONITOR is stored. If MONITOR is in the first module, addresses are absolute. However, if MONITOR is in the second module, then it treats the first module as addresses 56K to 64K, the second module as 0 to 8K, the third module as 8K to 16K, and so on.

2.2 Parameter List.

A parameter list contains a number of lines which are of four forms.

2.2.1 a, b

This form specifies a monitor point.

a is an integer giving the address of a monitor point relative to the beginning of the store module in which MONITOR is stored. $a \leq 8190$. (MONITOR uses $a=8191$ privately to mark the end of the parameter list.)

b is an integer specifying the output modes.

(see 2.4)

2.2.2 a,

This is similar to 2.2.1 but b is given the same value as the previous monitor point. $b = 0$ if this form is used for the first point.

2.2.3 $c_1 \pm c_2 \pm \dots \pm c_m$

This form specifies the address of one store location whose contents are output whenever its associated monitor point is reached.

If $m = 1$ the address is directly defined.

If $m \geq 2$ the address is indirectly defined as follows:

Take C_1 and treat as an address.

Take the contents of this address, add or subtract C_2 and treat as an address.

Take the contents of this address, add or subtract C_3 and treat as an address.

Take the contents of this address, add or subtract c_n and treat as the defined address.

Each $c \leq 65535$

N.B. This evaluation is done each time that the monitor point is reached during the running of the test program, and may vary from one time to the next.

$$2.2.4 \quad c_1 \pm c_2 \pm \dots \pm c_m / d_1 \pm d_2 \pm \dots \pm d_n$$

This form specifies the first and last address of a block of consecutive store addresses whose contents are output whenever its associated monitor point is reached.

Either address can be directly or indirectly defined as in 2.2.3. The order is immaterial; MONITOR distinguishes the larger from the smaller.

Each monitor point is specified by forms 2.2.1 or 2.2.2 and its associated core locations follow with any number (including zero) of forms 2.2.3 or 2.3.4. Every form must occur on a separate line. Empty lines are ignored. The list ends with a halt character. Up to 10 monitor points may be specified in a parameter list.

2.3 Character Set for Parameter List.

Meaningful characters are 0 to 9 (L) (H) +, -, , /

(S) (T) (R) (P) (E) cr are ignored

All other characters are errors.

2.4 Output

At each monitor point the form of output is determined by the value of the integer b specified in forms 2.2.1 or 2.2.2. This integer is regarded as a five bit binary number, and the bits have the following significance.

Bit	Decimal Equivalent	Set	Clear
1	1	Layout using spaces.	Layout using tabs.
2	2	Output A and Q registers.	No output of A and Q registers.
3	4	Decimal Output.	No decimal output.
4	8	Octal output.	No Octal output.
5	16	Instruction output	No instruction output.

2.4.1 +1

A line of output may contain a store address and the contents of this address in up to three different modes. It is convenient for the user to produce a hard copy aligned in vertical columns. If bit 1=0, this alignment is done by outputting a tab character where appropriate, and a new line is started with a single \textcircled{L} character. This is the simplest and most convenient output where print-up is done on a Friden Flexowriter.

If bit 1 = 1, the alignment is done by outputting the required number of space characters, and new lines begin with a cr lf blank sequence. This method is required for a Teletype Model 33.

2.4.2 +2

If bit 2 = 1, the contents of the accumulator and the Q Register are output in addition to the store locations specified. Labels A and Q are output for identification.

2.4.3 +4

If bit 3 = 1, output is in the form of a signed decimal integer with non-significant zeros suppressed. Values lie in the range - 131072 to + 131071.

2.4.4 +8

If bit 4 = 1, output is in the form of an unsigned 6 digit octal number without zero suppression.

2.4.5 +16

If bit 5=1, output is in instruction form. Bit 18 of the contents of the location is output as /, bits 14-17 as the function number and bits 1-13 as an unsigned decimal address with zero suppression.

The address of a single location and the first address of a block of locations is always output as a label, and thereafter every address which is a multiple of 5 is labelled. Indirectly defined addresses are evaluated at the time of output, according to the method given in 2.2.3, but are not distinguished in the actual output from directly defined addresses.

Note that any combination of Integer, octal and instruction modes may be selected.

e. g. $b = 22$, i. e. $2 + 4 + 16$ will give output using tabs of A, Q and specified store locations in decimal and instruction forms.

The output for each monitor point always commences with * a where a is the monitor point address.

It is not necessary to output the contents of any registers or store locations at a monitor point. In this case the output of the monitor point address is an indication that the test program has reached the monitor point. This can be useful for investigating the continuity of a program.

2.5 Restore.

A parameter list can be cancelled and all monitor points deleted by entering α QMON+1. The test program is restored to its original state. A new parameter list can now be read in and further monitoring performed. Restore must not be used if an error is detected during the input of a parameter tape.

