COMPUTER MAINTENANCE

KIRBY ST

I.C.T. ATLAS COMPUTER

SUPERVISOR AND FIXED STORE ROUTINE SPECIFICATIONS

VOLUME 1

Both the Atlas computer and the routines described in this document are the result of a collaboration between Manchester University and I.C.T. Limited (formerly Ferranti Limited, Computer Department).

ATLAS SUPERVISOR AND FIXED STORE ROUTINES

This document contains specifications of the Atlas Fixed Store and Supervisory Routines.

It should be used in conjuction with the Volume which contains an overall description of the routines and their relationships to each other.

As all the routine specifications are not yet available this document will be corrected and supplemented from time to time.

2.9.63

Purpose: To co-ordinate entrances to supervisor extracode routines from extracode or interrupt routines.

Registers of Fixed Store: 41

Number of Instructions Obeyed:

Entered from extracode: 16 or 17

Entered from interrupts: First entry: 12 on interrupt control +8 on extracode control. Subsequent entries 9 or 10 on interrupt control

Subsidiary Store:

"Main Program Controls"

Two half words holding main program controls when an SER is being obeyed.

0 = B126 of main program
0.4 = $\frac{1}{2}\lfloor M/E \rfloor$ - V6 of main program

"Current SER base"

One half word recording the SER queue to which the current SER was entered in digits 1, 0.

Digits 1 0 =	0 0	Entered from extracode
	1 0	Top priority queue
	0 1	Tape queue
	1 1	Slow queue

"Slow SER queue"

2L + 1 half words where L is the maximum number of permitted SER in the slow queue.

"Tape SER queue"

2M + 1 half words, where M is the maximum number of permitted SER in the tape queue

"Top priority SER queue"

2N + 1 half words, where N is the maximum number of permitted SER in the top priority queue. The leading half word in this queue is the "In Supervisor Switch"

"Current SER Entry Address"

One half word holding address of entry or for re-entry to the current SER in digits 23 - 3

Digit 0 = 0 Record only B100 on "Halt SER"
= 1 Record B100 - 104 on"Halt SER"

Parameters Used: (1) to (14)

(1) = Entry address from extracode
(2) = Entry address from interrupt for top priority queue
(3) = Entry address from interrupt for tape queue
(4) = Entry address from interrupt for slow queue
(5) = Address of "Current SER Entry Address"
(6) = Address of "Main Program Controls"
(7) = Address of "Current SER base"
(8) = Address of "Slow queue" relative to the start of subsidiary stor
(9) = Address of "Tape queue" relative to the start of subsidiary stor
(10) = Address of "Top priority queue" relative to the start of subsidiary store. This is also the address of the "In Supervisor Switch.
(14) = Instruction Counter Storage Register

Connections with other Routines:

Entry: a) At (1) in extracode control with B96 = Entry address E. Supervisor routine entered at E, and main program subsequently resumed in Main Control

b) At 1(1) in extracode control with B96 = Entry address E. B97 = Resumption Address R. As (a) except that main program is resumed in extracode control at R or in main control if B97 odd. (this case not at R)

c) At (2), (3), (4) in interrupt control for entry to top priority, tape, slow queue respectively. Entered with
B111 = Entry information
B112 = Entry address

Working Space: B100, 101, 102

Notes:

1. (1) is entry A of the co-ordinator. (2), (3), (4) are entries B, C, D of the co-ordinator to "long interrupt routine".
2. If an SER is being obeyed on entries (2), (3), (4) the requests are queued.
3. On entry at 1(1), B96, 97 remain unaltered. B96 is transferred to the "Current SER entry address".
4. When an SER is entered from interrupt routines, B111 is copied to B100 and B112 transferred to the "Current SER entry address".
5. When an SER initiated either by an interrupt routine or by an extracode is entered, the Instruction Counter Clock value is preserved.
6. The "In Supervisor Switch" reads
   -0.3 when no SER is active
   +0.3 when an SER has been initiated but not entered due to further interrupts
   +N.0 ($N \geq 0$) when an SER is being obeyed.

Purpose: To select next action on completion of an SER, i.e. to continue at address specified by Program Scan Exit if this is set, otherwise to initiate a new SER if any is queued, or to resume main program if not.

Registers of Fixed Store: 27

Number of Instructions Obeyed:

5 or 7 if Program Scan Exit set.
8 if SER queues non-empty or Program Change Marker set.
10 if SER queues empty, Program Change Marker not set and current program uses program branching.
22 otherwise (normal return to main program).
Interrupts are inhibited for a maximum of 19 instructions

Subsidiary Store:

"Program Scan Exit"

One half-word, containing zero normally. If non-zero on exit from an SER, the SER is continued at the address in "Program Scan Exit". In this continuation, if digits 1, 0 are zero, the SER base is unchanged; otherwise the base is set from these digits.

"Program Change Marker"

One half- word, normally zero, set non-zero if any main program has been halted or made free by SERs.

"Program Branch Indicator"

One half-word, normally negative, set positive if the current program uses program branching.

"Supervisor Exit Link"

One half-word, normally containing the value of (18) of R202 which can be set otherwise if required to cause a special exit from supervisor when all current SERs have been executed.

Parameters used: (1) to (18)

(1) = Entry point to R202
(2) = Address of "Top priority SER queue" relative to start of subsidiary store (10/201)
(3) = Address of "Tape SER queue" relative to start of subsidiary store (9/201)
(4) = Address of "Slow SER queue" relative to start of subsidiary store (8/201)
(5) = Address of "Program Scan Exit"
(6) = Address of "Program Change Marker"
(7) = Address of "Program Branch Indicator"
(8) = Address of "Main Program Controls" (6/201)
(9) = Address of Current SER Entry Address (5/201)
(10) = Entry address of R207, "Select Main Program" (1/207)
(11) = Address of "Instruction Counter Storage Register" (14/201)
(12) = Entry address of "Select next program branch" (1/224)
(13) = Entry address of R211, "Resume SER" (1/211)
(16) = Address of "Supervisor Exit Link"
(18) = Address within R202 - normal value of "Supervisor Exit Link".

Connections with other routines:

Entered: At (1) at conclusion of all SER's in extracode control (entry G of the co-ordinator)

Exit: To one of several routines tested for in the following sequence:-

(1) If Program Scan Exit non-zero, then

a) If digits 1, 0 non-zero, to "Resume SER" to set new base and exit to specified address. B110 = Old contents of "Program Scan Exit", B119 = digit 1, 0

b) if digits 1, 0 zero, direct to specified address with base unchanged.

(2) To "Select Main Program" if SER queues non-empty or "Program Change Marker" non zero, with B109 = total no. of entries in SER queues.

(3) To "Select Next Program Branch" if the current main program uses program branching.

(4) To current main program, resetting "Main Program Controls" and the Instruction Counter.

R203: Store Location and Lock-out

Purpose: A supervisor extracode subroutine entered when peripheral transfers are to be initiated, and by various supervisor routines. Locates store blocks with labels b to b+n of the current program in the block directory, and inserts lock out digits where appropriate. According to the entry parameters and the existing lock out status of the blocks, control is returned to a specified address in extracode control, to "Program Scan", to "Halt Main Program" or to "Monitor".

Registers of Fixed Store: 102

Number of Instructions Obeyed:

Varies markedly with the contents of the block directory.
Minimum (when program uses blocks 0, 1, 2, ...N only).

18 + 28(n + 1) If no lock out required
18 + 38(n + 1) If lock out required and blocks in core store

Subsididary Store:

"Block location table"

8 successive half words set by this routine to hold the directory positions of the blocks. Available for use by other routines.

"Block directory"

Consecutive half words, one per main store block. Entries in the main program area contain:

Digits 23, 0 = 0 0 Unused entry (digits 22-1 irrelevant)
1 0 Vacated entry (digits 22-1 irrelevant)
0 1 Block in core store
1 1 Block on drum

When the entry is for a block in one level store, with digit 0 = 1
Digits 22 - 12 contain the block label
11 - 1 contain the sector number
or 11 - 3 contain the page number
Newly defined blocks have "Sector number" 2047.

"Block Status Directory"

Consecutive half words, one per main store block. Entry m contains lock out status S of the block recorded in entry m of the block directory in digits 2 - 0. Remaining digits unused and unaltered by R203.

"Program Store Directory"

Successive half words, entry P containing for main program P:

Digits 12-2 Location relative to the start of the block directory of the area reserved for program P.
Digits 23-13 Number of blocks reserved for program P less one
Digit 1 1 if Process switch set; zero otherwise
Digit 0 Unused and unaltered by R203

"Page Directory"

Alternate half words, entry p corresponding to page p. Each entry contains:

Digit 23	:	1 if page empty, 0 if occupied
Digit 22	:	1 if page locked down, 0 otherwise
Digits 20-14	:	Program number of program owning the block (0 for supervisor)
Digits 12-2	:	Location in block directory, relative to start, of the block occupying page p.
Digit 0	:	0 if page locked out, 1 otherwise

Remaining digits unused and unaltered by R203

"Current Program Number"

One half word containing the number of the current main program in digits 8-2, digit 1 irrelevant, rest zero.

"Block Label Limit"

One half-word, normally set to *4, set otherwise by object program extracode, to request a trap if any attempt is made to define a new block by non-equilvalence in the program with a block label $\geqslant$ this.

"Block Label Trap Address"

One half word, set by above extracode giving required value of B127 for entry to trap routine.

"Non-equivalence Switch"

One half-word, normally even, set odd by R204 when a non-equivalence in object program has occurred. It is reset by R203.

Parameters Used: (1) to (37)

(2) = Address of "Block Directory"
(3) = Address of "Block Status Directory"
(4) = Address of "Program Store Directory"
(5) = Address of "Current Program Number"
(6) = Address of "Page Directory"
(7) = Two half-words temporary working space in subsidiary store
(8) = Two half-words temporary working space in subsidiary store
(9) = Entry address (5) of "Halt Main Program" (5/204)
(10) = Entry address of "Program Scan" (1/202)
(11) = Entry address of "Monitor routine - Block allocation exceeded" (1/703)
(12) = Alternative entry address to R203 for program P other than current, with P in B108 digits 8-2, rest zero.
(13) = "Reserved Block Label limit"
(19) = Address of "Block Label Limit"
(26) = Entry address of "Monitor routine - Reserved block label used" (2/703)
(27) = Address of an exit in fixed store from Extracode (e.g 513,0,0,0)
(28) = Address of "Main Program Controls" (6/201)
(30) = Entry address of "Locate Supervisor Block" (1/212)
(31) = Address of "Block Label Trap Address"
(32) = Address of "Non-equivalence switch"
(33) = Monitor mark for illegal block label. Planted in B100 as indicator for R703
(35) = Address of "Block Location Table"

Connections with Other Routines:

Entry: With B110 : E return address in extracode control, digits 23-3
B109 : Digits 22-12 b: First block label
Digits 5-3 n: One less than number of blocks required
Digits 2-0 L: Lock out reason (see Note 1)
Digit 23 H: Halt requirement (see Note 3)

Exit: To address E in extracode control with

B109 : Digits 22-12 = b + n. Other digits unaltered
B108 : Digits 20-14 : Main program number, P
Digits 12-2 : Block directory location of block b relative to the start of the area reserved for program P.
Digit 23 : Set to 1
Remaining digits zero.

B107 : Digits 2-0 : S Previous lock out status of block b + n.
Remaining digits zero.

"Block location table": Half word i holds location in block directory of block b + i relative to the start of the block directory in digits 12-2.
Remaining digits zero.

Temporary Working Space:

B105 - 110, B_t
Two words of subsidiary store, (7) and (8)

Notes:

1. Lock out reason L is specified as follows on entry:-

L = 0 : No lock out
1 : Drum transfer to cores
2 : Drum transfer to drum
3 : Read from tapes or slow peripheral
5 : Write to tapes or slow peripheral

2. Lock out status S of each block is recorded in the block status directory as follows:

S = 0 Not locked out
1 Drum transfer to cores
2 Drum transfer to drum
3 Transfer from peripheral, leave in cores
4 Transfer from peripheral, write to drum
5 Transfer to peripheral, leave in cores
6 Transfer to peripheral, write to drum
7 Transfer to peripheral, lose.

3. If H = 1 on entry (digit 23, B109) the program is halted "In Supervisor" by exit to "Halt Main Program" if any S $\neq$ 0. If all S = 0 L is transferred to all S and control is returned to E.

4. If H = 0 on entry, then either L = 0 or L = 1 or 2 and n = 0. If L = 0, control is returned to E. If L = 1, the program is halted "In Supervisor" if S = 2 or 7; if S = 4 or 6, it is changed to 3 or 5 and "Program Scan" is entered; if S = 1, 3, 5, "program scan" is entered with S unchanged. If L = 2, the program is halted "In Supervisor if S = 1 or 7; if S = 3 or 5, it is changed to 4 to 6, and "Program Scan" is entered; if S = 2, 4, 6, "program scan" is entered with S unchanged. In all cases if S = 0, L is transferred to S and control is returned to E.

5. When the program is halted "In Supervisor" the re-entry address is the "current SER entry address" in subsidiary store. B100-104 are preserved and restored. The program is resumed when the relevant block is unlocked.

6. Monitor is entered if any block label b exceeds the allowed limit on public block labels or the allowed number of blocks is exceeded, unless there is any block with S = 7, in which case the program is halted (R204) for this block.

7. If any block is not previously defined, it is entered to a vacant space in the block directory with S = 0, block label b in digits 22 - 12, all ones in digits 11-0 and digit 23 = 1.

8. If S = 0, L $\neq$ 0 the page directories are locked out and page address registers set to *4 for all blocks b to b + n in core store.

9. The lock out status S need not be altered when a page is written to the drum to create an empty page after selection by the learning program.

10. For program 0, exit is to R212 "Locate Supervisor Block" if the block is not defined in the Block Directory.

11. If "Block Label Limit" has been set by Extracode and the block required is above this limit and is being requested after a non-equivalence, then B127 is set to a Trap Address specified by the Extracode and it is arranged that an immediate switch to Main Control will occur on exit from the Supervisor.

Purpose: A supervisor extracode routine to halt a main program P. Entered at different points according to the method of and reason for, halting. The program is halted ready to resume either in the current supervisor routine or out of supervisor at the current M or E.

Register of Fixed Store: 39

Number of Instructions Obeyed:

(i) Entries at (2) (for programs to be halted for one of the standard listed reasons): 4 if program branching in use, otherwise 28+7n where n other programs are already halted for the given reason

(ii) Entry at (1) for slow peripheral in use, 6 if program branching in use; otherwise 21.

(iii) Entry at (5) for Block b not available; 5 if program branching in use; otherwise 20.

(iv) Entry at (6) for Block b not available but Resume out of Supervisor: 4 if program branching in use; otherwise 11.

(v) Entry at (17) for halt for non-listed reason: 2 if program branching in use; otherwise 9 if Resume out of Supervisor, or 17 if Resume in Supervisor.

Subsidiary Store:

"Program Status Directory"

Successive halfwords, entry P containing for program P:

Digits 23-12: Halt information:
Digit 23 = 0 if halt for block belonging to the program and digits 22-12 = Block label of block

or Digit 23 = 1, digit 22 = 0, and digit 20-14=Q, where Q is the number of the next program halted for same reason as P (If Q=0, no further program is halted)

or Digit 23 = 1, digit 22 = 1, digits 20-14=R; Program P is halted because peripheral R is in use.

Digits 8-2 = Number of program with next lowest priority to P
Digit 1 = 1 if resumption after halt is to be In Supervisor; 0 otherwise
Digit 0 = 0 if Program P is halted; 1 otherwise

"Program Switch Directory"

Successive half-words, entry P for program P

Digit 23 = 1 if program P uses program branching, 0 otherwise. Remaining digits unused and unaltered by R204.

16.7.62

"Main Program Short Dumps"

3 words per program. Half-words P, N+P, 2N+P etc. hold B100, B101 ... B104 and the SER resumption-address when program P is halted In Supervisor, where N is the length of the program list.

"Leading Halt Markers"

Successive half-words, one for each standard reason for halt:- full drum queue, full tape queue, permission to start, operator's output busy, etc. containing:

Digits 23-21 : Number of SERs halted for this reason in Top Priority SER queue
20-14 : First main program halted for this reason
13-8 : Number of SERs halted for this reason in Tape SER Queue.
7-0 : Number of SERs halted for this reason in Slow SER Queue

Parameters Used: (1) to (17)

(1), (2), (5), (16) and (17), alternative entry addresses (see below)
(3) = Maximum number of main programs (digits 8-2)
(4) = Address of Non-equivalence switch (32/203)
(7) = Address of "Main Program Short Dump"
(8) = Entry address of R202 - "Program Scan" (1/202)
(9) = Address of "Program Status Directory"
(10) = Address of "Leading Halt Markers" relative to start of Program Status Directory
(11) = Address of "Current SER Entry Address" (5/201)
(12) = Entry address to "Halt Program Branch" (1/239)
(13) = Address of "Program Change Marker" (6/202)
(15) = Address of "Program Switch Directory"

Connections with other routines

Entered at (1) For halt for slow peripheral in use:
B109 = Peripheral Number R in digit 9-2, rest zero

(2) For halt for listed reason
B109 indicates reason : 0.1 for Full Drum Queue
0.5 for Full Tape Queue
etc.
corresponding to entries 0(10). 0.4 (10) etc.

(5) For block b not available. Resume in Supervisor. After halting, return to Program Scan (R202)
B109 = b in digits 22-12, remainder irrelevant

(6) As (5), but resume "out of supervisor". B109 as for (5) B108 = E: Return address from R204 after halting (digit 0 = 0)

(17) For halt for non listed reason
B108 = Return address E in digits 23-3
Digit 0 = 0 Resume out of supervisor
1 Resume in Supervisor

In all cases B110 holds program number P in digits 8-2 on entry, with digit 23-9 zero and digits 1,0 irrelevant.

Exit: To "Program Scan" (R202) after entries (1), (2), (5).
To E after entries (6) and (17)

If program P involves program branching the routine exits to "Halt Program Branch" with B110 unaltered and B108, B109 set as follows

Entry point	B108 on exit	B109 on exit
(1)	0.1(8)	$*6 + 2^{12}$ R
(2)	0.1(8)	unaltered
(5)	0.1(8)	b (digits 22-12) rest zero
(6)	E	b (digits 22-12) rest zero
(17)	unaltered	*4

On final exit to E after entry (6), B110 is preserved, and B109 digits 22-12.

The remaining digits of B109, and B108 are altered.

Working Space: Entries (1), (5), (6), (17) : B108 - B110, Bt
Entry (2) B105 - B110, Bt

Notes:

1. In entry (6), B108 may be set to (8), the entry to Program Scan, if exit from supervisor is required.

2. When a program is halted to Resume In Supervisor, B100-104 and the "Current SER Entry Address" are preserved and subsequently restored when the program is resumed. The program is resumed at the Entry Address without recovering B0-B99 or resetting page address registers.

Purpose: A supervisor subroutine entered on conclusion of a peripheral transfer or a store organisation routine to unlock a store block and, where appropriate, set up the correct page address register and clock and lock out digits in the page directory. If a program or program branch is held up for this block, it is made free to proceed. The routine may be entered from store organisation routine when the block is not locked out to set up the page address register and page directory clock and lock out digits.

Registers of Fixed Store: 68

Number of Instructions Obeyed: varies from 9 to 47 depending on the conditions

Subsidiary Store:

(9) = "Program in store control" One half word holding the number of the program in control of store in digits 8-2. Remaining digits zero.

(10) = "Number of free blocks" One half word holding the number of free un-reserved store blocks in digits 23-3.

(13) = "Program Wait Times" One half word per program holding time at which program was made free or halted.

Parameter used: (1) to (27)

(16) = - ⌈Limit of reserved block labels⌋ = *44

Cross references:

(4)	=	(6/203)	"Page Directory"
(5)	=	(2/203)	Block Directory
(6)	=	(3/203)	Block Timers
(7)	=	(8/303)	Page timers
(8)	=	(9/204)	Program status Directory
(11)	=	(1/214)	Free program
(14)	=	(1/312)	Change page address registers
(15)	=	(15/204)	Program switch directory
(17)	=	(6/229)	Current time
(18)	=	(6/202)	Program change marker
(19)	=	(2/312)	Contents of page address registers
(20)	=	(7/202)	Program Branch Indicator
(21)	=	(1/234)	Free program branch

Note: R205 refers to instruction 7(14), which is the last one of R312, namely 121, 126, 110, 0 which is used for exit to R.

Connections with other Routines

Entry at (1) After tape and on-line peripheral transfers
B109 = Page number p in digits 11-3.
Remaining digits zero.
B110 = Return address, R, digits 23-3
Return to R with B109 = p, digit 0 = 1 unless page p holds a supervisor block which is lost on completion of transfer.
B 110 is unaltered.

16.7.62

Entry at (2) From drum routines when block is in core store
B109, 110 as for entry (1)
Return to R with B109, 110 unaltered.

Entry at (3) From drum routines when block is on the drum or is unallocated
B108 = Digits 12-2 Position of block in block directory relative to the start of this directory
Digits 20-14 Program owning the block
Remaining digits irrelevant
B110 = Return address R, digits 23-3
Return to R with B109 = 0, B110 unaltered

Entry at (27) To free program or branches if halted for block b
B106 = Program number digits 8-2 Digits 23-9 zero
B108 = Block number b digits 22-12
Digit 23 = 1
Remaining digits irrelevant
B109 = digit 0 = 0. Rest irrelevant
B110 = Return address R, digit 23-3
Return to R with B107 - 110 unaltered

Entry at (12) From "Free program" (R 214) to free branch for listed reason with
B108 = 0
B109 = Reason digits 4-2
B110 = Return address R, digits 23-3

Exit a) To "change page address register", R312, with
B109 = Page number digits 11-3. Digit 0 = 0 or 1, remainder zero
B110 = Return address R digits 23-3
B108 = *4 or digit 23 = 0, 22-12 = block label
or digit 23 = 1, 22-12 = block label

b) To "Free program" R214 with
B109 = 1.0
B110 = Return address R

c) To word 7 of R312 " 110 → 126" with
B110 = Return address R
B109 = Page number of zero if entry at (2)

Subroutine:

"Free Branch" : Entered with
B108 = block directory entry of block b (after entry at 1,2,3 or 27), 0 after entry (12)
B109 = Reason, digits 4 - 2, after entry (12)

Exit to 2(12) of R205 with B105, 107-110 unaltered and bt' = 0

Temporary working space: B105 - 109, Bt

B109 unaltered after entry (2)
B109 digit 0 forced to 1, remaining digits unaltered, after entry (1)

Notes:

1. On entries 1,2 the page directory must be correctly set up in digits 20-14, 12-2. The clock and lock out digits are set by R205, the remaining digits are unaltered.

2. On drum entries 2, if the lock out status S is $\geqslant 3$, digits 13, 0 of the page directory are set to zero and the P.A.R. is set to *4. All other directories remain unchanged. Entry 3 must not be used if $S \geqslant 3$.

3. On entry 1, if S is even, the "write to drum" digit is set in the relevant page timer.

4. With the exception of case (2), S is set to zero and the program is made free if halted for this block. If the program is currently in control of store and is involved in program branching, the relevant branches are made free.

5. If S = 7 on entry (1), the page and block directories are vacated, the PAR is set to *4, one is added to the number of free blocks, and the leading routine halted for "No free store blocks" is made free via R214

6. If the block is in core store, the following page directory entry and PAR are set up

		Page Directory		PAR	
Program	Block label	Clock digit	Lock out digit	L.0 bit	22-12
Current	$<$*3400	0	1	0	b
"	$\geqslant$*3400	0	0	1	0
Supervisor	$<$*3400	0	1	1	0
	$\geqslant$*3400	0	0	1	b
	$<$*3400	As clock digit	1	1	0
	$\geqslant$*3400	in directory	0	1	0

7. The routine may be used in conjunction with R329 "Removed lock down" by setting B110 = (2/329) on entry to R205 and setting a link in B104. The page number is preserved in B109 on exit.

8. On entry (1) only, the current time is recorded in the Program Wait Timers if the program is made free.

9. When the store block is unlocked on exit, B107 holds the program number (to which the block belongs) in digit 8-2, remaining digits zero.

10. The routine may be entered at 1(3)from drum routines irrespective of the location of the block, provided
    B108, 110 are set up as for entry (3)
    B109 holds the page number if the block is in core store, and is even when on the drum
    If in core store, the page directory has digits 13, 0 = 0
    If on the drum, $S < 3$.

11. On entry (1), and on entries (2) & (3) if $S < 3$, B106 digits 8-2 on exit contains the program number associated with the block concerned.

Purpose: A closed subroutine entered from an SER to enter a new additional SER to one of the SER queues. An alternative entry used by the drum supervisor to restart an SER at the conclusion of a drum transfer.

Register of Fixed Store: 14

Number of Instructions Obeyed: 9 on normal entry; maximum of 14 drum entry 6 instructions are obeyed with interrupts inhibited.

Parameters used: (1) to (4)

1,2 = Entry addresses (see below)

(3) = "SER Queue Address". 3 alternate half words in fixed store containing the addresses relative to the start of subsidiary store of "Slow SER Queue", "Tape SER Queue" and "Top Priority SER queue" in order.
Digits 1,0 hold 11 (slow) 01 (tape) 10 (Top priority)

(4) = Parameter (9) of R214 (Free program)

Connections with Other Routines

Entered:

At (2) from an SER or from R214 with
B110 = Return address in extracode
B109 = Starting address of new SER in digits 23-3.
Digits 2-0 zero on entry from SER
B108 = Entry information for new SER, found in B100 when the SER is entered.
B107 = Queue to which entry is to be made:
2.0 Top priority queue
1.0 Tape queue
0.0 Slow queue

At (1) from drum supervisor with
B108 = Information as preserved in drum queue
B109 = Starting address of new SER in digits 23-3
Queue number in digits 2-1 : 00 Main Program queue
01 Tape queue
10 Top queue
11 Slow queue

B110 = Return Address

Enter in this case with b126 odd (i.e digit 0 = 1)

Exit: In all cases to address specified in B110 on entry.
B108 - 110 are unaltered on exit.

Working space: B106, B107

Notes:

1. Interrupts are permitted on exit (register 3*6 contains zero).

2. Entry (2) constitutes entry F of the co-ordinator.

3. If the drum supervisor enters with digits 2, 1 of B107 zero, then the main program number P is held in B108, and program P is made free to proceed, by entry to R214. The entry address, B109 digit 23-3 is irrelevant in this case; the required re-entry address has previously been planted by R318.

4. In all other cases, an entry is made to the SER queue in the next vacant position and the count of the number of entries is stepped on.

Purpose: A supervisor extracode routine entered from "Program Scan" to select the highest priority main program which is free to proceed. If the SER queues are not empty, immediate exit is made to "Update SER queues".

Registers of Fixed Store: 10

Number of Instructions obeyed: Maximum of 10 + 3n where the n highest priority main program are halted.

Subsidiary Store:

"Leading program number"

One half word containing the address in the program list of the top priority main program in digits 8-2. Remaining digits irrelevant.

Parameters Used: (1) to (9)

(2) = Address of "Current program number" (5/203)
(3) = Address of "Program change marker" (6/202)
(4) = Address of "Program status directory" (9/204)
(5) = Address of "Supervisor program change" routine (1/222)
(6) = Parameter (15) of "Program Scan" (15/202)
(7) = Entry address of "Update SER queues" (1/208)
(8) = Entry point (2) of "Supervisor program change" routines (2/222)
(9) = Address of "Leading program number"

Connections with other routines:

Entered: From "Program Scan" with interrupts inhibited

Exit: To 'Update SER Queues' with interrupts permitted if B109≠0 on entry
To "Program Scan" with B126 odd if the first free program is the current program. Program change marker is set to zero before exit.
To "Supervisor program change" otherwise with
B110 = First free program number in digits 8-2
B100 = Program Status directory entry of first free program.
The Program change marker is unaltered.

Working space: B100, 110

Purpose: A supervisor extracode routine entered from "Select Main Program" when SER queues not empty to select the next SER awaiting execution in an SER queue and to update the relevant queue.

Registers of Fixed Store: 25

Number of Instruction Obeyed:

Maximum 18+7N where there are N entries in the SER queue from which an entry is extracted. A maximum of 5 orders are obeyed with interrupts inhibited.

Parameters Used: (1) to (6)

(2) = Relative address of fixed store table "SER queue address" (3/206)
(3) = Entry address to "Resume SER" (7/211)

Connections with other routines:

Entry: From "Select main program" (R207)

Exit: To word 2 of "Resume SER" with
B110: Entry address of next SER as stored in SER queue
B109: Digits 1,0 = 10 (from top priority SER queue)
01 (from tape SER queue)
11 (from slow SER queue)
Digit 23 = 0
B100: Entry information of next SER as stored in SER queue.

Working space: B100, B106 - 110, B_t

Purpose: To enter an SER after extraction from an SER queue or after setting of "Program Scan Exit". If necessary B100-104 are recovered before entry.

Registers of Fixed Store: 14

Number of Instructions Obeyed: 4 minimum; 21 maximum

Subsidiary Store:

"Current SER Dump Address" : One half word holding the address of dump for B100-104

Parameters: (1) to (8)

(2) = Address of Current SER entry address (5/201)
(3) = Address of Current SER dump address
(4) = Address of "Current SER base" (7/201)
(7) = Alternative entry (see below)

Connections with other routines

Entry:

(a) From "Program Scan" (R202) with
B110 = Entry address of new SER
B109 = Digits 1, 0 of B110

Exit is to the new SER with base planted from B109, entry address = B110 with digits 2-0 zero, B109, 110 only are altered.

(b) At (7) from "Update SER Queues (208) with

B110 = Entry address; digit 0 = 0 (no recovery of B lines)
1 (recover B lines 100-104
B109 = SER Queue Number: 10 top priority queue
01 tape queue
11 slow queue
B100 = Entry information or dump address

Exit to the new SER with base, entry address and where appropriate the dump address planted and where required B100-104 recovered.

Working Space: B108-110 Bt

Notes:

1. The current SER base, entry address and dump address are set in subsidiary store.

2. 'The Dump Area' (in the case when B100-104 are to be recovered) need not be in cores on entry to this routine. If it is not, a 'non-equivalence in supervisor' will occur when reference is made to it and appropriate action will be initiated by the non-equivalence interrupt routine in the usual way.

R212 Initial location of supervisor block

Purpose: A supervisor routine entered from "Store location and Lock out" when a supervisor block required in the one-level store is not found. It halts the SER concerned, and, if this is the first such halts, initiates a new SER to locate this block on a reserved sector or on tape and calls it to the one-level store.

Registers of fixed store: 8

Instructions obeyed: 4 to 8

Parameters used: (1) to (5)

Cross references:

(2)	=	(2/213)	Leading halt marker
(3)	=	(1/299)	Entry to main store SER "Find Supervisor Block"
(4)	=	(2/206)	Entry to Enter SER to Queue
(5)	=	(1/213)	Halt SER

Connections with other routines

Entry at (1) from R203 with
B109 = block label digits 22-12.
Remaining digits irrelevant

Exit to Halt SER with
B109 = 1.4 (reason)
Exit from there to program scan

Subroutine

"Enter SER to Queue" entered at (2/206) with
B107 = 0 (slow queue)
B108 = block label digits 22-12. Rest zero
B109 = entry address of SER "find supervisor block"
B110 = return address
Return to return address. B lines irrelevant

Temporary working space: B107 = 109

Notes:

1. The reason for halt is 1.4

2. The main store SER is only initiated once. On conclusion of the leading request, this will free all such halted SER, and may then be called again.

3. Program branch routines will arrange to halt the entire program if one branch is halted for this reason.

R213: Halt SER

Purpose: To halt the current SER (for full drum queue, tape queue etc.) and exit to program scan. A re-entry address and B100 are preserved, and, if necessary, B101-104 also. An alternative entry is available from the drum routine R318, "Call to cores", to preserve B100-104 before entering the request to the drum queue.

Registers of Fixed Store: 37

Number of Instructions Obeyed: 5 if current SER is based on Main Program (with exit to R204, "Halt Main Program")

For SER not based on Main Program, 20 if B100 only to be preserved, 34 if B100-104 to be preserved.

Subsidiary Store:

"Halt postions in SER queues"

Three alternate half words, recording the next location at the base of the slow, tape, and top priority SER queues respectively to which halted SER's are to be placed. Originally set to the address of the last half word in each queue. These addresses are relative to the start of the subsidiary store.

Parameters Used: (1) to (15)

(1) Normal entry address
(2) Address of "Leading Halt Marker" relative to start of subsidiary store (see R204)
(3) Address of "Halt Positions in SER Queues"
(4) Address of "Current SER Entry Address" (5/201)
(5) Address of "Current SER Base" (7/201)
(6) Internal cross-reference
(7) Address of "Current Program Number" (5/203)
(8) Entry address of "Program Scan" routine (1/202)
(9) Address of fixed store table, "Leading Halt Units" : Three alternate half-words containing +0.1,+32.0 and *1
(10) Internal cross-reference
(11) Alternative entry from drum routine (see below)
(12) Address of "Current SER Dump Address" (3/211)
(13) Internal cross-reference
(14) Parameter (2) of R204, "Halt Main Program"
(15) Internal cross-reference

Connections with other routines

Normal entry; at (1) with B109 = Reason for halt in half-word position
0.0 Full drum queue
0.4 Full tape queue
etc.

Exits after entry at (1):-

(i) To Halt Main Program if SER is based on main program with B109 = Reason +0.1
B110 = Current Program Number.

(ii) If SER is not based on main program, to Program Scan after recording B100-104 where appropriate and recording the re-entry address and dump location or B100 at the base of the relevant SER queue.

Alternative entry at (11) from drum transfer routine with B110 = Return address E.

B-lines used: Normal entry B105-110
Drum entry B105, B107 only

Notes:

1. If the SER Re-entry Address is odd, B100-104 are preserved where specified by the SER Dump Address, and this address is preserved in the halted part of the SER queue. If the SER Re-entry Address is even, B100 itself is preserved in the 'halted-part' of the SER queue. These alternatives are distinguished by inverting digit 21 of the entry address as it is put in the queue in the first case. In both cases the 'Reason for Halt' is preserved in digits 2-0 of the Re-entry Address in the queue.

2. (a) For the normal entry at (1), if B100-104 are to be preserved the 'Dump Area' must be in cores on entry.

   (b) For the entry at (11) from the drum routine, the Dump Area may not be in cores. If it is not then this entry will have arisen from an interrupt in R211, and the Current SER Base will have been set negatively, and no attempt will be made in R213 to preserve B100-104.

R214: Free Program

Purpose: A supervisor extracode subroutine to make free in the relevant queue or the main program list the leading SER or main program if any halted for full drum queue, tape queue etc. The SER lists are checked in order of priority and then the main program list. Within each queue, priority is given to the routine first halted for the specified reason. The routine is freed for eventual selection by R202, "Program Scan", control at the end of R214 or its continuation elsewhere being returned to the Return Address specified by the routine calling for R214.

Registers of Fixed Store: 43

Number of Instructions Obeyed:

2 if routines halted for specified reason
17 if a main program only is halted for specified reason
A maximum of 37+5n if an SER is halted, where there are n halted entries in the halted part of the relevant SER queue.
7 instructions are obeyed with interrupts inhibited.

Parameters used: (1) to (13)

(2) = Address of "Program Status Directory" (9/204)
(3) = Address of "Halt positions in SER queues" (3/213)
(4) = Address of fixed store table "SER queue addresses" (3/206)
(5) = Address of fixed store table "Bottom of SER queue addresses": Three alternate half words in fixed store containing the addresses, relative to the start of the subsidiary store, of the last half word in the slow, tape and top priority SER queues respectively.
(6) = Address of fixed store table "Leading halt units" (9/213)
(7) = Address of "Program Change Marker" (6/202)
(8) = Alternative entry to R214 (used by R217)
(9) = Alternative entry to R214 (used by R206)
(11) = Entry address of "Free program branch" (1/234)
(12) = Address of "Leading halt markers" (2/213)
(13) = Entry address (2) of "Enter SER to Queue" (2/206)

Connections with other routines:

Entry: At (1) with B110 = Return address E
B109 = "Reason for halt" in half word position (as for R213)
- 0 for full drum queue
0.4 for full tape queue etc.

Exit: 1) To "Free Program Branch" (R234) if no routine is halted for the given reason, to check if any branches of the current program are halted. An immediate exit to E occurs if no branching is in use.
2) Direct to E with B107, 108 altered, if a main program but no SER is halted for the given reason.
3) To "Enter SER to Queue" (R206) if any SERs are halted for the given reason, with
B107 = Queue number - 2 for Drum, 1 for Tape, 0 for Slow
B108 = Entry "information" for SER freed
B109 = Entry address for SER freed
B110 = E

Entry: At (9) from R206 to "Enter SER to Queue" if based on main program
B108 = Program no. in digits 8-2, rest 0
B110 = Return address E
Exit to E with B107 altered

Entry: At (8) from R217 to move up entries in halted part of SER queue after one has been removed by R217 and to place this in active part of queue
B107 = 1.0 (for tape queue)
B106 = Address of 'Entry Address' of entry removed from queue, relative to start of subsidiary store.
B109 = 'Entry Address' of entry as removed from queue
B110 = Return Address E
Exit to E via R206 with B105-7 and Bt altered.

Notes:

1. The time at which a main program is made free is not recorded.

2. If the halted SER has digit 21 of the re-entry address inverted (see R213), the original is recovered and digit 0 is set to 1. Otherwise digits 2-0 are cleared.

R215 Set and reset full recovery switch

Purpose: An SER subroutine to set and reset a bit in a program status directory causing the program to be resumed in Supervisor and with B1-99 and Subsidiary store working space recovered. If the bit is not set (normal state) resumption in Supervisor is with B100-104 only recovered.

Registers of fixed store: 9

Instructions obeyed: 3 to 5 depending on entry

Parameters used: (1) to (6)

Cross references: (5) = (9/204) Program status directory
(6) = (5/203) Current program number

Connections with other routines:

Entry at (1) to set switch for current program
(2) to set switch for program P. B110 = P (digits 8-2) Digits 23-8 zero
at (3) to reset switch for current program
(4) to reset switch for program P, B110 = P (digits 8-2) Digits 23-8 zero

In all cases B109 = Return address

Exit in all cases to return address with
B110 = Program number (8-2)
B108 altered

Temporary working space: B108

Notes:

1. The switch is digit 9 of the program status directory. Digit 9 = 1 forces full recovery.

2. The digit is "forced" to 0 or 1 irrespective of its original setting.

3. Routines which set or reset this switch should specify such in order that other routines entering them may know the status of the switch.

R216 Establish tape exit

Purpose: To preserve a link address for return on completion of a tape transfer, or on completion of other activities. The address is preserved in the tape SER queue, halted region, and is recovered from there by use of R217

Register of fixed store: 6

Instructions obeyed: 6

Parameters used: (1), (2)

Cross reference: (2) = (3/213) Current halt positions in SER queues

Connections with other routines:

Entered in supervisor at (1) with
B100 = deck number, digits 7-3, Rest zero
B109 = link address to be preserved (digits 23-3)
digits 2, 1, 0 all 1
B110 = return address from R216

Exit to return address with B108 altered

Temporary working space: B108

Notes:

1. An entry is placed in the next available position of the halted region at the base of the Tape SER queue comprising

Information : B100 on entry (deck number)
Re-entry address : B109 on entry

2. This entry appears as an SER halted for reason 7 (see R213) but the leading halt marker is not affected and hence reason 7 can be used as a halt reason provided SER's based on the tape queue are not so halted.

3. One space per deck is reserved in the tape SER queue for this purpose. Care must be taken that only one such entry is made per deck at any one time.

R217: Tape exit to supervisor control

Purpose: An SER subroutine to recover a link address, recorded via R216, and to enter the SER starting at this address in the active region of the tape SER queue.

Registers of fixed store: 10

Instructions obeyed: Maximum of 6+4N where the link address is recorded N places from the end of the tape SER queue, plus entry to R214

Parameters used: (1) to (3)

Cross references:

(2)	=	(5/214)	Base of SER queues
(3)	=	(8/214)	Alternative entry to "Free Program"

Connections with other routines:

Entered at (1) with B109 = deck number (digits 7-3) rest zero
B110 = return address from R217

Exit to (8/214) to move entry to active part of tape queue with
B109 = Entry address of new SER, digits 23-3. Digits 2-0 zero
B106 = Location of entry in halted part of queue (such that address is stored in (B106)*7)
B107 = 1.0 (tape queue)

Exit thence to return address with B105-109 altered

Temporary working space: B105-109, B_t

Notes:

1. A search is made from the base of the tape SER for an SER halted for "reason 7" and with information = deck number. It is assumed without checking that such an entry has been made previously by R216.

2. When the SER is finally entered, B100 will contain the deck no. (digits 7-3) and the SER will be based on the tape queue.

3. In order to force this exit following a tape transfer, the entry in the tape queue recording the type of order (B104 on entry at (16/421) should have digit 22 set to 1.

R218: Step block directory reference

Purpose: An SER entered from tape routines and other SER which process more than one block of an object program. The routine finds the block directory location of a block with label one more or one less than a given block.

Registers of fixed store: 10

Instructions obeyed: 9 or 10 plus entry to R203

Parameters used: (1) to (4)

Cross references: (2) = (4/203) Program Store Directory
(3) = (2/203) Block directory
(4) = (12/203) Alternative entry to R203

Connections with other routines:

Entry at (1) to step label back by 1 (B126 even)
at 1.1.(1) to step label forward by 1 (note B126 odd)
with B107 = Program no. P(20-14)
Directory location of block B relative to start of area for this program (12-2)
Remaining digits irrelevant
B110 = Return address R

Exit: To (12/203) with
B110 unaltered
B109 = B ± 1 (22-12). Rest zero
B108 = P(8-2)

Exit back to R with B109 = B ± 1 Rest zero
B108 = P(20-14); Directory location of B ± 1 relative to area of P(12-2)
Block location table, entry 0 = Directory location of B ± 1 relative to start of BD
B107 = Lock out status of block (2-0). Rest zero.

Temporary working space: Together with R203, B105-110, B_t

R220: Reserve and free operators output

Purpose: A supervisor subroutine to reserve an output channel common to many SER's prior to transmission of a message, and to free the channel for other users after transmission.

Registers of fixed store: 9

Subsidiary store:

(3) = "Table of Operators Output Channels"
One half word per common output channel in successive half words, entry n for logical channel n.
Digit 0 = 0 if free. 1 if reserved for use.
Digits 23-1 = Address of working area for this channel less *7 (see R240)

Parameters used: (1) to (5)

Cross references: (4) = (1/213) Halt SER
(5) =2(1/230) Free program

Connections with other routines:

Entry at (1) to reserve logical channel n with
B101 = n; digits23-2. Rest zero.
B110 = Return address, digits 23-3

Exit to return address if channel free with reserved mark set in subsidiary store and
B100 = Address of working area of channel less *7
Digit 0 = 1
B108, 109 altered

Exit to "Halt SER" if channel not available with
B109 = 2.0 (reason for halt)
Exit thence to program scan.

Entry at (2) to free logical channel n with
B101, B110 as for entry (1)

Exit to "Free Program" with
B109 = 2.0
B100 = Address of working area of channel less *7. Digit 0 = 0
B110 unaltered

Return from there to return address with B105 - 109 altered

Temporary working space:

B108, 109. On entry (2), B105 - 107 altered by free program.

R220: continued

Notes:

1. On the Manchester University Atlas, the chief operators output is 0, the tape operators is 1 (B101 = 0.4 on entry).

2. On entry (1), if the channel is already reserved, resumption will be at the re-entry address when the channel is freed. This address is not altered.

3. Reservation will always be followed by transmission to a buffer via. R240.
B100 is set up on exit from R220 as required on subsequent entries to R240.

4. Space is reserved in subsidiary store for the most common output channels in address (3). Other channels may also be used by preserving their working area address less *7 in a register in main store, R, and entering R220 with B101 = R -(3). R must be known to be in core store on entry to R220.

R221 Find tape deck number

Purpose: A supervisor extracode subroutine to find the absolute deck number of a tape deck with program label B, to halt if this deck is not available, and to monitor if the deck is not defined.