2.6 Example

2.6.1 Parameter tape.

36, 27
594
1586/1594

38, 30
1000+0/1000+18
1594/1586

(H)

2.6.2 Output tape.

(Location 1000 contains the value 54)

*36

A	000000	0	0
0	202345	8	1253
594	000062	0	50
1586	000033	0	27
	000000	0	0
	000146	0	102
	000016	0	14
1590	000017	0	15
	003067	0	1591
	777772	/15	8186
	000001	0	1
	000062	0	50

*38

A	6	000006	0	6
0	66788	202344	8	1252
54	122877	357775	14	8189
55	49750	141126	6	598
	16980	041124	2	596
	41550	121116	5	590
	589	001115	0	589
	-98106	500306	/4	198
60	590	001116	0	590
	-16384	740000	/14	0
	49751	141127	6	599
	41553	121121	5	593
	32999	100347	4	231
65	8792	021130	1	600
	57413	160105	7	69
	33359	101117	4	591
	65607	200107	8	71
	33359	101117	4	591
70	16980	041124	2	596
	596	001124	0	596
	114690	340002	14	2
1586	30	000036	0	30
	0	000000	0	0
	109	000155	0	109
	14	000016	0	14
1590	15	000017	0	15
	1591	003067	0	1591
	-6	777772	/15	8186
	0	000000	0	0
	50	000062	0	50

MON-6

Chapter 3: ERRORS

Whenever an error is detected, a message is printed on a new line of the teleprinter in the form ERROR n where n is the error number. If this output is diverted to the punch, it will appear as 1" of run-out followed by cr lf (R) ERROR (S) n. All errors are detected during the reading of the parameter tape and cause reading to stop and MONITOR to end at a dynamic stop. The test program is not affected in any way until the complete parameter tape has been read and accepted as free from error. If an error is detected, a new parameter tape must be prepared and read in from the beginning.

No MONITOR errors can occur while a test program is running and being monitored.

The following errors are detected.

- n = 1 odd parity characters
- 2 Unused characters.
- 3 Any c or d \geq 65536 in 2. 2. 3 or 2. 2. 4
- 4 Integer missing
- 5 Impermissible sequence of separators + - , /
- 6 b \geq 32 in 2. 2. 1
- 7 Number of monitor points \geq 11
- 8 a \geq 8191 in 2. 2. 1 or 2. 2. 2

It is not regarded as an error to specify the same monitor point more than once, although there is little advantage in doing so. MONITOR would be entered separately for each occurrence and the appropriate output generated. However, entry QMONH would not restore the test program to its original form correctly.

Chapter 4: METHOD USED, & SOME RESTRICTIONS ON MONITOR POINTS.

At QMON the parameter tape is read, and analysed and the information stored in a buffer (see 6). If no errors are detected, the instruction in the test program at each monitor point is preserved in the buffer and replaced by a function 8 instruction causing an entry into MONITOR. Each monitor point has a different entry into MONITOR. At entry the A and Q registers and locations 1, 3, 5, & 7 of the store module containing MONITOR are preserved. The required information is output, A, Q, and locations 1, 3, 5, and 7 are restored and the true instruction of the test program is obeyed from within MONITOR workspace. If this instruction causes a jump, then the test program is re-entered at the jump point; otherwise the following instruction in MONITOR workspace is a function 8 instruction causing re-entry to the test program at the instruction immediately following the monitor point.

Note that it is inadvisable to monitor a function 11 instruction because the SCR value stored refers to MONITOR workspace and not the true position in the test program.

Monitor points can be on more than one level of program provided they are chosen so that MONITOR is never entered at one level when it has been interrupted in monitoring at a lower level.

For a large program operating at several levels or in several store modules, several independent MONITOR routines may be used to cover different modules or levels. Monitoring at differing levels may lead to a certain amount of jumbling in the output.

At QMON+, all the monitor points in the buffer are scanned and the preserved instructions of the test program restored.

If some monitor points are not on top level, the programs on higher levels than those monitor points must not require their B-registers to be retained from a terminate to their next interrupt, (since, on leaving MONITOR, the B-register may be set back to an EARLIER value).

IF MONITOR IS NOT IN THE FIRST STORE MODULE, monitor points should not be placed where the current-level B-register is of significance.

Chapter 5: OPERATING INSTRUCTIONS

The program to be monitored is read into store in the normal manner (e.g. if it is a binary tape, read it in by initial instructions; if it is a SIR tape, either load it by 1-PASS SIR or assemble it by 2-PASS SIR for loading by initial instructions.)

Read MONITOR into an unused area of store (either load it by 1-PASS SIR or assemble it by 2-PASS SIR for loading by initial instructions.)

If the program to be monitored, and MONITOR, are both loaded by 1-PASS SIR, they must each be separately terminated by a %. If both are assembled by 2-PASS SIR they must be assembled separately. This is to ensure that only the literals of MONITOR follow MONITOR; (see Chapter 6).

Load the parameter tape (with the tape reader in Mode 3) at the label QMON. The tape will be read and checked until (H) is read, when MONITOR ends at a dynamic stop.

Enter test program, which will run at normal speed, except for hold-up during MONITOR output whenever a monitor point is reached.

Enter MONITOR at QMON+1 to delete all monitor points and restore the test program to its original state: provided no errors were found on the parameter tape.

Chapter 6: STORE USED

574 consecutive locations
38 literals
612 total.

MONITOR

stores the information on the parameter tape immediately after its *literals in* store. Each monitor point specified by 2. 2. 1 or 2. 2. 2 uses three locations (for the monitor point address, the displaced instruction of the test program, and the output mode respectively). Addresses specified by 2. 2. 3 and 2. 2. 4 take m and $m+n$ locations respectively. One location is used finally to mark the end of the parameter list.

Chapter 7: TIME TAKEN

The parameter tape is read at the speed of the reader. The monitoring output, is punched at the speed of the punch; the test program otherwise runs at normal speed. Restore time is a few millisecs.