Registers of fixed store: 19

Instructions obeyed: 11 minimum
44+5D maximum where D is the number of tape decks

Subsidiary store:

(5) = "Deck allocation directory"
Alternate half words, entry d corresponding to deck d.
Digits 20 - 14 Program label B
Digit 13 1 if Orion tape, 0 for Atlas tape
Digit 8 - 2 Program number P
Digit 0 1 if not available, 0 if available
Remaining digits unused and unaltered by this routine

Parameters used: (1) to (9)

(3) = "Number of tape decks" digits 7 - 3. Remainder zero
(6) = Monitor counter "Tape not defined" digits 8-2

Cross references

(2) = (5/203) Current program number
(4) = (1/215) Set and Reset Full Recover switch
(7) = (14/700) Monitor interrupt and SER
(8) = (9/204) Program status directory
(9) = (1/204) Halt program

Connections with other routines

Entry at (1) to find deck B of current program
B100 = B digits 8-2
digit 1 = 1 if Orion tape, 0 otherwise.
Remainder zero.
B109 = Return address digits 23-3

Entry at 1(1) to find deck B of program P
B100, 109 as above
B110 = Program number P digits 8-2
Remainder zero

Exit (a) to R with B100 = Absolute deck number, digits 7-3.
Remainder zero

(b) to Halt Program at (1/204) if deck not available with
B109 = Deck number digits 7-3 digit 1 = 1
B110 = Program number

Exit to program scan with B100 altered.

R221 continued

(c) To Monitor at (14/700) with
B100 = Fault number "Deck not defined" digits 8-2
Exit to program scan

Subroutine "Set and reset full recover switch" entered at 5(1/215)
to reset switch for current program with
B109 = Return address R.
Exit to R with B108 altered

Temporary working space: B106 - 110, B100

On exit (a), B100, 106-108 only.
B110 = Program number of exit.

Notes:

1. If the deck is not available, program P is halted with the full recover switch set. Resumption is at the re-entry address with PAR and B1-99 recovered, also B101-104. B100 is altered on resumption

2. On exit to R, the full recover switch is reset

Purpose: A supervisor routine to re-enter an SER based on a main program, to enter full program changing when required, and to enter a test routine when no program is free to proceed.

Registers of Fixed Store: 26

Number of Instructions Obeyed: From 6 to 18

Subsidiary Store

(12) ="Local time of test program"

One half word used for accumulating count of time spent in test programs.

Parameters Used: (1) to (18)

(3)	=	(1/207)	Select main program
(4)	=	(1/223)	Check full program change
(5)	=	(7/204)	"Main Program Short Dump"
(6)	=		Number of main programs in digits 8-2, rest 0
(7)	=	(5/201)	"Current re-entry address"
(8)	=	(7/201)	"Current SER base"
(9)	=	(5/203)	"Current program number"
(10)	=	(9/204)	"Program Status Directory"
(11)	=		"Duration of test program
(13)	=		Entry to test program
(14)	=	(1/202)	Program Scan
(15)	=	(9/205)	Current store program number
(17)	=	(17/202)	

Connections with other Routines

Entry at (1) from R207 with
B100 = Program Status Directory of new program
B110 = Program number of new program, digits 8-2
Remaining digits zero.

Entry at (2) to resume SER with
B110 = Program number of new program, digits 8-2
Remaining digits zero
B109 = 0.2

Exit: After Entry at (1)

a) To (17) of R202 with B100,110 as on Entry (1) but current program number reset to current store program number. This exit is if the current program has been previously re-entered In Supervisor, and is to re-check for change of program.

b) To test program if B110 = 0 on entry (no program free to go) with B110 = Entry address of "Program Scan"
B109 = Duration of test program. This quantity is also added to the accumulated "local time of test program".

(2) To "Check full program change" (R223) if the new program requires full recovery of B lines and working space with
B100, 110 unaltered
B101 = Current program number digits 8-2
Remaining digits zero.

Exit after entries (1) and (2) to

(d) New SER with B100-104 recovered, re-entry address reset current SER base zero, and current program number set with program number (digit 8-2)
digit 1 = 1
remaining digits zero

Temporary Working Space: B101, 108, 109

Notes:

1. Entered at (1) from R207 when a program change may be required.

2. Entered at (2) to resume an SER based on a main program without altering subsidiary store working space or B1-99.

3. The program status directory of program 0 has d9 = d1 = d0 = 1, d8-2 = 0. When no object programs are present, the leading program number is zero, the current program number 1 and the test program is run continuously.

R223 Interrogate full program change.

Purpose: A supervisor extracode routine to investigate whether to proceed with a full program change. If the current program is halted for a drum transfer and insufficient time remains to complete a program change, test routines are entered. If two programs of equal priority are involved, a change is only made if the current program is halted. If the dump block for the new program is not in core store, a drum transfer may be ordered and program scan re-entered. If the dump block is in core store, full program changing is entered.

Registers of fixed store: 43

Instruction obeyed: 22 minimum; 49 maximum

Subsidiary store:

(15) = "Criterion for program change"
One half word holding minimum drum angle at which program changing is to be carried out.
Digit 5-3 reading from 0 to 6

(16) = "Location of dump blocks"
One full word holding
digits 36 - 26 Location in block directory relative to the start of the directory of the first dump block
digit 25 - 24 Both 1
digits 12 - 2 Location of the last dump block
digit 1, 0 Both 1
Remaining digits zero.

Parameters used: (1) to (18)

(17) = Drum V store = *6002

Cross references:

(2)	=	(15/204)	Program switch directory
(3)	=	(20/315)	Drum queue
(4)	=	(2/315)	Address of 1st entry in drum queue
(5)	=	(36/314)	Subroutine to find block location
(6)	=	(9/204)	Program status directory
(7)	=	(2/203)	Block directory
(8)	=	(16/222)	Alternative entry to Supervisor Program Change
(9)	=	(3/203)	Block Status directory
(10)	=	(35/203)	Block location table
(11)	=	3(1/207)	Alternative entry to Select Main Program
(13)	=	3(1/207)	Alternative entry to Select Main Program
(14)	=	(18/318)	Alternative entry to Call to Cores

Connections with other routines

Entered at (1) from Supervisor program change with
B100 = Status directory of new program
B101 = Current program number digits 8-2
B110 = New program number digits 8-2

R223 continued

Exit (a) to 1(16/222) to enter "idle" program

(b) to "Change working registers" at 1/226 with
B104 = dump block label of new program, digits 22-12
remaining digits as in block directory
B109 = program switch directory of new program
B101 = current program number digits 8-2
Entry 0.0 of block location table = new program number

(d) to Select main program to resume program selection with
B100 = Status directory of new program. B126 odd

(c) to "Call to cores" at -2(18/318) to enter a priority drum transfer with

B101 = 0
B102 = digits 12-2 block directory location dump block digits 1,0 both 1 (priority, operand)
B105 = 2.0 (Priority 2)
This sets B103 and enters the drum queue routine

Subroutine: "Block location", entering drum transfer routine at(36/314)with
B110 = Return address
B102 = Program no. P digits 20-14
Location in block directory relative to the start of area of program P digits 12-2
Return with B102 unaltered
B107 = Program no. digits 8-2 remainder zero
B108 = Location in block directory relative to the start of the block directory, digit 12-2

Temporary working space: B102-104, B107-110, Bt, block location table.

Notes:

1. "Idle" is entered if all the following conditions apply
   a) Current program halted for a block B
   b) This block is at the head of the drum queue and is involved in a non-equivalence or "read to cores" drum transfer (not a supervisor drum transfer)
   c) The difference between the required φ and present θ is less than a criterion set in subsidiary store.
2. Program selection is resumed if the current program is free to go and is recorded as of equal priority to the new program (digit 10 of program status directory = 1 for both program).
3. Program selection is resumed if the dump block of the new program is not in cores and any dump block is already being called to core store.
4. A priority drum transfer (priority 2) is entered to the drum queue to call the dump block to core store if it is on the drum and no other dump block is in the drum queue. Exit is then made to "Program Scan".
5. The dump blocks occupy consecutive positions in the supervisor area of the block directory and have reserved block labels; their block timers are reset normally by the drum routines.

R224: Find next branch

Purpose: An SER in locked down main store entered from R202 when resuming an object program which is branching. Finds the top priority free branch, switches registers as necessary, and arranges that this branch will be resumed by returning to R202 (program scan).

Registers of Main Store: 105

Instructions Obeyed: 2 with interrupts inhibited if branch change marker not set.

Parameters Used: (1) to (28), (50), (52) to (54)

0.4 (28) = half word holding lowest B register to be changed, digits 8-2
Normally set at 45.4; reset by R237 for full switch of B registers.

(18) = Block label of main store block containing Branch table and routines.

(19) = No. of words subsidiary store to be preserved, digits 23-3.

(20) = Monitor marker "Branches locked"

(10) = One half word holding "Current branch controlling store" (position in branch table)

Cross References:

(11)	=	(7/201)	SER base
(12)	=	(5/201)	SER re-entry address
(13)	=	(9/205)	Current program number
(14)	=	(9/204)	Program status directory
(15)	=	(4/203)	Program store directory
(16)	=	(35/203)	Block location table
(17)	=	(3/203)	Block status directory
(21)	=	(1/202)	Program scan
(22)	=	2(17/202)	Re-entry to program scan
(23)	=	(2/239)	Kill branching
(24)	=	(2/226)	Switch registers
(25)	=	(1/203)	Locate store block
(26)	=	(1/225)	Resume branching
(27)	=	(14/700)	Monitor
(50)	=	(50/238)	Branch table
(52)	=	(52/238)	Top priority branch
(53)	=	(53/238)	Branch change marker
(54)	=	(54/238)	Current branch

R224: continued

Connections with other routines

Entry at (1) under extracode control from Program Scan (R202) with interrupts inhibited.

Exit: a) To program scan at (22/224) with interrupts inhibited if branch change marker not set, B100 altered

b) To program scan at (1/202), branch change marker reset to zero, with highest priority free branch set to run unless this branch requires resumption In Supervisor.

c) To entry address In Supervisor of highest priority free branch with B100-104 recovered if branch was set to resume In Supervisor.

d) To "Kill branching": Entered at (2/239), B126 even, B107 = 0.1(14/700) : Entry to monitor
Entered if all branches halted for other branches.

e) To "Halt branching": Entered at (2/239), B126 odd,
B109 = Halt reason for main program
(recalculated from branch halt reason to correspond to B109 on entry to 239 from R204).
B108 = Entry to Program scan
Exit thence to Halt Program

This exit taken if all branches halted - program halted for same reason as top priority branch.

Subroutines:

a) Store location and lockout:

Entered at (1/203) to lockout dump page
B109 = block label (p.22-12) p0=1
B110 = return address
Exit with directory position in word 0 of block location table. B105-110 altered.

b) Switch register

Entered at (2/226) to switch B lines and sub. store with
B107 = return address
B103 = No. of words of subsidiary store -1, full word position
B104 = Address of dump for current
B105 = Address of dump for new
B106 = 45.4 to 0. Lowest B line to switch
Exit with registers switched and B100-102 unaltered.

Temporary working space: B100-110 B_t

R224: continued

Notes:

1. Exit is immediate if the branch change marker is zero. The marker is reset to zero before entry to a new branch out of supervisor, but not if resumed In Supervisor.

2. If the process switch is set on entry, no change is made from the current branch; if the current branch is halted, then the full program is halted.

3. The program is monitored if all branches are halted for others.

4. If all branches are halted, the program is halted for the same reason as the highest priority branch not halted for another branch.

5. If a branch is resumed in supervisor but without recovery of working extracode space, the branch owning extracode is recorded in half word (10/224).

6. When a full branch change occurs, dump positions are arranged (by R237) to fill the block used for branch table and programs and to continue to other sequentially numbered blocks. In each block one empty space is preserved for the current branch, so that switches use only one block in core store; the address of the empty space is recorded in the last half word of each block and is updated by R224. The blocks are called to core store by non-equivalence In Supervisor with the process switch set, and the P.A.R. and Page Directory are subsequently locked out.

7. When a new branch is started, the current registers are dumped to the area for the new branch and recovered from there.

R225: Resume program branching

Purpose: An SER in main store occupying the same block as the branch table, entered when a program is resumed following a program change, and when branching is establishing. The block is called to core store by non-equivalence in supervisor. The routine R225 locks the block down, resets program directories and arranges a search for the top priority free branch. An alternative entry as a subroutine omits searching for the branch (used by R224).

Registers of main store: 43

Instructions Obeyed: Maximum 35 + 9N where N branches are active plus entry to R203 to locate one block, no. lock out.

Parameters Used: (1) to (18), (50), (52), (55)

(16) = *6001 Page address registers

Cross references

(3)	=	(7/202)	Current program branch indicator
(4)	=	(6/203)	Page directory
(5)	=	(8/303)	Page timers
(7)	=	(1/203)	Entry to locate store block
(8)	=	(1/202)	Program scan
(9)	=	(15/204)	Program switch directory
(10)	=	(9/205)	Current program number
(11)	=	(4/203)	Program store directory
(12)	=	(9/204)	Program status directory
(13)	=	(35/203)	Block location table
(14)	=	(2/203)	Block directory
(15)	=	(9/224)	Re-entry to find next branches
(17)	=	(2/312)	Contents of Page Address Registers
(50)	=	(50/238)	Branch table
(52)	=	(52/238)	Top priority branch
(55)	=	(55/239)	Store of directories

Connections with other routines:

Entered at (1) from program change sequence on resume program B126 even. Process and full recover switches set to 1 on entry.

Exit: To program scan after resetting switches and arranging to resume branches (see note).

Alternative entry at (1) with B126 odd from Find next branch, to reset switches.

Exit: To 9/224 with switches reset, branch table unaltered B104 = 110 altered (B105-110 by R203).

Subroutine

"Locate store block" entered at (1/203) to locate block of current program
B110 = Return address
B109 = Block label 22-12 Rest zero
Exit to return address. B107 = Lock out status
Word 0 of block location table = directory entry
B105-110 altered.

Temporary working space: B104-110, B_t

Notes 1. When entered by either route, switches are reset as follows:-

a) Digit 23 of the switch directory is set to 1.

b) Process and full recover switches are reset to the values preserved in (55/239). (They are assumed "set" in the directories on entrance to R225 - usually set by R239).

c) The page occupied by this routine is locked down and the page number recorded in the current program branch indicator.

2. On entry with B126 even, all filled entries in the branch table are scanned. If halted in supervisor, the branch is made free; if halted out of supervisor for block b, the branch is made free if b is not locked out. If halted for another branch, the branch is left halted. Subsequently program scan enters R224 to find next branch, which recovers each free branch in order. The branch change marker is assumed set (by R239) and is not changed by R225.

R226 Change working registers.

Purpose: To interchange public B lines and current program working space in subsidiary store from the current program to a new program, and to set the page address registers appropriately, and to record the dump position of the previous current program.

Registers of fixed store: 94

Number of instructions obeyed: Minimum 360 + 3P + 5S
Maximum 360 + 10P + 5S
Where P is the number of pages, S the no. of words of subsidiary store to be moved.

Subsidiary store:

(18) = "Current program unit counter"
One half word in the current program working area holding the current accumulated instruction counter modulo 2048.

(11) = "Program change loop"
Four words holding the program

11)	113	122	121	0
	101	122	121	0
	202	126	121	-2(0)
	121	126	0	(3/226)

Half words 0.4(11) and 1.4(11) are available as working space for other routines. They are reset by R226

Parameters Used: (1) to (26)

Cross references:

(4)	=	(5/229)	0.1 sec. clock
(5)	=	(13/205)	Program wait timers
(6)	=	(9/227)	Current program wait time
(7)	=	(3/303)	Alternative entry to "Update timers"
(9)	=	(5/203)	Current program number
(10)	=	(15/204)	Program switch directory
(12)	=	(6/203)	Page directory
(14)	=	(2/203)	Block directory
(16)	=	(9/205)	Current program in store control
(17)	=	(20/303)	Program instruction counter timers
(21)	=	(1/227)	Entry to Resume new program
(22)	=	(9/202)	Current program branching switch
(23)	=	(12/239)	Entry to "Suspend program branching"
(25)	=	(35/203)	Block location table

Other Parameters

(8) = Fixed store table of dump block position. 7 half words, entry i holding n i (digits 23-3) where n is the no. of full words in a dump area.
(13) = No. of pages (digits 23-3) Digits 2-0 zero.
(19) = Leading address in subsidiary store area to be moved (digits 23-3)
(20) = Number of full words in subsidiary store area to be moved, digits 23-3
(24) = V store instruction counter = *600434

R226 continued

Connection with other routines

Entry at (1) from R223 with
B104 = dump block label, digits 22-12.
Remainder irrelevant
B101 = current program number
B109 = Program switch directory of new program
Word 0 of block location table = new program number

Exit (a) to "Suspend program branching" if current program involved in branching, with B101, 104, 109 as on entry. Return to (26) with these registers and block location table unaltered.

(b) to "Resume new program" at (1/227) with current program no. and current program in store control both holding new program number, and all working registers switched.

Alternative entry at (2) to switch registers with

B104 = address to which current is to be dumped digits 22-3. Rest zero
B105 = address from which now is to be recovered digits 23-3. Rest zero
B103 = no. of words of sub. store to be moved, digits 8-2 Rest zero
B107 = Return address
Return with B103 - 105, Bt altered.

Subroutine

"Update timers" entered at (3/303) with
B110 = Return address
Return with B104, 109 block location table unaltered.

Temporary working space: Entry (1) B100 - 110, Bt, block location table, 0.4(11), 1.4(11)
Entry (2) B103 - 105, Bt, 0.4(11) 1.4(11) if B106-0

Notes:

1. The current time is recorded in the Program Wait Directory of the current program, and this quantity is subtracted from the accumulated waiting time.

2. The page timers are updated before registers are switched.

3. The current V store instruction counter is added to the Unit Timers of the current program.

4. An empty section is preserved in each dump block to accommodate the current program. The location of this is held in half word 0.0 of the dump block.

R226 continued

5. The page address registers and clock digit of the page directory are set as follows

   a) If page directory locked out : Untouched

   b) If page belongs to neither new or old program : Untouched

   c) If page belongs to new program : PAR = b digits 22-12. Digit 23 = 0. Clock digit (13 of Page Directory) = 0

   d) If page belongs to old program : PAR *4. Clock digit = corresponding digit of program switch directory.

   "Contents of PAR" in subsidiary store are unaltered, and the PAR are changed by a direct transfer, since previous contents must be *36 or *4

6. The dump area holds

   Word 0 L
   1 Digits 39-37 Last 3 bits M; 32-26 B121; 23-0 B119
   2 M x $\frac{1}{8}$
   3 Subsidiary store working area
   3+S B127
   B1, B2, ... B99

   When formed during program branching by entry at (2) with B106 $\neq$ 0, registers 3+S on hold B99, B98, ... BN, B127
   The dump block must be in core store entry at (2).

16.7.62

R227 Resume new program

Purpose: A supervisor routine entered after switching working registers to check and resume a new program. The instruction counter timer and wait timers are reset, the latter being checked against the permitted limit; the supervisor exit trap is set to normal or monitor; any current input or output pages are locked out in core store and the program is resumed "in supervisor" if required

Registers of fixed store: 42

Instructions obeyed: 37 minimum (normal)
52 maximum

Subsidiary store:

(8) = "Program monitor directory"
A sequence of half words, entry P corresponding to program P holding monitor information when the program is not current (e.g. off line tape faults etc.)

Digit 23: 1 if already trapped for off line monitor
Digit 22-12: Marker digit for type of fault
Digit 11-3: Page no. if "page locked down" fault
Digit 0: 1 if awaiting entry to monitor, 0 otherwise

(9) = "Total wait time" One half word in current program working area holding total wait time for tapes etc., units 0.1 secs. in digit 22-2.

(10) = "Limit wait time" One half word in current program working area holding maximum permitted wait time, units 0.1 secs. in digits 22-2

(12) = "Input/output blocks" One word in current program working area, first half word for input, second for output, each holding Digit 12-2 block directory location of currently selected input/output relative to the start of the area for the current program. Remaining digits unused by this routine.

(16) "Number of program changes". One half word holding total number of program changes, digit 23-2.

Parameters used: (1) to (24)

(13) = "Input block no." digits 22-12 Rest zero. Block no. used for current input.

(21) = "Additional Wait time" time added to allow for monitor digits 22-2. Units 0.1 secs.

(22) = "Monitor marker for exceed wait time". One digit in range 22-9, rest zero.

(23) = V store instruction counter = *600434

Cross references:

(2)	=	(1/202)	Program Scan
(3)	=	(2/222)	Alternative entry to Supervisor Program Change
(4)	=	(1/709)	Off line trap routine
(5)	=	(2/700)	Monitor interrupt and SER
(6)	=	(18/202)	Normal setting of supervisor exit trap
(7)	=	(13/205)	Program wait timers
(11)	=	(16/202)	Supervisor exit trap
(14)	=	(2/203)	Block directory
(15)	=	(9/204)	Program Status directory
(17)	=	(4/203)	Program store directory
(18)	=	(18/226)	Current program unit instruction counter
(19)	=	(20/303)	Program instruction counters
(20)	=	(5/203)	Current program no.

Connections with other routines

Entry at (1) from R226

Exit (a) to "Program Scan" if out of supervisor resume

(b) to (2/22) if in supervisor resume with
B110 = Program number digits 8-2
B109 = 0.2

Entry at (24) for subroutine "Set P.A.R. of input/output block"
B100 = 0 (input) 0.4 (output)
B108 = New block label, digits 23-12.
Remainder irrelevant
B110 = Current program number, digit 8-2.
digits 23-9 zero
B109 = Return address R

Exit to R with P.A.R. altered, B107 altered

Subroutine "Monitor interrupts and SER" entered at (2/700) with
B126 odd
B100 = Marker "Exceed wait time"
B102 = Return address R

Exit to R with B110 unaltered

Temporary working space: B100-102, B107-110, Bt

Notes:

1. The unit instruction counter is modified by subtracting the V store clock.
2. If the allowed wait time is exceeded, a standard quantity is added to the limit and the program is monitored.
3. If an off line monitor is ready and the program is not already in such a monitor routine, the supervisor exit address is set to R709. Otherwise it is set to the normal value.
4. The current input/output blocks have their PAR locked out if in core store.

R228: Fixed store program branch routines

Purpose: A fixed store routine to check the validity of branch extracodes and to enter locked down main store routines to implement them.

Registers of fixed store: 15

Instructions obeyed: 4 to 5 depending on entry

Parameters used: (1) to (16)

Cross references:

(4)	=	(1/238)	Start branch routine
(5)	=	(3/238)	Kill branch routine
(6)	=	(2/238)	Halt branch routine
(7)	=	(1/201)	Enter supervisor
(8)	=	(7/202)	Current program branch indicator
(12)	=	(1/237)	Main store establish branching
(13)	=	(2/233)	Enter processing
(14)	=	8(18/202)	Return to main control
(16)	=	(56/237)	Difference block labels (-512)

Connections with other routines

Entry from Extracode jump table at

(1)	for "Start branch B at S"
(2)	for "Kill branch B"
(3)	for "Halt if branch B active"
(10)	for "Jump to S if branch B active"
(11)	for "Establish branching, B Branches"

B121 = B, B119 = S

Exit: After entries (1), (2), (3), (10) :-

To (1/201) to Enter Supervisor with

B91 = digits 7 - 2 of B121

B96 = (1/238) for entry (1)
(3/238) for entry (2)
(2/238) for entry (3)
-0.7 (2/238) for entry (10)

Exit from there to R238 in supervisor (locked down in main store)

If branching not in use, an illegal instruction, 0, is encountered at 7(1/228) causing monitoring.

Exit after entry (11) if branching not in use to enter processsing (local) at (2/233) with B126 odd
B97 = (1/237) with modified block Label (see R237)

Exit from there to (1/237) in extracode (not in supervisor)

Exit if branching already in use to the exit order in (14/228)

Alternative Entry: to R228 at (15) from R238 if all branches killed in main control to End Program extracode instruction

Temporary working space: B91, 92, 96, 97

Notes:

1. Extracode "Establish branching" is continuedin main store by copying a block from the Supervisor into the main program area. The continuation, R237, is entered in extracode control, not in Supervisor, with the local process switch set.

2. The remaining extracodes are continued In Supervisor in R238, which is locked down in core store whilst branching is in use for the current program. Information in extracode B lines is therefore preserved on entry to R238, as no halting can occur.

R229

R299 Clock interrupt routine R229/1

Purpose: To analyse interrupt from instruction counter 0.1 sec. clock and 1 sec. clock and to enter appropriate routine.

Registers of fixed store: 34

Instruction obeyed:

Instruction counter interrupt : 2 under interrupt control
0.1 sec. interrupt : 12 (minimum) " " "
60 (maximum) " " "
1 sec. interrupt : 13 under interrupt control

Parameters used: (1) to (19)

(13) = clock "look at me" line = *600434
(14) = one second clock in V store = *600431
(18) = exit from interrupts = *4004
(19) = tape channel registers = 8*6003

Subsidiary store:

(5) =	"0.1 sec. clock"	One half word holding time in units of 0.1 secs. digits 23 - 1.
(6) =	"Main clock"	One half word holding copy of V store clock, units 1 sec., digits 22 - 0.
(9) =	"Tape faults"	One half word holding 1 in digit i if block address interrupts have failed on channel i and are being dealt with by the appropriate SER.
(10) =	"Excess clock counter"	One half word holding number of 0.1 sec. interrupts since last 1 sec. interrupt, digits 5 - 1.
(15) =	"Miscellaneous marker"	One half word. Digit 2 = 0 if one second completed, 1 otherwise. Remaining digits unaltered and unused (by this routine)

16.7.62

R 229 continued

Cross References:

(2)	=	(1/301)	Entry to Instruction Counter Interrupt Routine
(3)	=	(1/230)	One Second Routine
(4)	=	(3/299)	Entry to channel interrupt Fail monitor routine
(7)	=	(57/402)	F1 Tape flip-flop
(8)	=	0.4(57/402)	F2 Tape flip-flop
(10)	=	(54/402)	Block address interrupt jump Power of 2 - alternate half words, entry $n=2^n$, n=0-7
(16)	=	(4/201)	Entry to Supervisor
(17)	=	(3/201)	Entry to Supervisor

Connections with other routines:

Entry at (1) from R500 on detection of clock or Instruction Counter Interrupt.

Exit: (a) To Instruction Counter Interrupt routine if look at me is set, without resetting look at me.

(b) To "Enter Supervisor" at (4) if one second clock interrupt and previous SER has been completed with

B111 = 0.4
B112 = Entry address of one second SER.

(c) To 2048*4 if 0.1 sec. interrupt and no tape faults are detected or if previous tape faults are still being dealt with. Also if 1 sec. interrupt and previous SER is not completed.

(d) To "Enter Supervisor" at (4) if tape fault is detected, with B112 = Entry address of Channel Interrupt Fail SER.

Temporary working space: B111 - 115, B123.

Notes:

1. If a one second interrupt occurs and the previous one second SER is still in progress, the SER is not re-entered. If the previous SER is completed, the SER is entered to the SER queue. The clock is recorded in subsidiary store and the "excess clock" set to 0 by the 1 sec. interrupt.

2. If a block address interrupt fails to arrive on channel n, digit n of F1, F2 both read 1. For all such channels, the TCR's are set to "End Transfer" and "stop" and the block address interrupt jumps are set to 2048*4. Faulty channels are recorded in subsidiary store and a maximum of one SER is entered to the tape queue by the 0.1 second interrupt routine. F2 is reset by the routine to F1. F1 is reset to 0 for faulty channels.

R230 One Second S.E.R.

Purpose: An SER entered every second by the clock interrupt routine. Detects tapes engaged, deck timers elapsed, entering appropriate SER's to the SER queues; enters test routines every 2 seconds, and schedule routine, long term scan and a timer SER when appropriate; checks the arrival of drum interrupts; fi ally transfers control to the one second routines for slow peripherals.

Registers of Fixed Store: 64

Number of Instructions Obeyed: 42 (normal) to 170 (maximum).

Subsidiary Store:

(7) = "Previous tape engage digits"
One half word containing copy of tape engage digits (digit = 1 if tape disengaged) at last one second SER in digits 15-0.

(8) = "Newly engaged tapes"
One half word containing 1 in digit n if tape n newly engaged and being analysed by an SER. Normally reads zero.

(9) = "Deck timer directory"
Alternate half words, entry n for deck n, containing
Digit 23 : 1 if timer set, 0 otherwise
Digit 22-15 : Timer. Reads "all ones" when timer not set.
Digit 14-2 : No. of blocks on tape after block 0 read. Zero if special check. Non zero otherwise.
Digit 0 : 1 if tape in normal use
0 if tape reading block 0 or in special check
Digit 1 : 0

(11) = "Timed exit criteria"
One full word
0.0 = Time for entry to variable timed routine (units 0.1 secs. digits 22-2)

(12) = "Timed exit addresses"
0.0 = Entry address of variable timed routine
0.4 = Entry address of long scan routine

(14) = "Schedule timer"
One half word containing time for entry to schedules, units 0.1 secs., dig. 22-2.

(16) = "Total time of test routines"
One half word containing total time test routines have run for, digits 23-0.

Parameters used: (1) to (31)

(25) = Minimum duration of test routines in 2 second intervals, digits 23-0
(28) = Failure marker for "Drum transfer incomplete"
(30) = No. of decks, digits 22-3, rest zero
(31) = Tape V store = *6003

Cross references

(6)	=	(1/525)	Entry to Slow Peripheral One Second Scan.
(10)	=	(54/402)	Block address interrupt jumps
(13)	=	(6/229)	One second clock
(15)	=	(12/222)	Local time on tests
(17)	=	(10/313)	Drum check location
(18)	=	(15/229)	Miscellaneous markers
(19)	=	(2/206)	"Enter SER to Queue"
(20)	=	(1/217)	Tape exit to supervisor control
(21)	=	(4/299)	Entry to "B.A. Interrupts failed".
(22)	=	(1/405)	Normal setting of BA interrupt jump for recover long search.
(23)	=	(5/299)	Entry to "Tape Engage"
(24)	=	(1/231)	Entry to "Schedule execution programs"
(26)	=	(5/229)	0.1 sec. clock
(27)	=	(6/299)	Entry to Test Program
(29)	=	1(2/340)	Entry to Drum Monitor

Connections with other routines:

Entry at (1) with B100 = 0.4

Exit: (a) to "Slow peripheral one second scan" which eventually exits to Program Scan.

(b) to "Drum Monitor" under interrupt control with B111 = Fault marker "Drum transfer incomplete" if drum failure detected.

Subroutines "Enter SER to Queue"
Entered at (2) with B126 odd
B107 = Queue number (0 = Slow, Tape, 2.0 Drum)
B109 = Entry Address
B108 = Entry information
B110 = Return address
Return with B106, 107 altered.

"Tape exit to supervisor control"
Entered at (1) with B109 = Deck number
B110 = Return address.

"Schedule execution programs"
Entered at (1). Return to 4/230. B lines irrelevant on entry and exit.

"Test programs"
Entered with B109 = Time Limit
B110 = Return address
B lines irrelevant on exit.

Temporary working space: B100 - 103, B107 - 111, Bt.

Notes:

1. After entry, digit 2 of "Miscellaneous marker" in subsidiary store is set to 1 to permit entry of another one second SER to the SER queues.

2. This routine is never halted.

3. The routine assumes control to be in fixed store; modification is necessary if it is run in main store.

4. The routine assumes one deck per channel, and requires minor modifications to deal with tape switching.

5. If any tapes have been engaged since the last One Second SER, they are recorded in subsidiary store and, if not already active, an SER is entered to the slow queue to take action on all such tapes. Disengaged tapes are not recorded in subsidiary store.

6. If any deck timer becomes exhausted action is either (a) Set SER to bring to normal speed and block address jump to cause line up of the tape. (b) Assign tape to previously specified tape SER in the case of addressing and test routines. (c) Enter a monitor SER to the tape queue and disengage the tape if no block 0 is detected after the assigned interval.

   Conditions a, b, c are set up in the deck timer directory when the timer is set. To cause an interval of n-1 to n secs. to elapse, set the timer by adding n in digits 22-15 to the timer directory.

7. One timed routine can be entered to the slow SER queue. The entry address and time of entry are set in subsidiary store. The time of entry to this and the "long scan" routine are reset by R230 to hold 0.111..... preventing further entries till these times are reset. Test routines are entered every 2 seconds so that the total time spent in tests since the last one second SER reaches a limit. If this limit has already been reached, test routines are not entered.

8. The drum check location is set positive if a drum transfer is expected, and monitor is entered if it remains positive at the next one second SER.

R231 Schedule execution programs.

Purpose: A supervisor extracode routine entered by the One Second SER at intervals of not more than 4 seconds to re-order the priorities of programs in the execute list. Only marked programs are considered and these are inverted in pairs if the computing time of one program exceeds that of the next program of lower priority.

Registers of fixed store : 41

Number of instructions obeyed: Minimum 7
Maximum 7+16 P where there are P entries in the execute list.

Parameters used: (1) to (9)

(2) = Interval of entry to this routine, units 0.1 secs. digits 22-2. Rest zero.

Cross references:

(3)	=	(14/230)	Scheduled timer
(4)	=	(9/207)	Address of leading program no. relative to start of program status directory.
(5)	=	(9/204)	Program status directory
(6)	=	(8/227)	Program monitor directory
(7)	=	(20/303)	Program instruction counter directory
(8)	=	(6/202)	Program change marker
(9)	=	(4/230)	Return to one second SER.

Connections with other routines:

Entered at (1) with B101 = 0.1 sec. clock.

Exit to (4/230) with B101 altered and schedule time reset to cause next entry.

Temporary working space: B100-104, B109-110, Bt.

Notes:

1. Only programs with d14 of program monitor directory = 1 are considered.
2. For these programs d12-7 of monitor directory holds d14-9 of instructions counter at last entry to R231. Only these digits are considered in comparison of computing times.
3. If two programs A, B are adjacent in priority, A higher than B, and A has computed longer than B in the last interval, their priority positions are interchanged and the routine then considers the remaining programs below both A and B in priority. If A has computed less than B, B is compared with the next program etc. If any change in priority is made, the program change marker in subsidiary store is set non zero to cause re-selection of a main program.

R232: Branch block monitor R232

Purpose: An SER in locked down main store entered from R703 when store allocation or block label is exceeded by the current program to step back by 1 the main control of the branch in current control (which may be in supervisor control only, with M recorded in its dump).

Registers of main store: 13

Instructions obeyed: From 1 to 12 depending on state of branch.

Parameters used: (1) to (4), (50) (53) (54)

Cross references:

(2)	=	(1/202)	Program scan
(3)	=	-1(2/703)	Reference point in R703
(50)	=	(50/238)	Branch table
(53)	=	(53/238)	Branch change marker
(54)	=	(54/238)	Current branch
(4)	=	(10/224)	Current branch in store

Connections with other routines:

Entered at (1) from R703 with
B101 odd if non equivalence

Exit: a) To 9(2/703) if non equivalence. No registers altered

b) To 7(2/703) if branch in control of store. B109 altered

c) To Program Scan if branch not in control, with Branch change marker set, and current branch set to resume:-

In Supervisor, in control of store,
with B100, 101 preserved re-entry at 5(2/703)

Temporary working space: B109, B110

Notes:

The routine supplements R703 in that it arranges for R703 to monitor the program or extend the available store, and to implement the latter, M is reduced by 1 unless the routine was entered via non-equivalence. Thus the program can be resumed by direct re-entry after, for example, store blocks are supplied.

R233 Enter Processing

Purpose: An extracode subroutine to permit extracode and other extensions of an object program such as monitor to be obeyed from main store. A reserved sector on the drum is called to one level store and becomes a part of the object program. This calling routine is designed to give rapid access to such a block already existing in core store.

Registers of fixed store: 29

Number of instructions obeyed: 22 to 24. 22 if required block already in core store. A maximum of 19 orders obeyed with interrupts inhibited.

Parameters used: (1) to (10)

Cross references:

(3)	=	(5/203)	Current program number
(4)	=	(8/227)	Program monitor directory
(5)	=	(4/203)	Program store directory
(6)	=	(2/203)	Block directory
(7)	=	(6/203)	Page directory
(8)	=	(1/235)	Entry to "Call Supervisor Sector to main program
(10)	=	(1/201)	Entry to "Enter Supervisor"

Connections with other routines:

Entry at (1) to set Master Process Switch before entering required routine (See note 3).
at (2) to set local switch.

In both cases, on entry

B97 = Address of entry to main store extension, E. If digit 0 = 1, B127 must also contain E on entry, and a routine will be entered at E under main control. Otherwise the routine is entered at E under extracode control.

B126 = must be odd on entry at (1) or (2) (digit 0=1)

Exit: a) To address E with B93, 96, altered but otherwise main B line and extracode working space unaltered.

b) To "Call Supervisor Sector to Main Program" if the required sector is not already attached to the program, with

B97 as on entry
B93 = location in block directory which the block will occupy relative to the start of the block directory.

This entry is via entry 1(1) of R201, entered with interrupts inhibited.

Temporary working space: B93, B96-97, B100-102

Notes:

1. Before calling the required block into use, the process switch in the program store directory is set: this permits access via R203 to all block labels attached to the current programs and the main store routine must check itself that reference to blocks of the main program are legal. The supervisor block itself has a reserved block label, *3400.

2. If B97 is negative on entry (private store address) the process switch is set, and an immediate exit is then made to E (either in main or extracode control). Such an entry may be used to set the process switch.

3. Entry (2) is used for local setting of the process switch by specific extracodes.
Entry (1) causes a master switch to be set also in the program monitor directory, so that subsequent exit to reset the local switch from an extracode does not in fact reset the switch. This is used during monitor routines in main store.

4. The last position in the block directory area for the program is permanently reserved for a supervisor block. If a new block is required, the previous block in use is lost and if required again will be called from its reserved sector.

5. If the required block is already in core store, the page address register is set up and the page directory is unlocked to permit normal use of this block.

R234: Free branch

Purpose: An SER in locked down main store entered from the central co-ordinator via R205 when branching is in use, to free any branch halted for a given reason. Control returns to R205

Registers of main store: 18

Instructions obeyed: 3 + 5N to 3 + 14N where N branches are in use.

Parameters Used: (1) to (4), (50), (52), (53)

Cross References:

(50)	=	(50/238)	Branch table
(52)	=	(52/238)	Top priority branch
(53)	=	(53/238)	Branch change marker
(4)	=	2(12/205)	Exit to R205
(3)	=	(9/205)	Current program in store control

Connections with other routines:

Entry at 1) from R205 with
B108 = 0, B109 = reason for free (0 to 3.4) as for R213
B108 ≠ 0, B108 = block label, digits 22-12 of block unlocked.

Exit to (4) with 107 altered
B106 is used but reset to the current program number.

Temporary working space: B107

Notes:

1. The halt reason in word 0.4 of each entry in the branch table which is halted is tested for the required reason. Digits examined are

   p23 = 0 if halt for block, 1 otherwise
   p22-12 = block label if halt for block
   p0 = 1, p4-2 = reason for halt, remainder zero if halt for listed reason.

2. Branches are made free by setting digit 0 of word 0.0 of the entry to zero. The branch change marker is set non zero to cause subsequent entry to find next branch.

R235 Call supervisor sector to main program.

Purpose: A supervisor extracode routine to order a drum transfer to read a block from a reserved sector to the core store and to incorporate the copy as part of the one level store of the current main program.

Registers of fixed store: 19

Number of instructions obeyed: 14+5n where n entries in the block directory are scanned before the required sector is found.

Parameters used: (1) to (8)

(7) = Total number of entries in Block Directory, digits 12-2.

Cross references:

(2)	=	(2/203)	Block Directory
(3)	=	(5/203)	Current program number
(4)	=	(4/203)	Program store directory
(5)	=	(14/317)	Alternative entry to "Lose block b"
(6)	=	(14/314)	Alternative entry to "Drum transfer routine"
(8)	=	(3/203)	Block Timers

Connections with other routines

Entry at (1) with

B93 = location in block directory relative to the start of the directory to which block will be assigned.

B97 = block label of block to be called, digits 22-12. Remaining digits irrelevant.

Exit to -4(14/314) to make entry in drum queue with

B100	=	Digits 20-14	Current program number
		Digits 12-2	Directory position of block in area of current program relative to the start of this area. Remaining digits zero.
B102	=	Digits 12-2	Directory position of reserved sector relative to the start of the block directory.
		Digit 13 = 1	Remaining digits zero.

Subroutine:

"Lose block b". Entered at 6(14/317) with

B100 = Return address

B105 = Directory entry (relative to start of block directory)

Exit with B105 unaltered, and with the previous block occupying this positions lost.

Temporary working space: B100-103, B105

Notes:

1. Reserved supervisor sectors are recorded at the base of the block directory with

   digit 22-12 block label to be attached when block called to one level store

   digit 11-1 absolute sector number of reserved sector.

2. No provision is made in R235 for absence of one of these sectors.

3. The previous block occupying the one level store location whose directory entry is to be used is lost.

4. The entry parameters to the drum queue cause the drum routine to treat this request as "Duplicate block of program 0 on drum to be block of current program in core store". No lock out is applied to the reserved sector in the block directory, but the one-level store block is locked out with S = 1.

R236 Exit from processing

Purpose: An extracode subroutine entered on the conclusion of extracode and main program extensions in main store. After entry to such routines by R233, exit is made via R236, which resets the process switch and locks out any block of store used by the extracodes. Provision is made for return to a previous block of system program.

Registers of Fixed Store: 25

Number of Instructions obeyed: 10 to 20. 20 in normal case of emerging from an extracode to object program. A maximum of 19 instructions are obeyed with interrupts inhibited.

Parameters used: (1) to (8)

(7) = - ⌊ lowest reserved block label in digits 22-12 ⌋ = * 44

Cross references:

(2)	=	(5/203)	Current program number
(3)	=	(8/227)	Program monitor directory
(4)	=	(4/203)	Program store directory
(5)	=	(2/203)	Block directory
(6)	=	(2/223)	"Enter processing"
(8)	=	(6/203)	Page directory

Connections with other Routines:

Entry at (1) in extracode control with
B126 odd
B127 resume address R

Exit a) To the address in B127 under main control

b) To "Enter processing", entered at (2), with
B97 = R (digits 23-1), digit 0 = 1
B127 as on entry.

When the master process switch is set and R is a reserved block label.

Temporary working space: B101 - 103, and for exit (b) only B97

Notes:

1. This routine forms the equivalent of the "return to Main Control" facility in hardware. It exits to main control after locking out any core store block used by an extracode in main store.

2. If the master process switch is not set, the local process switch (digit 1 of the Program Store Directory) is set to zero. Otherwise the local switch is left set.

3. If the master switch is not set or the exit address in B127 is *3400 the last entry in the block directory for the current program is inspected. If the corresponding block is in core store, the corresponding page directory is locked out and the PAR is set to *4. Exit is then made to main control.

4. If the master switch is set and the exit address is *3400, R233 is entered to cause re-entry to processing at the address in main control. Final exit is to the required address in B127 under main control.

R237: Establish branching

Purpose: A main store routine entered in extracode control via R228 from the extracode "Establish branching". Sets up initial branch table and parameters and enters "Resume branching" to set up the program as branch 0.

Registers of main store: 55 (overwritten by dumps during the subsequent operation).

Instructions obeyed: Up to 37 + 10B for B branches, plus entry to "resume".

Parameters used: (1) to (13), (50) to (54), (56), (57)

(6) No. of words in short dump, digits 23-3

Cross references:

(4)	=	(10/224)	Current branch in store
(7)	=	(5/203)	Current program number
(8)	=	(4/203)	Program store directory
(9)	=	(2/203)	Block directory
(10)	=	(1/201)	Enter supervisor
(11)	=	(1/324)	"Rename" extracode
(12)	=	(1/225)	Resume branching
(13)	=	(28/224)	B reference in "Find next branch"
(50)	=	(50/238)	Branch table
(51)	=	(51/238)	Last entry in table
(52)	=	(52/238)	Top priority entry
(53)	=	(53/238)	Branch change marker
(54)	=	(54/238)	Current branch

Connections with other routines:

Entered at 1) via "Enter processing" with local process switch set
B121 = B
B119 = 0 (short recovery) ≠0 (long recovery)
Entry is in extracode control, not in supervisor.

Exit to (1/225): Resume branching : Entered with B126 even

Exit to Resume branching and thence to program scan.

Subroutines:

"Rename" extracode entered at (1/324) via 1(1/201) with
B97 = Return address in extracode in R237
B96 = 2(1/324)
B91 = new block label
B119 = old block label
Exit via program scan to return address.

Temporary working space: B91-97, 119, 121, B_t

R237: continued

Notes:

1. R237 occupies the end of the block holding R238 etc. and is subsequently overwritten by branch dumps. The block is copied to the main program area by "Enter Processing", and occupies the last position in the current program area of the Block Directory. When called it has block label B-1. It is moved from this entry in the block directory by renaming it B. The block is originally loaded with label B which must later be changed in the block directory to B-1.

2. Initial settings of branch parameters include last entry in table, current entry (set up at the end of the branch table as Branch 0), top priority entry, lowest B line to be switched, and initial dump areas, recorded in word 4 of each entry in the branch table. These are established from the last entry in the table so that one area remains "empty" in each block should more than one be required. The address of this empty area is recorded in the last half word of each dump block, new blocks being created, if necessary, by non equivalence under extracode control.

3. Branch changing parameters (length of dump, range of B lines) are set up by R237 from B119 on entry (lowest B line to be preserved). If this is specified in excess of B99 (digits 8-2) it is treated as B99, preserving one B line only.

R238: Main store branch extracodes

Purpose: To implement extracode instructions "Start branch B at 5" "Kill branch B", "Halt if branch B active" and "Jump to S if branch B active".

Registers of main store: 80

Instructions obeyed: Start: 11 + 4n where n bottom entries in branch table are filled. Kill & test: Vary markedly with state of branch table.

Parameters used: (1) to (17); (50) to (54)

(50) = Branch table — One entry per branch, each entry 10 half words as follows:

- 0.0 :- p23 = 0 if in use, 1 if not in use
  p7-2 = Branch number
  p1 = 1 if newly started branch
  p0 = 1 if branch free, 0 if halted
- 0.4 :- Halt reason (see R239)
- 1.0 :- Position in branch table of branch of next lowest priority relative to start of branch table, digits 23-0.
- 1.4 to 4.0:- Dump for re-entry address, B100-104 (see R239)
- 4.5:- Address of dump area for this branch

(51) = -2(50) Contains address of word 0.0 base entry of branch table, relative to start of branch table = 5(B-1) where B branches established.

(52) = 0.4(51) One half word containing address relative to start of branch table of word 0.0 of entry of top priority branch.

(53) = 1(51) One half word containing zero if no branch status changed, non-zero otherwise.

(54) = 1.4(51) One half word containing address relative to start of branch table of word 0.0 of entry of current branch.

The above occupy the same block as R238 in main store, and are set up initially by R237 when branching is established.

(10) = Monitor setting : Excess branches

Cross references:

(9)	=	(1/202)	Program scan
(11)	=	(14/700)	Monitor
(13)	=	(15/228)	"End Program" extracode
(14)	=	(2/239)	End branching

R238: continued

Connections with other routines:

Entry at (1) for Start B
(2) for Halt if B active
-1(2) for Jump if B active, B126 odd
(3) for Kill B

via R228

All entries In Supervisor B91 = B (digits 7-2)
B121 = B (digits 8-2)
B119 = S

Exit a) To Program Scan normally, with branch change marker (53) set non zero, which resumes main program after entry to branch change.

b) To monitor at (14/700) if branch table full on entry (1) with

B100 = Marker
B126 odd

Exit from there to Program Scan

c) To kill branching at (2/239) if all branches killed, with

B107 = Program Scan

Exit to Program Scan and thence to end program.

Temporary Working Space: B100-108, B_t

Notes:

1. On entry (1), the branch table is scanned from the base for a vacant entry. The branch is inserted with digits 1, 0 of word 0.0. both 1, and word 0.4 = entry address S. The new branch is recorded as priority B and is linked to the previous lower priority branch. If there is no vacancy in the table, the number of branches specified by the extracode "Establish" is exceeded, and the program is monitored.

2. On entry (2), the branch table is scanned to find any branch labelled B. If one is found, the current main control is set to S or the current branch is halted, the reason being recorded in 0.4 of the entry as

p23 = 1, p7-2 B, rest zero

If no entry is found, no change in status is made.

3. On entry (3), if B121$\geq$64, the entry for the current branch is vacated. If B121$<$64, entries for all branches labelled B are vacated. The table is then scanned for any branches halted for B; if the current branch was killed, this scan is only made if no other branch has the same label. Halted branches are made free. If all branches have been killed, End Program is entered after concluding branching.

4. After any change in status of the branch table, the branch change marker (53) is set non zero. Before exit to main control, R224 will be entered to effect any change of registers.

R239: Halt branch

Purpose: An SER in locked down main store whilst the program is in control, entered from the central co-ordinator and other routines to halt a branch for a specified reason. Depending upon the method of entry, the current branch is halted or program branching is suspended and the central co-ordinator re-entered to halt the entire program. An alternative entry provides suspension of branching and return to a specified address.

Registers of main store: 49

Instructions obeyed: 16 to 24 to halt current branch
32 to 39 to suspend branching and re-enter co-ordinator.
24 to suspend branching as a subroutine.

Parameters used: (1) to (14), (50), (53) to (55)

Cross references:

(3)	=	(5/201)	SER Re-entry address
(4)	=	(9/205)	Current program n
(5)	=	(7/202)	Current program branch indicator
(6)	=	(6/203)	Page directory
(7)	=	(9/204)	Program status directory
(8)	=	(15/204)	Program switch directory
(9)	=	(1/225)	Entry to "Resume branching"
(10)	=	4(6/204)	Re-entry to R204 after entry for halt program
(11)	=	(26/226)	Re-entry to R226 after suspend branching
(12)	=	(8/303)	Page timers
(13)	=	(4/203)	Program store directory
(50)	=	(50/238)	Branch table
(53)	=	(53/238)	Branch change marker
(54)	=	(54/238)	Current branch

Connections with other routines:

Entry at (1) from R204 (Halt Program) if branching in use with
B110 = Current program number
B108 = Return address Digit 0 = 0 (out of Supervisor halt)
1 (in Supervisor halt)
B126 odd on entry: R204 was entered at (17) for special halt
B126 even: B109 = Reason for halt:-
Listed reason n :- Digits 0=1 Digit 4-2=n. Rest zero
Block b :- Digits 22-12=b. Rest zero
Peripheral P :- Digits 23, 22=1, 20-14=P. Rest zero

Exit: a) Entry with B126 odd :- To (10). to halt program with branching suspended and with
B109 = *4
B108 : Digit 0 forced to 1
No other B lines altered.

b) Entry with B126 even.
Exit to address in B108 after halting branch with B109 altered.

R239: continued

Alternative Entry: at (2) for subroutine suspend branching with
B107 = Return address

Exit To return address with re-entry address set to (1/225)
B110 = Current program number
B108 altered

Alternative Entry: at (14) from Program change (R226) to suspend branching

Exit To (26/226) after suspension with
B110 = Current program number
B107, 108 altered

Temporary Working Space: Various of B107-110 depending on entry conditions, as described above

Notes:

1. If an In Supervisor halt is requested, B100-104, and the current re-entry are stored in words 1.4 to 4 of the current entry in the branch table and Digit 0 of the "halt reason" is set to 1. The branch change marker is set non zero in all cases at entry (1).

2. If only the current branch is to be halted, digit 0 of word 0 of the current entry is set to zero, and the reason for halt is recorded as

   Digit 0 : 1 if In Supervisor restart, 0 otherwise.
   If halt for block: Digit 23= 0, 11-1=0
   22-12 = block label
   Otherwise, digits 23=1, 22-1 = contents of B109
   22-1 on entry

3. If the program is to be halted (entry 1, B126 odd) the current branch is not recorded as halted. Branching is suspended (see note (4)) and R204 is re-entered to cause the program to be halted normally.

4. Branching is suspended by

   a) Resetting digit 23 of the Program switch directory

   b) Setting Current Branch indicator to negative

   c) Setting the process switch (store directory digit 1) and full recovery switch (p9, status directory).

   d) Setting the re-entry address to 1/225 to resume branching when the program is freed. When not in current control, a program therefore appears as if not branching

   e) Lock down is removed from the page of core store holding the branching routines and branch table, and the timer T is set to 3.

   The current value of the program status directory and the switch directory are recorded before alteration in (55), 0.4(55) to allow the switches to be reset on resumption. The initial values when branching is established are 64, 0 (no process switch, in supervisor recovery).

R240: Supervisor output routine

Purpose: A series of supervisor subroutines to enable SER's to output information through a buffer via the peripheral supervisor. Depending upon parameter settings, the routine may be used to print on operators or engineers output (share between SER's), to print directly on a private reserved output, or to form an output document for inclusion in the output well.

Registers of store: 135

Instruction obeyed:

Write one character	:	13 normally
Write string	:	25 per character
Layout	:	23 to 29
Break output	:	3 to 15

Parameters used: (1) to (31)

Cross references:

(7)	=	(5/201)	SER re-entry address
(8)	=	(1/513)	Restore character positions
(9)	=	7(3/230)	Enter SER to slow queue
(13)	=	(7/201)	Current SER base
(14)	=	(12/213)	Current SER dump address
(15)	=	(11/213)	Alternative entry to Halt SER
(16)	=	(5/203)	Current program number
(17)	=	(1/204)	Halt main program
(19)	=	(1/202)	Program scan
(21)	=	(7/503)	Start output
(22)	=	(64/599)	Location of "next address" in private store of peripheral
(23)	=	(66/599)	Location of "output fault marker" private store of peripheral
(25)	=		Location of "store reference" in private store of peripheral
(27)	=	(1/206)	Enter SER to queue
(29)	=		Enter to Output Scheduler : Document complete
(30)	=	(2/220)	Free operators output
(31)	=	(5/599)	Peripheral subsidiary store addresses

Connections with other routines:

Entry: At (1) for Write string of characters
B100 = Address of working store of peripheral
B108 = Maximum no. of characters less one (digits 23-3)
B109 = Starting address in store, digits 23-0
B110 = Return address
Digit 0 = 0 (retain B100 only) 1 (retain B100-104)
Return to return address with B100 or B100-104 preserved, re-entry address set to the return address. Characters transferred until counter exhausted or character zero (0 0 octal) encountered.

R240: continued

At (2) for Write 1 character
B100 = Address of working store of peripheral
B109 = Character (digits 5-0). Remainder irrelevant
B110 = Return address
Digit 0 = 0/1 (recover B100 or B100-104)
Return as for entry (1), The character is not transferred if it is zero

At (3) for layout
B100, 110 as for entries 1,2
B109 = Carriage control bits (as for extracode 1066) in digits 8-0
Remainder zero
Return as for entry (1)

At (28) for "break output"
B100, 110 as for entries 1,2

Exit: to (3) to output layout (final) if private direct to output scheduler if output indirect, B100, 110 unaltered to "free operators output" if common output with
B101 = Number of output, digits 3-2.
B110 = Return address. In this case the re-entry address is not reset on exit, and B100-104 are preserved.

Subroutines:

(a) "Enter SER to Queue"

Entered at 2/206 via 7(3/230)with
B107 = 0 (slow queue)
B108 = Entry information
B109 = Entry address
B110 = Return address

Entered at 1(1/206) with B126 odd
B107 = SER base, digit 2.1. Rest zero
B108 = Entry information or program number
B109 = Entry address. Digit 0 = 1 for recovery of B100-104
B110 = Return address

(b) "Halt SER" entered at 3(11/213) to plant B100-104 with
B107 = Dump address
B110 = Link

(c) "Halt main Program" entered at (1/204) to halt for peripherals with
B109 = Peripheral number, digits 8-2
B110 = Program number
Exit to program scan.

(d) "Program Scan" entered at (1/202)

R240: continued

(e) "Restore character positions" entered at (1/513) with
B107 = Quantity to be shifted
B108 = 0 (no shift)
0.1, .2, .3 (3, 2, 1 6-bit shift left)
B110 = return address
Return with B107 shifted, to return address, or to return address +1 if B108 = 0.

(f) "Start output"

Entered at (7/503) to start output with
B100 = Address of private store of peripheral
B101 = Address in store of 1st separator
B110 = Return address
Return to return address on conclusion or failure with
B100 preserved.
Entered again as above to restart output after fault.

Working Space: B105-110, B_t

The routine can halt before returning control, and hence the routine returned to can have been written to the drum. After any non equivalence halt, B100 or B100-104 are preserved according to the setting of digit 0 of the return address. When the master SER has base 0 (object program) B100-104 are always preserved, and B1-99 also if the full recovery switch has been set.

Uses working space in a block of 11 words reserved for the peripheral (see note 4).

Notes:

1. Characters are 6-bit internal code characters. The routine can apply to any output peripheral so long as the character set used can be printed correctly.

2. On entry (1), the characters are taken from store packed, and may start in any character position as specified by B109 (e.g. digits 1,0 of B109 = 10 start with digits 6-11; then 0-5, then to next half word). The transfer is terminated when a zero character is detected or when the counter in B108 is exhausted. This counter is stepped by 1.0 and tested for zero. If it contains 1 in any of digits 23, 2-0 then the transfer will continue till a zero character. The character to be printed must be within one main store block, or can be in fixed or subsidiary store.

3. The same subroutine may be used for thee purposes

a) To output on a central operators or engineers output, shared between all SER's. The output must be preceded by reserving the equipment by entry to R220, and must conclude by freeing the equipment, either by entry to R220 or via entry (28) of R240 (break output). The number of the output channel is replaced by R220 by the address of the working area.

R240: continued

b) To output directly, without going through the output well, on a privately reserved peripheral. This must have been taken out of the system and assigned to the SER by the operator. The working area is set up by the SER (see note 4) and it's address is supplied to R240. It is assumed that the SER is not based on the peripheral; if it is, straightforward use of the peripheral supervisor may be made.

c) To form an output document for transmission to an equipment through the output well. The scheduler is informed of this output.

4. A working area of 11 consecutive words is used for each peripheral under control, and the address of this is specified in B100 on entry to R240. The first few words must be set by the the user when R240 is used for purposes 3b, 3c. They are set up permanently by the Supervisor for the central shared output devices. The registers contain the following

0.0 Subsidiary store address of peripheral relative to start of subsidiary store table digit 0 = 0 (a,b) 1(c).

0.4 (18/240) (b,c). Address of log routine (a)

1.0 (20/240) (a,b). Address of output scheduler (c)

1.4 Number of central output (digits 3-2), digit 23=1 (a) Zero (b) Positive (c)

2 "Current transfer address" plus *1. Originally set to start of buffer area for printing ⌊ if this has address A, word 2 contains A+(*1) ⌋

2.4 "Last separater address" plus *1. Originally set the same as 2.

3.0 "End of current buffer area" plus *1. If this address = E, the routine will use half word 1.0 E*7 as the last word of the buffer.

3.4 "Sum of ends". Two buffers of equal length are to be provided in cases, a, b and this half word is the sum of their end addresses, each plus *1. In case c, this may be the same as 3.0 as only one buffer is required. Digits 1,0 must be zero.

4.0 Length of one buffer region: Number of effective half words in one buffer region such that C⌊ 3.0 ⌋ = C ⌊ 2 ⌋ + C⌊ 4 ⌋ orginally. Both buffer regions must be of equal length.

4.4 to 10.4 inclusive: Used as working space by R240.

R240: continued

5. The routine R240, the working area, the buffer, the calling routine, and the message to be transmitted may all be in seperate main store blocks provided the re-entry address and dump address if necessary are set up by the calling routine. The blocks will be called to core store if necessary by non-equivalence, and they must therefore all bear reserved block labels. It is planned that one block should be occupied by R240, working area and buffers of of the central equipments; if used for purposes b,c, it is probable that the calling routine, working area and buffers will be in a common store block. R240 will function wherever the areas so long as each lies within one store block if in main store.

6. When direct output is required, the output device is started if idle whenever a complete line has been assembled. When one buffer is filled, input is halted until the other buffer has been cleared to the printer and, in the case of central output, copies to the log. The calling routine may therefore be halted awaiting the printer if the buffer is insufficient to cope with peak demands. The printer is made idle when no further complete line has been assembled. If output ends through a fault, no notification is made to the operator except that the equipment is disengaged; the routine is halted until the equipment is re-engaged when output is resumed.

R246: Re-arrange block directory

Purpose: A supervisor extracode subroutine in main store to re-arrange the block directory when an object program is started or ended, when the storage allocation of a problem changes or when supervisory routines or compilers are transferred between tape and drum.

Registers of main store: 143

Instructions obeyed: Varies markedly with the contents of the block directory. Approximately 8N + 2A instructions obeyed to move object programs where N is the number of entries moved and A is the distance moved.

Subsidiary store:

(10) = "Start of Program area": One half word holding location relative to the start of the block directory of the first entry in use by object programs, M, digits 12-2. Remaining digits zero. The supervisor can use entries 0 to M-0.4.

Parameters used: (1) to (26)

Cross references:

(11)	= (2/203)	Block directory
(12)	= (3/203)	Block status directory/Timers
(13)	= (6/203)	Page directory
(14)	= (4/203)	Program store directory
(15)	= (9/204)	Program status directory
(16)	= (9/207)	Leading program number relative to (15)
(17)	= (3/315)	Address of last entry in drum queue
(18)	= (2/315)	Address of current entry in drum queue
(19)	= (20/315)	Drum queue
(20)	= (18)-(19)	
(21)	= 6(30/314)	Entry to drum Transfer routine after read from drum
(25)	= (25/314)	Address of Empty Page
(26)	= 7(1/312)	Address of order B126 = B110

Connections with other routines

Entry at (1) to aquire space in object program region with
B105 = D Location relative to start of block directory of the entry above which spaces are to be assembled, digits 12-2. Remaining digits zero. B107 = A + 0.4 where A = number of entries required in digits 12-2.
B110 = Return address R
Return to return address with B106-109, Bt altered
B106 = -A (p12-2)

R246: continued

Entry at (2) to release space from object program region with
B105 = D Location relative to the start of the block directory of the first entry to be released, digits 12-2 : Remaining digits zero
B106 = A Number of consecutive entries to be released, digits 12-2
B110 = Return address R
Return to return address with B107-109, B_t altered.

Temporary working space: Entry (1) : B106-109, B_t
Entry (2) : B107-109, B_t

Notes:

1. On entry (1) it is assumed that A entries in the block directory supervisor region are empty. On entry (2) it is assumed that A consecutive entries in the object program area from D onwards are empty.

2. When space is required for object programs, the last A entries of the supervisor region are cleared, if necessary by moving them up in the directory. Any object program areas above the position where entries are required are then moved up by A places.

3. When space is released from an object program area, any areas above are moved down by A places. Any supervisor blocks having block labels in the range M to M+A-1 are moved down to occupy the corresponding directory position in the supervisor area of store (from which they must have been moved by earlier entrance to this routine at (1).

4. When a directory position is moved, the page directory is altered appropriately if the block occupies core store. The page directory of the page involved in a read transfer from the drum, if any, is also altered. The drum queue is scanned, and any references to blocks of the supervisor which have been moved are compensated. The references to leading entries in the block directory for object programs are altered in the program store directory where necessary.

5. The limit of the supervisor area, both in the program store directory of Program 0 and in (10/246) are corrected by this routine. The record of number of free blocks is not altered; in particular, on entry (2), no programs halted for space are made free.

6. This routine is never halted, and leaves the S.E.R. re-entry address untouched.

R247: Co-ordinate organisational extracodes

Purpose: A subroutine to assist extracode routines to pass information between supervisor main store locations and an object program maintaining full protection without explicit testing of parameters

Registers of fixed store: 21

Instructions Obeyed: 5 or 6 on entry from an extracode
3 to load a B register and return to object program
6n + 2 to transfer n half words to or from object program store.

Parameters used: (1) to (10)

Cross references:

(5)	=	(1/201)	Enter supervisor
(6)	=	(1/215)	Set recovery switch
(7)	=	(1/202)	Program scan
(8)	=	(3/215)	Reset recovery switch
(9)	=	(99/900)	Extracode working space
(10)	=	(8/311)	Exit to main control

Connections with other routines:

Entry at (1) from extracodes of type "set ba", in extracode control not in supervisor
B91 = Address of supervisor routine to be entered

Exit to (1/215) to set full recovery with
B109 = B91 on entry
Exit thence to requested S.E.R., which may be in main or fixed store

Entry at (3) from extracodes requesting transfers to and from programmers store, in extracode control, not in supervisor, with
B119 = S (Programmers store)
B91 = Address of S.E.R. to be entered

Exit to (1/215) as after entry (1)

Entry at (4) on conclusion of S.E.R. in supervisor
a) if S.E.R. was entered via (1),
B91 = quantity to be loaded to Ba
b) if S.E.R. was entered via (3)
B119 = S (Programmer store)
B92 = (N-1) in digits 8-2 where N half word are to be transferred to or from S
Extracode working space (99/900) holding N half words to be transferred (note $N \leqslant 10$)
B91 = 0 if exit to main program required
or B91 = Address of S.E.R. to be re-entered
B97 = Even if transfer to S required
Odd if transfer from S required

Exit: To main program, or back to requested S.E.R. with
B93, 94, 96 altered
B97 altered to even
B119 stepped by N in digits 8-2.

Subroutine: "Reset full recovery switch"
entered at (3/215)
B109 = Entry to program scan, to resume program.

Notes:

1. This routine is used by those extracodes which require information transfer between an object program and information in supervisor main store. The latter cannot be referenced under normal extracode control, but only after entry to an S.E.R. Within an S.E.R. access to individual items of an object program is only possible after checking all references for legality (e.g. store space within the limit *34, Ba not B126 etc). To avoid these difficulties, R247 transfers to and from object programs under extracode control, not in supervisor, whilst an S.E.R. assembles information from supervisor stores into extracode B lines and working space. To achieve this, the full recovery switch is set on entry to the S.E.R. and is reset by R247 on exit, ensuring valid use of extracode working areas. Routines using R247 are expected to be of rare occurrence in time and the time consumed by entry to supervisor and indirect transfers is therefore not a serious factor.

2. If no information is to be transferred, entry (1) may be used, and B121 set to zero by the S.E.R. before exit to (4).

3. If more than 5 words are to be transferred, repeated entry must be made from the S.E.R. to (4), setting B91 to cause return. On return, note that all supervisor B lines are lost since R247 exits from supervisor. Note also that when reading from object program (comparatively rare) B97 must be set odd before each entry at (4), since it is altered by R247. If no words are to be transferred, the program may be resumed in main control by setting the main program control, (6/201) in subsidiary store to (10/247) and entering R247 at (4).

4. The S.E.R. entered may be in fixed or main store. The S.E.R re-entry address is set on entry to cause re-entry via R247 in the event of a halt. On entry B96, 97 are altered; B97 is even

R248: Read or write isolated words

Purpose An S.E.R. providing a subroutine to enable other S.E.R. to read or write a single half word from or to any block of store used principally by routines manipulating input and output wells, where the first half word in each block constitutes the link to the next block on a system tape.

Registers of fixed store: 15

Instructions obeyed: Maximum of 15 plus one entrance to R318 and two to R312

Parameters used: (1) to (7)
(4) = Reserved block label used for temporary reference to block (digits 22-12). Rest zero
= *3667

Cross references:

(3)	=	(35/203)	Block location table
(5)	=	(2/318)	Alternative entry to Call to Cores
(6)	=	(1/318)	Call to cores
(7)	=	(1/312)	Set P.A.R.

Connections with other routines:

Entry at (1) if location of block in block directory is known, with
Word 0 of block location table = position relative to start of block directory, digits 12-2. Rest zero.
B108 = Program no. (20-14)
Location in BD relative to start of area for program (12-2). Digits 21, 0 irrelevant. Rest zero
B110 = Return address R. Digit 2 = 1 (read) 0 (write)
B104 = Address within block of half word required (digits 11-2)
B102 = Quantity to be transfered (if write)

Entry at (2) if only block label is known, with
B109 = block label (digits 22-12)
program no. (digits 8-2)
digits 9,0 irrelevant. Rest zero
B110 = Return address R. Digit 2 as above
B102, 104 as on entry (1)

Exit If block not in cores or drum queue full, return to address specified in S.E.R. re-entry address via R318. Resumption here is when the drum queue has a vacancy or the block is in cores. B100 or B100-104 preserved according to current S.E.R. base and re-entry address, which are unaffected by R248. When block in cores, exit to R with

B102 = Quantity read (read)
Unaltered (write)
B110 = Unaltered
B109 = Page no. of block digits 10-3. Rest zero.
B108 = *4
B105-107 altered.

Subroutines

"Call to cores" a) entered at (1/318) with
B110 = Return address if in cores
B109 = Block label (22-12) Prog. no. (8-2)
Digit 0 = 1 (operand)
Digit 9 = 1 (Don't change timer) Rest zero.

b) entered at (2/318) with block location table set up as on entry (1/248)
B110 = Return address if in cores
B108 = BD location relative to program start (12-2)
Prog. number (20-14)
Digit 0=1 (operand) digit 21=1 (Don't change timer)

Exit when block in cores with B105-110 altered,
B109 = Page number digits 10-3

"Set PAR" entered at (1/312) with
B108 = new PAR setting, digits 23-12. Rest zero
B109 = page number digits 10-3
B110 = return address
Exit with B109, 108 unaltered, B107 altered.

Temporary working space: As for R318, plus half word 2.0 of block location table. A temporary setting of a page address register is used, parameter (4).

Notes:

1. The block is called to core store if necessary and the page directory is set with p 21 = 1 (don't change timer). The relevant entry in the Block Timers is therefore unaffected.

2. If the block is also required to be locked down, this may be achieved by setting
Digit 22, B108 = 1 on entry (1)
or digit 10, B109 = 1 on entry (2)

3. If the block is not required further, the master routine should release it by setting the "write to drum" digit in the appropriate page timer.

4. This routine should be used sparingly, since on each entrance, a search of the block directory may be required in R318. If several words are required from the same block the user should set the PAR himself and read the remaining words directly, or use R318 directly.

R249: Special halt and free object program

Purpose: A subroutine to halt the current object program for a given reason, suspending program branching if in use, and later to free all object programs halted for the given reason. Used in rare cases when it is necessary to reserve a route in the supervisor for use by one object program at once.

Registers of fixed store: 15

Instructions obeyed: To halt, 6 plus entry to R204 at (14/204)
To free, 3 + 6P where there are P object programs, plus 4 for each program made free

Parameters used: (1) to (7)

Cross references:

(3)	=	(5/203)	Current program number
(4)	=	(9/204)	Program status directory
(5)	=	(9/207)	Leading prog. no. relative to (4)
(6)	=	(17/204)	Halt program
(7)	=	(9/214)	Free program

Connections with other routines:

Entry at 1) To halt current object program with
B107 = reason for halt, digits 20-12 rest zero
B106 = return address R
Exit to return address with B108-B110 altered and program halted in status directory, in supervisor.

Entry at 2) To free all programs halted for given reason with
B109 = reason for halt, digits 20-12, rest zero
B106 = return address R
Exit to return address with B105, 108, 110 altered,
B108 = 0

Subroutines:

a) "Halt main program"
Entered at (17/204) with
B110 = Current program no. (digits 8-2)
B108 = Return address (digits 23-3) Digit 0 = 1
(In Supervisor halt)
Return with B108, 109 B_t altered

b) "Free program"
Entered at (9/214) with
B108 = Program number (digits 8-2)
B110 = Return address
Return with B107 altered.

Temporary working space:

Entry 1) B108-B110 B_t
2) B105, 108, 110

Notes:

1. On entry (1), the program is halted in supervisor, and is resumed at the re-entry address with B100-104 preserved. The full recovery switch is unaltered. Program branching is suspended if in use - resumption will cause branching to be re-established.

2. On entry (2), all programs halted for the given reason are made free, and will be resumed In Supervisor in their order of priority in the program list; no action is taken if no programs are halted.

3. The status directory is set to: p23-21 all I's 20-12=n when the program is halted.

R250: Co-ordinate timed routines

Purpose: To control entry to timed routines, ensuring they are run in the correct sequence. An SER in main store entered to request running of the routine, and entered on conclusion of all such routines.

Registers of main store: 29

Instructions obeyed: Initial entry : Up to 12+2N where N such routines are permitted
Entry at end : Up to 11 + 8N

Parameters used: (1) to (9)

(3) = "Table of requests"
A table in the same block as R250, containing one word per timed routine. Sufficient entries must be allowed for to accommodate as many timed routines as can require simultaneous scanning by the clock routines.

(5) = No. of entries in table (digits 23-3)

Cross references:

(6) = (11/230) Timed exit criteria
(7) = (12/230) Timed exit address

Connections with other routine:

Entry at 1) To initiate a timed routine
B108 = Time at which routine to be entered
units 0.1 seconds, digits 22-1
B109 = Address of routine to be entered
B110 = Return address from R250
Exit to return address after recording entry in queue
B106, 107 B_t altered.

Entry at 2) To conclude a timed routine (and re-enter in queue if necessary)
Parameters as for entry 1) B109=0 if no new entry to be made.
Exit to return address with B107 to B109, B_t altered.

Temporary working space: Entry 1) : B106-107, B_t
Entry 2) : B107-109, B_t

Notes:

1. The available requests are presented one at a time for scanning by R230. The requests are supplied in time sequence. Each request is entered via R230 when the current time equals or exceeds the requested time; an SER is placed in the slow queue, eventually entered at the specified address with B100 = time at entry (0.1 seconds digits 22-1).

R250: continued

2. In order to prevent multiple entries to the SER queue, the time criterion is set by R230 to *376 when a routine is entered to the queue. This will only be reset after entry to R250 at (2) by the timed routine. The entry may simple to be notify the conclusion of the routine (B109=0 on entry) or may be, in addition, to request another timed entry (e.g. routines repeated at timed intervals).

   The timer should only freed in this manner when a position in the slow SER queue is available to record any subsequent timed routine; hence in general re-entry to R250 should be made only when the location is the queue used by the current routine is known to be available, which is either when it has transferred itself to another base in the same or other queues or when it had ended or can no longer be talked.

3. When no timed requests remain in the list held by R250, the time criterion is set to *375, a time never met if the timer is reset say every 24 hours.

R251: Analyse system input/output tapes

Purpose: An SER subroutine to find whether input/output tapes are present and if so if they are active, and also to terminate use of them. The routine deals with the special case of a combined input/output tape.

Registers of fixed store: 13

Instructions obeyed: 4 to 13 depending on conditions

Parameters used: (1) to (4)

Cross reference:

(4) = (15/229) Miscellaneous marker

Connections with other routines:

Entered at (1) to analyse tapes
B126 = even (output)
odd (input)
B110 = Return address

Exit to return address with Bt altered

B109 = 0 if tape present, busy
B109 non zero, even, if tape not present
B109 = 0.1 if tape present, not busy
In the latter case, the tape is recorded by this routine as busy.

Entered at (2) to end use of tapes, to record as not busy
B126 = even (output)
odd (input)
B110 = return address

Exit to return address with B_t altered, B109 = 0.1

Temporary working space: B109, B_t

Notes:

1. The state of the tapes is recorded in the miscellaneous marker half word as follows:

Digit 23	1 if tapes combined, 0 otherwise
22	0 if input busy, 1 not busy
21	0 if output busy, 1 not busy
1	1 if input present, 0 otherwise
0	1 if output present, 0 otherwise

2. If the two tapes are combined in one, digits 1, 0 both read 1, and digits 22, 21 are set and reset together by this routine.

R252: Link SER to output scheduler

Purpose: An SER in main store occupying the same block as R240 providing links to and from the central output routines when SER's are producing output to form part of the output well. The routine preserves and restores working registers so that the SER can be continued as part of the output scheduler, R253, which will occupy a separate block of main store.

Registers of main store: 17

Instructions obeyed: Entry (1) : 4 Entry (2) : 13

Parameters used: (1) to (10)

(3) = "Start of block - end", such that word 3.0 of the working area of the current peripheral = start of block + (3) (see R240)

Cross references:

(4)	=	(1/253)	Entry to "Step output from SER"
(5)	=	(5/202)	Program scan exit
(6)	=	(12/240)	Alternative entries to Supervisor output
(7)	=	(20/240)	
(8)	=	(2/592)	)
(9)	=	(3/592)	) Location of link words in block, relative
(10)	=	(4/592)	) to start of block

Connections with other routines:

Entry at (1) via R240 when block filled (word 1 of working area of peripheral = (1/253) - see R240).
with B100 = Address of working area of peripheral less *7

Exit to 6(12/240) with B105 preserved, and program scan exit set to 1/253. Exit thence to 1/253, setting re-entry address, with B100 - 104 preserved in store and B100 unaltered

Entry at (2) via R593 when new block supplied with
B100 unaltered (as on Entry 1)
Link words recorded in working area words 6, 6.4, 7.

Exit ot 5(20/240) to continue output with link words reset, new block in core store with same block label as old block, and B100-105 recovered. Exit thence to master routine after recording characters.

Temporary working space: B105, 106, 109

Notes:

1. It is essential that there be no halt for non equivalence as contol passes between R240 and R252. Halts as control passes to and from R253 are permissible.

2. R253 is continued on the same base and re-entry address as the master routine entering R240. The B registers are preserved in the peripheral working area, leaving the SER queues and dump address if any to accommodate halts for non-equivalence.

R253: Step output from SER

Purpose: An SER in main store entered via "Supervisor Output Routines" to pass to the output well a completed block of output formed by an SER. The routine supplements R593, which performs the same function for an output block from a main program. A new block is aquired and handed back to the originating SER with same block label as used originally.

Registers of main store: 47

Instructions obeyed: 31 on entry plus entrances to Locate Store Block and aquire one block
12 on exit with new block set up.

Parameters used: (1) to (21)

Cross references:

(3)	=	(25/593)	Reason for space request
(4)	=	(1/630)	Aquire block
(5)	=	(2/203)	Block directory
(6)	=	(3/203)	Block timers
(7)	=	(6/203)	Page directory
(8)	=	(2/592)	Locations of link parameters in block
(9)	=	(3/592)	
(10)	=	(4/592)	
(11)	=	(12/203)	Store location and lock out
(12)	=	(35/203)	Block location table
(13)	=	(1/312)	Set Page Address Registers
(14)	=	(37/593)	Entry to Step output
(15)	=	(2/252)	Re-entry to link routine
(16)	=	(5/201)	Re-entry address
(17)	=	(7/201)	SER base
(18)	=	(1/202)	Program Scan
(19)	=	(1/206)	Enter SER to queue
(20)	=	(5/203)	Current program number
(21)	=	(16/204)	Alternative entry to Halt Program

Connections with other routines:

Entry at (1) from R252 via program scan with
B100 = Working area of peripheral less *7
Re-entry address set to (1/253)
Exit to (37/593) to set up output conditions with
B102, 103, 104 = contents of link words (8) (9) (10)
B100 = working area of peripheral less *7 (note digit 23 = 1)
B96 = block label of new block, digits 22-12
Remainder irrelevant.

R253: continued

Re-entry at (2) from R593 with
Program scan exit set up as described for R593
B100 unaltered
B96, 103, 104 = new values of link words
Other B registers as described for R593 exit.

Exit a) If SER base = 0, to "Halt Main Program" at (16/204) with
B108 = Link (program scan) digits 23-3. Digits 0 = 1
B109 = 0.1
B110 = Current program number
Re-entry address set to (2/252)
Exit thence to program scan with current program set to resume in supervisor not halted

b) If SER base ≠ 0, to "Enter SER to Queue" at 1(1/206) with
B126 = odd
B107 = SER base, digits 2-1. Rest zero
(11 = slow, 01 = tape, 10 = top)
B108 = Working area of peripheral less *7
B110 = Program scan
Exit thence to program scan after inserting SER in queue.

Subroutines:

a) "Aquire block" entered at (1/630) with
B101 = Reason (output well)
B110 = Return address
Exit to re-entry or to return address with
B105 = Location of new block relevant to start of block directory.
Block directory = block label, digits 22-12
Rest ones.

b) "Store location and lock out" entered at (12/203) with
B108 = 0 (Supervisor block)
B109 = Block label (digits 22-12). Rest zero
B110 = Return address
Exit to return address with block location table = location of block relative to the start of the block directory, digits 12-2.

c) "Set Page Address Register" entered at (1/312) with
B108 = *4
B109 = Page number (digits 11-3)
B110 = Return address
Exit to return address with PAR set to *4, B107 altered.

Temporary working space: B101-110. B96 used but reset.

R253: continued

Notes:

1. It is essential that no halt occurs as control passes between this routine and R593, and hence they must occupy the same main store block.

2. On exit, the link words are recorded in words 6, 6.4, 7 of the working area of the peripheral. If the SER base is zero, the current main program is set up to resume in Supervisor at (2/252) with B100 perserved (= working area of peripheral less *7). If the base is non zero, an SER is inserted to the appropriate queue, to be entered at (2/252) with B100 preserved. Exit is then to program scan, which may jump via Program Scan exit to one of the exit routines listed in R593. By this technique the exit routines are entered with preservation of B100 - 105, as described, and the initiating routine is eventually resumed via R252.

3. The routine demands the simulatanecus presence in core store of the peripheral working area snd the block newly filled. It does not call the newly aquired block to core store.

R255: Fixed store output organisation

Purpose: A fixed store routine obeyed on extracode control, entered from extracodes Break output, Delete output, Rename output as input, and Define output stream. In the first three cases, the named output is selected as the current stream, and one new line character is output. Exit is to a supervisor routine in main store in all cases.

Registers of fixed store: 14

Instructions obeyed: 2 to 12

Parameters Used: (1) to (8)

Cross references:

(3)	=	(1/247)	Enter Supervisor setting full recovery
(5)	=		Main Store Define output
(6)	=	53*4	Extracode 1065 : Output layout
(7)	=	(99/900)	Extracode working space
(8)	=	48*4	Extracode 1060 : Select output
(4)	=	(1/264) - (5/255)	[1/264 = Main store break output]

Connections with other routines:

Entered at (1) from extracode jump table for extracodes

Break output :	Digits 2-0 of B126 = 0.4
Delete output :	" = 0.5
Define output as input:	" = 0.7

Exit to (1/264) In Supervisor with full recovery switch set and B96 digits 2-0 ≡ B126 digits 2-0 on entry Program set for return in extracode to start of extracode "Select output", previous current output stream recorded in word 4(99/900).

Entered at (2) from extracode jump table for extracodes "Define output"

Exit to (5) in main store, in supervisor with full recovery switch set.

Subroutines:

a) Extracodes 1060, 1065. These are entered with Main Control in fixed store, and exit in extracode control to the return address in B127

B119 = Output stream to be selected for 1060
= 2.1 (New line character) for 1065

R255: continued

b) "Enter Supervisor and set full recovery" entered at 1(1/247) with

B91 = Address of Supervisor routine S
B97 = Extracode address for return to main program

Exit to S in Supervisor with full recovery switch set,
B96 = S(digits 23-0).

Note:

In the case of extracodes Break, Delete, Define as input, the named stream is selected as current and a new line character is inserted to close the existing record. After dealing with the document in main store supervisor routines, control returns to select the original output stream; if this stream no longer exists (after deletion) output 0 is selected - this is under control of R264 in main store.

R260: Change program store allocation

Purpose: An SER in main store to implement extracode 1165 "Store allocation = n blocks". Blocks are aquired or released, and the block and page directions are compensated. An alternative entry provides a subroutine in supervisor for use by other supervisory routines.

Registers of main store: 89

Instructions obeyed: 13 if no change made in allocation
34 + 4N if allocation increased by N plus entry to R246 to shift block directory.
25 + (7 to 27) N if allocation reduced by N, plus entry to R246

Parameters used: (1) to (21)

Cross references:

(7)	= (5/203)	Current program number
(8)	= (4/203)	Program store directory
(9)	= (5/201)	Re-entry address
(10)	= (15/204)	Program switch directory
(11)	= (1/631)	Aquire blocks
(12)	= (1/246)	Move block directory up
(13)	= (2/203)	Block directory
(14)	= (3/203)	Block timers
(15)	= (4/247)	Reset full recovery switch and exit
(16)	=2(17/203)	Entry to Store Location (check S=7 and monitor) (see note 3)
(17)	= (21/203)	Entry to Store Location (halt for block) (see note 3)
(18)	= (6/203)	Page directory
(19)	= (12/227)	Record of directory of current input/output
(20)	= (2/246)	Move block directory down
(21)	=-3(23/205)	Amend free block counter

Connections with other routines:

Entry at (1) from extracode via (1/247) with
B91 even
B119 = required blocks (digits 22-3) Rest irrelevant

Alternative entry at 3(1) from supervisor with
B100 = return address, in supervisor
B101 = required blocks (digits 23-3). Rest zero
In both cases, entry is In Supervisor, with SER base = 0 and full recovery switch set for the current program.

Exit After entry for extracode, to (4/247) with B121 = 0 to reset full recovery switch and exit.
Otherwise to return address in B100 with re-entry set to this address.

R260: continued

Subroutines:

(a) "Aquire n blocks" entered at (1/631) with

B109 = no. of blocks n (digits 23-3)
Digits 2, 1, 0 = Priority: 0.1 Peripheral, top priority jobs
0.3 Tape jobs
0.4 Long, low priority jobs

B110 = Return address

Exit to return address with n free blocks in supervisor area and counter of free blocks reduced by n. B100 - 110, Bt altered.
If no space available, halts and subsequently returns to the re-entry address.

(b) "Move block directory" entered

(i) at (1/246) to aquire space (n blocks) with
B107 = -n + 0.4 (digits 23-2)
B105 = Location of last block in area of current program relative to the start of the block directory (this is the entry used by monitor routines), in digits 12-2. Rest zero
B110 = Return address
Exit to return address with vacant entries above the monitor block, all directories compensated appropriately.
B105 unaltered.
B106 = -n (digits 23-2)

(ii) at (2/246) to release space (n blocks), with
B106 = n (digits 12-2)
B105 = Location of first block to be released relative to start of block directory.
B110 = Return address
Exit to return address with entries removed and all directories compensated appropriately.

(c) "Store Location" entered at 2(17/203) to halt program for any block with lock out = 7, and to monitor if no such block is found.
B110 = Start of program area in block directory
B108 = Current program number
Exit either to halt for block, resume at re-entry address or to monitor "Store allocation exceeded".

(d) "Store location" entered at (21/203) to halt program In Supervisor for block b
B109 = block label b, digits 22-12. Rest irrelevant
B108 = current program number
Exit to halt for block, resume when block unlocked at re-entry address,
B100 to B104 recovered.

(e) "Amend count of free blocks" entered at -3(23/205) to step count by 1

B110 = Return address
Exit to return address with B105 - 110, Bt altered.

R260: continued

Temporary working space: B100 = 110 Bt

On exit to supervisor routine, after alternative entry at 3(1) B100 is unaltered B102 = Current program no. (digits 8-2) and the full recovery switch is left unaltered.

Notes:

1. If no change is required in store allocation, an immediate exit is made. Note that the last entry in the program area of the block directory is not included in the count.

2. If more store blocks are required, all entries in the block directory above the last entry for the program are shifted up and the new entries are recorded as vacated (digits 23=1, 0=0). The block timers are set to zero. The remaining blocks retain the same position in the directory relative to the start of the program area.

3. If less store blocks are required, entries above the last entry are moved into empty or vacated positions. If any such block is locked out, the program is halted for this block to avoid alterations to the drum and tape queues. If any block moved is in core store, the page directory is altered. If a block is the current input or output block, the records of these are altered appropriately.

   If insufficient vacancies exist, a check is made of any blocks for which lock out = 7. If any exist, the program is halted for the first such block and is resumed when the block is lost. If no such blocks exist, the program is monitored.

4. The record of number of blocks in the program store directory is altered to the new count, n.

R261: Compiler hand-over

Purpose: An SER in main store entered from extracode 1142 to end compiling phase. Depending on entry conditions and the state of the job description, a new compiler may be defined, the compiler and working space lost, store allocation reviewed, and the program entered under main control or in supervisor.

Registers of main store: 53

Instruction obeyed: 48 to 56 if no blocks to be lost
Additional 5 + (5 to 15 per block) if compiler to be lost.

Parameters used:

(1) to (23)
(10) = Reserved block limit = *34

Subsidiary store:

(4) = "Job location"
One half word in current program working area holding address of first part of processed job description in supervisor store.

(23) = "Program compiler record"
One half word in current program working area holding compiler position in compiler directories, digits 23-2, if compiler is to be returned to supervisor store, zero otherwise.

Cross references:

(5)	=	(99/900)	Extracode working space
(6)	=	(5/201)	Re-entry address
(7)	=	(5/203)	Current program number
(8)	=	(4/203)	Store directory
(9)	=	(2/203)	Block directory
(11)	=	(3/203)	Block timers
(12)	=	(21/203)	Store location to halt for block
(13)	=	6(14/317)	Lose block
(14)	=	(15/228)	"End program" extracode
(16)	=	(1/260)	Change program store allocation
(17)	=	(15/204)	Switch directory
(18)	=	(4/247)	Reset full recovery switch and exit
(19)	=		Define compiler
(20)	=	(8/227)	Monitor directory
(22)	=	(1/295)	Return compiler

Connections with other routines:

Entry At (1) from extracode via (1/247), In Supervisor, with full recovery switch set.

B119 = Entry address to program, digits 23-0
If digits 23=1, End Program

ba : Digits 0 = 1 Define compiler
0 Do not define compiler
Digits 23= 1 Do not lose compiler
= 0 Lose compiler, digits 22-12 = lowest block label to be lost

R261: continued

Exit a) If "define compiler", to (19) In Supervisor
Return to (1/261) with ba even, B119, B121,
ba (digits 23-1) unaltered.

b) If "return compiler" (see note 5) exit to (1/295)
In Supervisor with
B95 = Compiler number
Return to (1/261) with compiler number reset to zero

c) To (4/247) with "Parameter" in B91, B121 = 45.0
B127 set to entry address or address of End Program extracode
Exit thence to program after loading B90 and resetting full recovery switch

d) To enter program In Supervisor with full recovery switch set,
entry address as recorded in "Execution time" in Job description
(see note 2)

Alternative entry at (21) in Supervisor to Reset Store Allocation with

B100 = Return address. Digit 0 = 1
B102 = Store allocation, digits 23-13. Rest zero.
(excludes last block, used for processing)
Full recovery switch set

Exit to return address via 3(1/260) with directories compensated
alters B101 - 110, Bt

Subroutines:

a) "Lose block" entered at 6(14/317) with

B105 = Location of block relative to start of block
directory digits 12-2
B100 = Return address
Exit to return address with B101-105 unaltered

b) "Store location", entered at (21/203) to halt for block b with
B109 = block label, digits 22-12. Rest irrelevant.
B108 = Current program number, digits 8-2
Exit to halt program and eventually to re-entry
address with B 1-104 recovered

c) "Change program store allocation" entered at 3(1/260) with

B101 = new store allocation, digits 15-3. Rest zero
B100 = return address
Exit to return address with directories compensated and
B102 = Current program number

Temporary working space: B91-95, B100-110, Bt, words 0.0 0.4 of
extracode working space.

R261: continued

Notes:

1. Storage allocation is reset to the value recorded in digits 23-13 of word 2.4 of the second section of the job description.

2. Exit conditions are determined by word 5.0 of the second section of the job description. If digit 23=0, exit is to the address specified in B119 on entry (or to End Program) with B121 = 45, to load the "parameter" from the job description into B90 before entry.

   If digit 23=1 and digit 1=1, the master and local process switches are set before entry. Further, if digit 2=1, the "program" is entered In Supervisor at the address in digit 22-3 of this half word.

   These switches supply facilities to enter program under extracodes control, or to load extra part of supervisor routines.

3. The compiler switch is reset to zero before exit.

4. If digit 23 of ba is 0 on entry, any blocks with labels between b and *3377 inclusive are lost from the program area, where b is in digits 22-12 of ba. If any such block is locked out, the program is halted until it is made free.

5. If on entry the Program Compiler Record is non zero, the current compiler is returned to supervisor store and the entries lost from the program area by entry to (1/295). R261 is subsequently re-entered by repeating the original entry with the Program Compiler Record now reset to zero and the compiler lost. See R295 for further details. The Program Compiler is set originally (to zero or to the compiler number) by R290 "Call Compiler n".

R262: Extracode Read Parameters

Purpose: An SER is main store to implement the extracode 1140 "Read parameter B to store S". The parameters are read from the processed job description or from the block or tape directories.

Registers of main store: 56

Instructions obeyed: Varies markedly with the state of the job. 24 for simple parameters.

Parameters used: (1) to (19)

Cross references:

(9)	=	(99/900)	Extracode working space
(10)	=	(5/203)	Current program number
(11)	=	(4/203)	Store directory
(12)	=	(2/203)	Block directory
(13)	=	2727*4	Table *4, *2 etc.
(14)	=	(5/221)	Deck allocation directory
(15)	=	(4/261)	Job location
(17)	=	(4/247)	Reset full recovery switch and exit
(18)	=	(5/201)	Re-entry address
(19)	=	(3/221)	Number of tape decks

Connections with other routines:

Entry at (1) from extracode via (3/247), in supervisor with full recovery switch set.

B121	=	0.0	Read Job title
		0.4	Store required
		1.0	Computing time
		1.4	Execution time
		2.0	Parameter
		2.4	Tape labels
		3.0	Input streams
		3.4	Output streams
B119	=	S	Store location for results

Exit: To (4/247) to transfer results with

B92 = no. of half words less one, digits 23-2

B91 = 0 : Exit to main control

$\neq$ 0 : Re-enter R262 at address in B91

Extracode working space loaded with results from word 0 onwards.

Exit : Return to R247 or exit to main control after resetting full recovery switch.

Temporary working space: B91, 92, 95, 100-110, extracode working space words 0-4

R262: continued

Notes:

The parameters are supplied as follows:-

Job title: From words 0 - 9.4 of section 1 of processed job description to stores S onwards. End on half word with digits 5-0 = 0.

Parameters: Execution time, Computing time:
From half words 4.4. 5, 5.4 respectively of section 2 of processed jobs description, to half word S.

Store Allocation: From half word 2.4 of section 2 of job description, to half word S. Digits 23-13 = number of blocks. Rest irrelevant.

Tape Labels: From deck allocation directory (tapes for which digits 8-2 = program number, digits 1,0 = 0)
Eight half words are produced, digits 23-8 of half word n correspond to tape labels 16n to 16n + 15 respectively Digits read 1 if tape defined, 0 otherwise. Digit 7-0 are always zero in all half words.

Input, Output Stream labels: From block directory (blocks with labels in range *36 to *37, digit 12 = 0 for input, 1 for output, digits 17-13 = stream number)
Two half words are produced, digits 23-8 of half word n correspond to stream numbers 16n to 16n + 15. Digits read 1 if stream define zero otherwise. Digits 7-0 are always zero in both half words.

R269: Start output - Select document

Purpose: A supervisor routine in main store entered when an output equipment is to be started or restarted. If no document is currently selected, the first document assigned to the equipment is selected. If no document is assigned to the equipment, exit is to a routine to select a document assigned to another equipment; otherwise exit is to a routine to start or continue output of the selected document.

Register of Main Store: 69

Parameter Used: (1) to (24)

Cross reference:

(8)	= (40/599)	Stream record in peripheral private store
(9)	= (9/257)	Output list
(11)	= (2/270)	Start output of document
(12)	= 0.4(29/264)	Start of Complete list relative to (9)
(13)	= (15/229)	Miscellaneous marker
(14)	= (58/270)	Dump tape record
(15)	= (59/270)	Dump marker : recover output
(16)	= (60/270)	Start dump tape
(17)	= (13/230)	Enter SER to tape queue
(18)	= (9)(13/263)	Start of current list
(19)	= (1/267)	Entry to change assignment of document
(20)	= 0.1(1/270)	Continue output of document
(21)	=1(9)(10/257)	Stream directory : current document
(22)	= 0.4(21)	Stream directory : coupled document
(23)	= (15/271)	Stream directory : document number
(24)	= 1.4(9)	Document counter

Connections with other routines:

Entered at (1) with B100 = Private store of peripheral less *7
Re-entry address set (1/269).

Exit: a) If document already recorded as currently being output, to (1/270) with

B126 = 1 in digit 0
B102 = Stream number, digits 8-3
B101 = Contents of "Current record" for this stream
B100 = Private store of peripheral less *7

b) If document found assigned to this peripheral (in coupled, complete, or current lists) to (2/270) with

B100, 101, 102, as for exit (a)

c) If no document found assigned to this peripheral, to (1/267) with B100 = Private store of peripheral less *7, and re-entry address set to re-enter R267.

R269: continued

d) If no document found assigned to this peripheral and digits 21 or 20 of "state record" in private store set to one, exit is to operator requests to relink peripheral or remove from the system. B100 = Private store of peripheral less *7.

Subroutine:

"Enter SER to tape queue" entered at (13/230) with
B109 = Start of routine to initiate recovery of output document from dump tape
B110 = Return address
Return with B100-104, B108 unaltered.

Temporary working space: B101-110

Notes: The components of the output list are searched in the following order and the appropriate action taken.

1.1 Current document entry for this stream, (21). If non zero, exit to 1/270. If zero:-

2. Coupled document entry for this stream, (22)
If non zero, this address is planted in the current document entry, the coupled document entry remaining unaltered. If zero:-

3. The list of complete documents is scanned. If a document is found assigned to this stream which has not been dumped,

a) If the system output tape is in use, no further scan is made, since output tape routines transfer documents to the relevant coupled lists.

b) If the system output is not in use, the document entry is removed from the list of complete documents and its address inserted in the coupled document entry for this stream. Action is then as in (2).

4. If the only documents found in the complete list have been dumped, the system dump tape is activated to recall these documents. Then, and also if no documents are found assigned to this stream:-

5. The current list is scanned for documents assigned to this stream and recorded as able to be output (digit 0, word 0.4 of document entry set to 1). If one such is found, the document entry is recorded as being output (see (3) above) and its address is recorded in the current document entry for this stream, with digits 0,1 both one.

6. If no document is found in Scan (5), exit is made to R267 to choose a document assigned to another device.

When a new entry is inserted in the current document entry for this stream, the main document counter is recorded as the document number for this stream, in (23), and the main counter is then stepped by 8.0 (one in digit 6).

R270: Start output of document

Purpose: A Supervisor routine in main store to start output of a document. The routine outputs layout characters at the start of a document, with additional layout, depending on the device for output stream 0. The routine is also entered when continuing output of a document, in which case an immediate exit is made.

Registers of main store: 92

Parameters Used: (1) to (28

Cross references:

(9)	=	(9/257)	Output list
(10)	=	*3667	Temporary block label
(11)	=	(64/599)	Current transfer address in private store of peripheral
(12)	=	(8/269)	Stream record in private store of peripheral
(13)	=	(41/599)	Block link in private store of peripheral
(14)	=	(21/269)	Stream directory, current document
(15)	=	(5/201)	SER re-entry address
(16)	=	(5/272)	Subroutine: Locate current block
(17)	=	(1/509)	Find V store type
(18)	=	(1/318)	Call to cores
(19)	=	(1/312)	Set PAR
(20)	=	(1/276)	Start output (standard code and shift)
(21)	=	(1/271)	Output heading
(22)	=	(4/272)	Continue output of document
(26)	=	(50/270)	Reason for space (temporary block for switched document)
(27)	=	(1/630)	Aquire one block
(28)	=	(2/203)	Block directory

Connections with other routines

Entry at (1) when output already started
B100 = Private store of peripheral less *7
B101 = Contents of current document entry for this peripheral (digits 19 = 1 if restarted output of layout - see note 5).
B126 = digit 0 = 1

Exit a) If not restarting layout, to (22) to continue output

b) If restarting layout, as for entry at (2)

Entry at (2) to start output with
B100 = Private store of peripheral less *7

Exit to (20) to start output of layout with
B100 = Private store of peripheral less *7
B101 = Initial address of output, digits 23 - 2 (word 1 of first block document)
B110 = Return address on completion of ouput

R270: continued

Alternative entries

1) At (7)

For subroutine "Set up stream parameters"
B100 = Private store of peripheral less *7
B105 = Return address
Exit to return address with B100, 105 unaltered
B101 = Stream no. digit 8-3
B102 = Current document entry
B104 = B109 = V store type, digits 23-2

2) At (5)

For subroutine "Call current output block to core store"
B100 = Private store of peripheral less *7
B103 = Return address
Calls to core store the block specified in digits 22-12 of entry (64/599) of the peripheral and records the block label *7667 in the page address register. If the block is not already in core store resumption is at the re-entry address which must be set externally. Uses B105 - 110, Bt.
B109 = Page number on exit, digits 23-3. B108 = *7667

3) At-1(7)

For combined subroutine (1) and (2)
B100 = Private store of peripheral less *7
B103 = Return address
Combined effects of (1) and (2). Uses B105 - 110, Bt.

4) At (24)

To start output at word 1 of block
100 = Private store of peripheral less *7
B105 = Block label digits 22-12, remaining digits irrelevant
B110 = Return address after output is ended.
Return with B100 preserved.

Subroutine Used:

a) "Call to cores" entered at (1/318) with
B109 = Block label digits 22-12
Digits 9,0 = 1 (operand, do not change timer)
Remaining digits zero
B110 = Return address
Return to re-entry address if block not in core store
to return address when in core store with
B109 = Page number
Uses B105 - 110, Bt

b) "Set page address register" entered at (1/312) with
B108 = Setting of P.A.R. Digits 23-12
B109 = Page number
B110 = Return address
Exit to return address, B107 altered.

R270: continued

c) "Locate current output block" entered at (5/272) with
B100 = Private store of peripheral less *7
B104 = Return address. Digit 0 = 1
Exit to return address if next block of output is linked and in main store with
B109 = Block label
Block status directory of this block recorded in the block link in the peripheral private store, (41/599)

Exit to program scan to idle if next block not linked and in main store or if waiting to be recorded on output tape (digits 2-0 of Status directory =011). A subsequent return will be made to (1/269) to restart output. Uses B105-110, Bt.

Temporary Working Space: B101-110, Bt

Notes:

1. Output comprises

a) Layout character : Anelex 1 Paper throw to channe 1
Paper tape 12 new lines
Card punch 2 new lines

b) 36 Upper case characters, ignored on Anelex and Card Punch

c) If the document is output stream 0

Paper tape 3 New lines
Card Punch 1 New card
Otherwise carriage control character zero

2. A document is recorded as Output 0 if digit 1 of word 0.0 of the document entry is one. In this case, R270 also sets digit 18 of the current document entry to one.

3. When a document has been removed from a peripheral to which it was linked by a semi-remote linkage, the document entry is left with word 1.4 of the document entry zero. In this case, R270 aquires a block for temporary use preparatory to printing information, and records the block label in (64/599)

4. (64/599) is set up either from the block link or from the position record in word 1.4 of the document entry.

5. When starting output, a halt may occur due to non availability of the first block of output. When it becomes available, R270 is re-entered at (1) via R269 with digit 19 of the current document entry one. The digit is reset by R270 when the block is found to be available.

6. Words 1 to 7.4 of the initial block of output are used to record the layout information. Word 0 is used to record the starting address of the programmers information, which is word 8 + 63N where N is recorded in digits 2-0 of word 1 of the document entry. These digits will normally be zero, and the information starts at 8 within the block. Word 0.4 of the block, holding markers indicating start and end of documents, is left untouched.

R271: Output heading

Purpose: A Supervisor routine in main store to output document and peripheral number, date and time before output of a document, and also when output is restarted following a break.

Registers of main store: 114

Parameters used: (1) to (27); (56), (60) (64) (67)

Cross references:

(9)	=	(9/257)	Output list
(10)	=	(10/270)	Temporary block label
(11)	=	(66/599)	Failure record in peripheral private store
(12)	=	(64/599)	Current address in peripheral private store
(15)	=		Stream directory: Document number
(13)	=	0.4(15)	Stream directory: Equipment name
(14)	=	(8/270)	Type directory
(16)	=	(5/201)	SER re-entry address
(17)	=	(57/270)	Date
(18)	=	(6/229)	Time
(19)	=	(24/270)	Start output at word 1
(20)	=	(2/275)	Fault routine
(21)	=	(5/270)	Call current block to cores
(22)	=	(1/276)	Output in standard code
(23)	=	(1/272)	Output next block
(24)	=	(5/273)	Entry to End Document
(26)	=	(7/270)	Subroutine set up stream parameters
(27)	=	(2/276)	Output preserving code conversion
(56)	=	(56/599)	)
(60)	=	(60/599)	)
(64)	=	(66/599)	) Working registers in peripheral private store.
(67)	=	(67/599)	)

Connections with other routines:

Entry at (1) from peripheral supervisor after entry from R270
B100 = Private store of peripheral less *7

Exit a) To (1/272) via peripheral supervisor when first block of output is completed
B100 = Private store of peripheral less *7

b) If output is simply to record that a document has been moved to another stream, exit to (5/273) to End Document, via the peripheral supervisor. B100 as above.

Alternative entry for Subroutine "Binary decimal output"

Entry at (2) B108 = Half word to be converted
B110 = Return address
If odd, B108 converted to internal code representation of the octal digits.
If even, conversion is of four digit units, (digits 23-20, 19-16 etc.)
In either case, 6 digits are converted, and a "point" inserted after each pair except the last
B105 = Address in store for results (2 half words)
uses B105, B106, B107

R271: continued

Subroutines Used:

a) "Start output at word 1" entered at (24/270)
B100 = Private store of peripheral less *7
B105 = Block label of block, digits 22-12. Remainder irrelevant
B110 = Return address on completion of output
Return to return address with B100 preserved.

b) "Call current output block to core store" entered at (5/270)
B100 = Private store of peripheral less *7
B103 = Return address
Return with current block in cores and PAR set to *7667

c) "Output in standard code" entered at (1/276) with
B100 = Private store of peripheral less *7
B101 = Initial address for output
B110 = Return address on completion of output
Exit to return address on completion with B100 preserved

d) "Output preserving code conversion" entered at (2/276) with
B100 = Private store of peripheral less *7
B101 = Initial address for output
B103 = Code conversion parameter + *1 (negative)
B110 = Return address on completion of output
Exit to return address on completion with B100 preserved

e) "Set up stream parameters" entered at (7/270) with
B100 = Private store of peripheral less *7
B105 = Return address
Exit to return address with
B100 preserved
B101 = Stream number digits 8-3
B102 = Current document entry
B104 = 109 = V store type, digits 23-2

Temporary Working Space: B101-110, Bt.

Notes:

1. If on return from output of layout or heading a fault is indicated (66/599) non zero) exit is to (2/275) with B100 unaltered, B103 holding the fault marker.

2. The routine outputs the following line of information before the title and information proper

a) Document number : 3 pairs of octal digits separated by "point" and followed by solidus

b) Description of output peripheral : 4 characters giving the output peripheral in use. This is only printed if digit 5 of the type directory (8/270) is set to 1.

c) If the document is being Continued after paper low halt
Restarted after a fault
or has been Moved from another stream
C, R or M is printed followed by the original stream description
if none of these conditions apply, spaces are left.

d) Date and time are then printed as Day. Month. Year;
Hours. Minutes. Seconds. These are followed by 2 New Lines.

3. If a document has been moved from this peripheral and was linked to it by a"semi-remote" linkage, the heading
DOCUMENT MOVED TO (Stream description) is printed, followed by date and time, and the output of the document is ended.

R272: Step to next output block

Purpose: A Supervisor routine in main store entered from the peripheral supervisor when one block of output has been completed. The routine up dates records and if necessary initiates routines to replenish the output well. If the document is not ended the next block is obtained and output; if ended, a routine is entered to print terminating information.

Registers of Main Store: 100

Parameters Used: (1) to (40)

Cross references:

(12)	=	(66/599)	Fault record in peripheral private store
(13)	=	(8/269)	Stream record in peripheral private store
(14)	=	(64/599)	Current address in peripheral private store
(15)	=	(41/599)	Block link word in peripheral private store
(16)	=	(13/258)	State of peripheral in peripheral private store.
(17)	=		Stream directory : Wells
(18)	=	0.4(17)	Stream directory : Counter
(19)	=	(15/299)	Miscellaneous marker
(20)	=	(5/201)	Re-entry address
(21)	=	(6/203)	Page directory
(22)	=	(3/203)	Block status directory
(23)	=	(12/203)	Store location and lockout, alternative entry
(24)	=	(1/202)	Program scan
(25)	=	(13/317)	Lose block alternative entry
(26)	=	-3(23/205)	Amend free block counter
(27)	=	(1/659)	Activate scheduler
(28)	=	(1/251)	Analyse system tapes
(29)	=	(13/230)	SER to tape queue
(30)	=	(2/248)	Read one half word
(31)	=	(22/271)	Start output (standard code)
(32)	=	(1/273)	End document
(33)	=	(31/499)	Start output tape
(34)	=	(38/264)	Idle bit in state of peripheral
(39)	=	(2/275)	Fault routine
(40)	=	2(6/273)	Aquire next document

Other parameters

(35)	=	0.4	Emergency bit in scheduler key word
(36)	=	8.0	Semi-emergency bit in scheduler key word

Connections with other routines:

Entry at (1) on completion of block of output, with
B100 = Private address of peripheral less *7

R272: continued

Exit a) If fault (66,'599) non zero), exit to (2/275) with

B100 unaltered
B103 = fault marker

b) If end of document (final transfer address not 511 or digit 16, word 0.4 of block = 1), exit to (1/273) with

B100 unaltered

c) If next block not available in store (or in store awaiting recording on system output tape) exit to program scan. Subsequently re-entry is at 1(4) when the block is made available (see note 3)

d) If next block is available, to (22/271) to start output with

B100 = Private address of peripheral less *7
B101 = Initial address for output : word 8 of block
B110 = Return address on completion of output (=1/272)

e) If output has been deleted by the object program, to 2(6/273) to find next document with

B100 = Private address of peripheral less *7

Alternative entries:

a) At 1(4) when peripheral restarted from an idling condition within a document, with

B100 = Private address of peripheral less *7
Exit as c) d) or e) above

b) At (5) for subroutine "Locate next block"

B100 = Private address of peripheral less *7
B104 = Return address Digits 0 = 1

If (41/599) is negative, exits to program scan.
Otherwise locates block with label in digits 22-12 of block link
If this has lock out digits S=3, exits to program scan.
Otherwise records, in (41/599) the block status directory of this block and exits to the return address with B109, digits 22-12, = block label of this block. B100 is unaltered.

c) At (11) for subroutine "Lose current block"

B100 = Private address of peripheral less *7
B104 = Return address

Exit to return address, after losing supervisor block with label in digits 22-12 of (64/599) and increasing counter of free blocks by 1 uses B102, B105-110, Bt.

d) At (8) for subroutine "Lose current block and start output"

B100 = Private address of peripheral less *7
B109 = Block label of block to be output, digits 22-12. Rest zero.

Loses current block as for entry (11) and then starts output of new block at word 8.0. Exits to program scan after starting output; returns on completion of output to (1/272)

R272: continued

Subroutines Used:

a) "Lose block" entered at 1(13/317) with
B108 = 0 (Program number)
B109 = Block label digits 22-12 Rest zero
B100 = Return address
Exit to return address with B102, 104 unaltered

b) "Amend free block count" entered at -3(23/205) with
B110 = Return address
Exit to return address with B100-104 unaltered.

c) "Activate scheduler" entered at (1/659) with
B109 = Reason for activation
B110 = Return address
Exit to return address with scheduler entered in SER queue unless already activated. B100-105 unaltered.

d) "Analyse system tapes" entered at (1/251) to analyse output tape
B110 = Return address, digits 23-3 Digits 2-0 zero
Exit to return address with
B109 even if output tape busy
odd if idle
(the routine is not entered if no output tape is in use)

e) "Enter SER to tape queue" entered at (13/230) with
B110 = Return address
B109 = Address of SER
B108 = Information for SER
Exit to return address after planting SER to the tape SER queue with
B106, 107 altered

f) "Read half word" entered at (2/248) with
B110 = Return address. Digits 2=1 (read)
B109 = Block label digits 22-12. Rest zero
B104 = 0.4 (word 0.4 in block)
Exit to re-entry address if block not in core store with
B100 preserved
To return address when block in core store, with
B102 = contents of word 0.4 of block. B100, 101 unaltered

Temporary working space: B101-110, Bt

Notes:

1. On completion of a block of output, the following counters are amended

a) Count of no. of blocks in document in stream directory (18). One added in digit 0 position.

b) Accumulated counter of blocks output on this peripheral. One added in digit 9 position of (8/269)

R272: continued

c) "Actual" back log: One subtracted in digit 3 position of (13/258)

d) "Print well back log": If system output tape is in use, one subtracted in digit 15 position of (13/258) unless back log already zero (i.e. digits 19-15 zero).

2. The following actions are taken to replenish the output supply

a) Activation of scheduler to initiate new jobs
No action is taken if digits 2 or 0 of (13/258) are one. If both are zero, the actual back log (digits 11-3 of 13/258) is compared with the emergency level, digits 8-3 of (17). If below this level, digit 0 of (13/258) is set to one and the scheduler is activated for Emergency. If the actual back log is above this level, it is compared with the semi-emergency level, digits 21-23, 0-2 of (17) If below this, the scheduler is activated for Semi-emergency, and digit 1 of (13/258) is set to one.

b) If the system output tape is in use, the print well level, if non-zero, is reduced by one and compared with the minimum print well, digits 14-9 of (17). If below this level, the system output tape is activated unless it is already busy.

3. The following conditions cause the peripheral to idle when the next block is not available.

a) Link word (41/599) negative:- A break occurs in the output stream in main store.

b) Link word has digits 22-12 = 0 (no block recorded as linked) and block status directory of current block has digits 22-12 = 0: No new block is linked to the current block.

c) Linked block has lock out bits = 3: Block awaiting transfer to output tape.

4. If the peripheral idles, the idle bit in (13/258) is set to one.

5. The current block is not lost until the next block is present in store.

6. If the current output has been deleted, digit 0 of (8/269) is one. It is cleared to zero. Normally the next document is selected before losing the current block. If entered from R270 to locate the current block, the current block is lost before selecting the next document.

R273: End output of document

Purpose: A Supervisor routine in main store entered when output of a document is complete. Closing information is printed at the end of the output, and the document is removed from the output list. If the paper low digit is present in the V-store a message is output on the operators output before selection of the next document; otherwise a routine is entered to select a new document.

Registers of main store: 89

Parameters used: (1) to (29)

(10) = *3667 Reserved block label for temporary use

Cross-references:

(9)	= (9/257)	Output list
(11)	= (8/269)	Stream record in private store of peripheral
(12)	= (64/599)	Current address in private store of peripheral
(13)	= (66/599)	Fault record in private store of peripheral
(14)	= (68/599)	V address in private store of peripheral
(15)	= (18/272)	Stream directory : Local counter
(16)	= (21/269)	Stream directory : Current document
(17)	= (22/269)	Stream directory : Start of coupled stream
(18)	= (5/201)	SER re-entry address
(19)	= (15/229)	Miscellaneous marker
(20)	= (1/312)	Set page address register
(21)	= (20/270)	Start output, standard code and shift
(22)	= (2/275)	Fault routine
(23)	= (11/272)	Lose current output block
(24)	= (1/269)	Find next document
(25)	= (3/275)	Print subroutine
(26)	= (23/275)	Message "Reload equipment"
(27)	=0.4(29/264)	Address of start of complete list relative to output list

Connections with other routines

Entered at (1) from R272 when output of a document is ended
B100 = Private address of peripheral less *7

Exit to (1/269) with B100 unaltered to select next document

Alternative entry at 2(6) from R272 to select next document
B100 = Private address of peripheral less *7
B101 = Stream number, digits 8-3

Exit to (1/269) as above

Subroutine Used:

a) "Set Page Address Register" entered at (1/312) with
B108 = Setting of register, Digits 23-12
B109 = Page number, digits 23-3
B110 = Return address
Exit to return address with B107 altered.

R273: continued

b) "Start output" entered at (20/270) to output in standard code, any shift.
B100 = Private address of peripheral less *7
B101 = Address of start of output
B110 = Return address on completion of output
Exit to return address on completion of output with B100 preserved.

c) "Fault routine" entered at (2/275) if output of closing information has faulted.
B100 = Private address of peripheral less *7
B103 = Fault marker, as in (51/599)

d) "Lose current output block" entered at (11/272) with
B100 = Private address of peripheral less *7
B104 = Return address
Exit to return address with B100 unaltered

e) "Print message" entered at (3/275) if paper low warning is present
B100 = Private address of peripheral less *7
B103 = Address of message
B104 = Return address after sending message
Exit to return address with B100 unaltered.

Temporary working space: B101 - 110, Bt

Notes:

1. The following message is output at the end of the document

"END OUTPUT (n) BLOCKS TAPE (a) (b)/0"

If no system output tape is in use, the tape section is omitted.

2. The tape name (a) is held in address 4.4 (28) and is assumed to contain up to four internal characters. Both (n) and (b) are printed as up to 4 decimal digits.

3. The current document is removed from the output list linkage, either from the complete list or the coupled list depending on digit 0 of the current document entry for this peripheral (1 = complete, 0 = coupled). The current document entry is reset to zero.

4. Paper low condition is detected by examining digit 2 of the appropriate V store line.

R274: Form restart address

Purpose: A supervisor routine in main store entered following detection of a fault on an output device. Computes a suitable restart address within the current output block and enters R270 to output a heading and resume output on the same device

Registers of main store: 31

Parameters Used: (1) to (5) ; (10) to (12)
(5) = Maximum store length for backspace (see note (2)).

Cross References:

(3)	=	-1(7/270)	Call current output to cores and set parameters
(4)	=	(4/270)	Alternative entry to print heading
(10)	=	(10/270)	Reserved block label
(11)	=	(64/599)	Current transfer address in private store of peripheral
(12)	=	(52/599)	Code conversion parameter in private store of peripheral

Connections with other routines:

Entered at (1) from R275 after printing message with
B100 = Private address of peripheral less *7

Exit to (4/270) to output heading with
B100 = Private address of peripheral less *7
B109 = Current transfer address for restart if at start of record within a block (positive)
or = Code conversion parameter + *1 if restart is at partial record start at beginning of block(negative)

Alternative exit to (4/270) to resume output of heading with B100 = Private address of peripheral less *7
B109 = Initial transfer address
This exit is taken when a fault has occured when printing a heading from words 1 to 7.4 of the current output block.

Subroutine Used:

"Call current output to cores and set parameters" entered at -1(7/270)
with B100 = Private address of peripheral less *7
B103 = Return address

Exit to re-entry address if block not in core store.
Otherwise to return address with block in core store with reserved block label (10/270) set in the page address register
B101 = Stream number
B102 = Current document entry
B104 = V store type

R274: continued

Temporary Working Space: B103-110, Bt

Notes:

1. If the transfer address is less than word 8 of the block, the heading was being output; the output is repeated from the start.

2. Otherwise the start of the document within the block is computed, which may be at word 8+63N depending on digits set in word 0.4 of the block. The document is scanned from here until a start of record is found in address S such that if the record ends in a location R, T-R is less than or equal to (5). Output is resumed at the start of record S. If no such position S is found, the fault occurred during output of the first record and if this starts at the beginning of the block, it may be a partial record. If this case, output is recommenced here but the existing shifts as recorded in (52/599) are preserved for later use in starting.

R275: Print message for output peripherals

Purpose: A supervisor routine in main store entered when a fault condition is detected on any output peripheral. Prints a message on the chief operators output and exits to routines to form a restart point or to resume output of the current document.

Registers of Main Store: 41

Main store tables and working space

Dump area : 5 consecutive half words used as dump for B lines during printing
Message locations : Consecutive half words containing starting addresses of message for faults
Tape low message: Start of message "Tape low"

Parameters Used: (1) to (22)

(14) = Dump area
(6) = Tape low message
(7) = Message locations
(5) = Fault marker for Tape low

Cross-references:

(4)	=	(1/274)	Form restart addresses
(8)	=	(8/269)	Stream record: private store of peripheral
(9)	=	(21/269)	Stream directory: Current document
(10)	=	(13/271)	Stream directory: External name
(15)	=	(1/220)	Reserve output
(16)	=	(2/220)	Free output
(17)	=	(1/240)	Print message
(18)	=	(3/240)	Print layout
(19)	=	(12/213)	SER dump address
(20)	=	(5/201)	SER re-entry address
(22)	=	(4/270)	Alternative entry to Start Output
(12)	=	(1/504)	Free peripheral
(11)	=	(52/599)	Code conversion parameter in private store of peripheral

Connections with other routines:

Entered at (1) for any fault with
B100 = Private address of peripheral less *7
B103 = Fault marker

Alternative entry at (2) for any fault requiring restart (see notes, 1,2) same entry parameters.

Exit : Unless tape low fault, to (1/274) with B100 preserved.
If tape low fault, to 1(4/270) with B100 preserved.

R275: continued

Subroutines Used:

a) "Reserve operators output" entered at (1/220) with
B101 = 0 (Chief operators output)
B110 = Return address
Exit to re-entry address if output busy; to return address otherwise with
B100 = Working address of operators output less *7

b) "Free operators output" entered at (2/220) with
B101 = 0
B110 = Return address
Exit to return address, B100-104 unaltered

c) "Output message" entered at (1/240) with
B110 = Working store of output less *7
B109 = Start of message
B108 = No. of characters less 1, digits 23-3
or 0.1 if message ends with characters zero
B110 = Return address Digits 0 = 1 for preservation of B100 - 104
Exit to return address with re-entry address reset, B100-104 unaltered.

d) "Output layout" entered at (3/240) with
B100 = Working store of output less *7
B109 = Layout character, digits 5-0
B110 = Return address. Digits 0 = 1 for preservation of B100-104
Exit to return address with re-entry address reset and B100-104 unaltered.

Temporary Working Space: B101-110

Notes:

1. In the case of a fault condition on output, the routine sets digit 20 of the current document record to 1, prints the message and exits to (1/274) to compute a restart point

2. When entered following detection of a programmed break for low paper, with "fault marker" (5), one is forced to digit 24 of the current document record and exit is to 1(4/270) to resume output at the current address.

3. The re-entry address must be set suitably before entry to this routine to allow for a halt if the operators output is otherwise engaged.

4. The standard output messages must be terminated by character zero (octal 00) and should include final spacing. Each is followed by the peripheral name, of up to 4 characters..

5. After the peripheral is made free, the code conversion parameter is restored to (52/599) in the private store, for access by R274.

R276: Select output code

Purpose: A supervisor routine in main store entered to select the code for output on an output device and to enter the peripheral supervisor to start output.

Registers of Main Store: $8 + \frac{1}{2}N$ where N = maximum no. of output peripherals

Instructions obeyed: 5 on entry (1), 2 on entry (2)

Parameters Used: (1) to (6)
(3) = Code conversion table (see note (1))

Cross reference:

(4)	=	(8/269)	Stream record in private store of peripheral
(5)	=	(7/503)	Entrance to peripheral supervisor
(6)	=	(6/502)	Entrance to peripheral supervisor

Connections with other routines:

Entry at (1) to output in standard code, perserving any existing shifts.
B100 = Private address of peripheral less *7
B101 = Initial transfer address
B110 = Return address on completion of output

Exit to peripheral supervisor at (7/503) with
B100, 101, 110 unaltered
B103 = Code conversion parameter

Entry at (2) to output with stated shifts, with
B100, 101, 110 as on entry (1)
B103 = Code conversion parameter plus *1

Exit to peripheral supervisor at (6/503) with
B100, 101, 110 unaltered
B103 decreased by *1, bits 0, 1 = 0

Temporary working space: B103

Notes:

1. The routine includes at (3) a table in consecutive half words, entry n being the code conversion parameter for output stream n. n is recorded in entry (8/269) of the private store of the peripheral in digits 8-3. The table (3) can be altered at will to provide different code conversion on different equipments.

2. The present maximum number of peripherals has been set at 16, giving a total length of 16 for routine 276. The routine occupies the same block in main store as R270-273.

R281: Set up program time checks

Purpose: A routine is main control to form in B registers the time checks to be applied to the object program from estimates supplied in the Job Description

Registers of main store: 15

Instructions obeyed: 15 plus extracodes 1312, 1314, 1343 twice, 1157 twice

Parameters used: (1) to (7)

Cross references:

(2)	=	(1/282)	Collect parameters
(3)	=	(1/201)	Enter Supervisor
(6)	=	(1/284)	Set up output
(7)	=	(1/283)	Set up tapes

Connections with other routines:

Entered at (1) under main control from R280

Exit to (1/283) in supervisor with subsequent return to
(1/284) in extracode control
B87, 90, 88 = Final contents of (9/227) Wait time
(21/303) Check time
(8/704) Overall time

Subroutine:

"Collect parameters" entered at (1/282) in supervisor extracode

Exit with B90 = Computing time estimate (secs) Digits 23-0
B87 = Execution time estimate (secs) Digits 23-0
via (1/202) to resume R281 in main control.

Temporary working space: B87 - 90, B96

Notes:

1. This routine occupies block *3403 in main store, and is copied to the object program from Supervisor store.

R282: Collect parameters

Purpose: A supervisor routine in main store entered from R281 to collect timing and output parameters from the job description and plant them in B lines or in program block *3403.

Registers of main store: 24

Instructions obeyed: 44 + 11 n where n output streams are defined.

Parameters Used: (1), (4), (6) to (12), (15)

Main Store Table:

(7) = Output list 16 half word in block *3403 set up by R282 and used by R284, 285

Cross-references

(8)	=	(4/261)	Start of Section 1 of Job Description (temporary store)
(11)	=	(5/201)	SER Re-entry address
(12)	=	(1/202)	Program scan

Other parameters:

(9)	=	5.4	Location of Computing Time in Section 2 of Job Description
(10)	=	5.0	Location of Execution Time in Section 2 of Job Description
(15)	=	2.0	Parameter in output list for "ANY, 1 BLOCK".

Connections with other Routines

Entered at (1) In Supervisor from R281, and (4/261) set to start of Section 1 of Job Description, and full recovery switch set.

Exit to program scan to enter R282 with
B87 = Execution time (secs) Digits 23-0
B90 = Computing time (secs) Digits 23-0
Output list entry n zero if no output n defined (see note)
Otherwise entry n copied from entry in Job Description

Temporary working space: B87, B90-92, B101-104

Notes:

If output 0 is not defined in the Job Description, it is set to be "ANY" type, length one block, as defined by parameter (15)

R284: Main Establish Output

Purpose: A supervisor routine obeyed under normal extracode control (not "in supervisor") to organise establishing of the output streams requested in the Job Description.

Registers of Main Store: 18

Instructions obeyed: 68 + 10 n where n output streams are defined.

Parameters used: (1) to (10)
(9) = *3417 Reserved working block label

Cross references:

(3)	=	(1/247)	Enter Supervisor
(5)	=	(1/236)	Exit from processing
(6)	=	(19/285)	Entry to Supervisor establish output
(7)	=	(7/282)	Output list
(8)	=	(3/285)	Entry to Supervisor establishment output
(10)	=	(1/286)	Set up input

Connections with other routines:

Entry at (1) in extracode control from R283 via Program scan
B92 = 0

Exit to R286 at (1) in Supervisor to set up input streams, with full recovery switch set.

Alternative entry at (1) in extracode control for extracode "Define output stream" with process switch set, B92 = output stream number (digits 5-2). Digit 0 = 1.
Exit to (1/236) with B126 odd to reset the process switch and re-enter the object program.

Subroutines:

a) "Enter Supervisor and set full recovery switch"
Entered at 1(1/247) with
B91 = Entry address In Supervisor
B97 = Exit address in extracode

Exit to Supervisor routine at B91, and when this exits, to B97 in extracode control.

b) "Supervisor establish output"
Entered at (3/285) in supervisor with
B119: Stream number. Digits 8-3, remaining digits zero
B92: Stream number. Digits 7-2, remaining digits irrelevant
B91: Digits 2, 1,0 = 100 (Select, output)

Exit to 2(1/590) and thence to program scan. Uses only B100-110, Bt.

Entered at (19/285) in supervisor to establish document with
B92 = Output stream number digits 7-2
B93 = Entry from output list for current stream
Exit to program scan via (1/259) with B92 unaltered.

R284: continued

Temporary working space: B91-94, 97, 119

Notes:

1. The routine creates a new block by referring to block *3417; this is used by R285 as the first block of the current output stream.

2. On exit after establishing all output streams, stream 0 is selected as the current stream, by use of extracode 1060.

3. The streams are set up in order 0, 1, 2 etc.

R285: Supervisor establish output

Purpose: A supervisor extracode routine in main store used as a subroutine by R284 to establish an output stream. One entry sets up the block label of the initial block and selects the relevant output stream. Another entry sets up the document in the output list and records the heading at the start of the block.

Registers of main store: 102

Instructions obeyed: Entry at (3) : 14 plus entries to R203, R590
Entry at (19) : 60 normally, plus entries to R257 R590, R259

Parameters Used: (1) to (24)

(8) =	(14/590)	Reserved label used for output
(6) =	511.4(8)	Standard end address within block
(12) =		Table of V-store addresses (see note 5)
(15) =		Routine "Set up private output"
(20) =		Negative of type for internal document digits 23-20 (see note 3)
(23) =		Parameter for "Internal document" (see note 3)
(24) =		Routine "Establishment private output"

Cross references:

(4) =	(9/284)	Reserved block label for temporary working
(5) =	(1/257)	Establishment output document
(9) =	(3/592)	Location in block of check length
(10) =	(4/592)	Location in block of peripheral number
(13) =	(2/283)	Job link word
(14) =	(1/259)	Write heading
	(1/203)	Store location and lock out
	(2/203)	Block directory
	(35/203)	Block location table
	(1/590)	Call output to Cores
	(15/229)	Miscellaneous marker
	(5/201)	SER re-entry address
	(2/590)	Set output PAR
	(73/596)	Current output: Record counter
	(74/596)	V-store address
	(75/596)	Current address
	(76/596)	End of store
	(72/596)	Current address
	(1/202)	Program scan

Connections with other routines:

Entered at (3) from R284 with
B92 = Stream number, digits 7-2, remaining digits irrelevant
B91 = Digits 2, 1 0 = 100 (Select output)
B119 = Stream number, digits 8-3, remaining digits zero

Exit to program scan via -2(2/590) : Jump to -2(2/590) with
B96 = Digits 2-1 of B91 on entry, digit 1 = 1

Exit to program scan with new output stream selected.

R285: continued

Entered at (19) from R284 to establish output document is central output list
B93 = Entry from output list of current program
Exit to program scan with B92 unaltered (stream number)

Subroutines:

a) "Store location and lock out" entered at (1/203) with
B109 = Block label (digits 22-12). Remainder zero
B110 = Return address
Exit to return address with block location table set to the location of the block in the block directory relative to the start of the block directory.

b) "Select input output", entered at -2(2/590) in Supervisor with
B96 = 0.5 (Select output)
B119 = Stream number (digits 23-3)
Exit to program scan when stream selected, B100-110 altered.

c) "Call output to cores", entered at 1(2/590) with
B100 = 0.4
B101 = Return address
Exit to re-entry address if block not in cores
Exit to return address when in core store, with PAR set up to *7666.

d) "Establish output" entered at (1/257) with
B107 Digits 23-22 Remote bits
8-3 Preferred stream number if any
2 1 if Priority, 0 otherwise
1 1 if Output 0, 0 otherwise
0 1 if Private, 0 otherwise
B108 Digits 23-20 Type
11-3 Length
0 1 if direct, 0 if indirect
B104 Return address
Exit to return address with B102 = Stream assigned to digits 8-3
B104-110 altered

c) "Write heading" entered at (1/259) with
B103 = Return address
Exit to return address with B100-110 altered.

Temporary working space: B91, 93-97, Bt

Notes:

1. On entry at (3), the block directory entry of block (4) of the current program is set to hold block label *36 a b
where a = stream number, digits 17-13
b = 0 for input, 1 for output, digit 12

R285: continued

2. The job link word is assumed to be set up by R283 to hold the the following (throughout, the main input stream refers to the input stream of the job description).

Digits 23-20 Type of output linked to main input stream
Digits 17-16 Type of linkage between input and output
Digits 11 1 if "direct" (i.e. short job), 0 if long
Digits 10 1 if priority, 0 if no priority
Digits 8-3 Stream number of output stream linked to main input stream
Digit 1 1 if the coupled output stream is not in the "ANY" category
0 if it is in the "ANY" category
Digit 0 1 if satallite coupling of streams, 0 otherwise
If no coupling exists between the main input stream and any output stream, digits 17, 16, 8-3, and 0 are zero

3. The type of output is transferred from the program output list, digits 3-0 to R257 in B108, digits 23-20.
If this type is an internal document, R257 is entered with

B107 = 0.1
B108 = (23)

This condition is detected by the type as described in the output list being the negative of (20)

4. If the entry in the output list has digits 23 = 1, the output is classed as "private" and an appropriate routine is entered.

5. Words (72) to (76) of R596 in subsidiary store are set up to their appropriate initial conditions. The V-store address is selected from a table of half words (12), entry n corresponding to type n as described in the output list.

6. If the output is classed as satellite, the heading is preceded by a record of four characters, of which the first character contains
Bits 5-2 Type of output requested for this document as for entry in output list
Bit 1-0 Zero
Thus a record is preserved of the intended destination of this output.

R286: Scan input list

Purpose: A supervisor routine in main store which selects input document entries from the processed job description and copies them into the main program area. For each stream, a processing routine is entered, and finally a routine is entered to call the relevant compiler.

Register of Main Store: 32

Instruction obeyed: About 100 per input stream defined

Parameters used: (1) to (8)

Main store working space: (6) = Document description
16 words in object program block *3403
to hold document description

Cross references:

(7)	=	(1/287)	Main establish input stream
(8)	=	(1/289)	Enter Call Compiler
		(4/261)	Job location record
		(5/201)	SER re-entry address
		(6/201)	Main program controls
		(1/202)	Program scan

Connections with other routines

Entered at (1) in supervisor, from R284

Exit: (a) To R287 to Main establish input stream
Document entry copied to (6). Entered at (1/287) under main control
Ultimate return to (3/286) with B99 unaltered to select next input document

(b) To R289 under main control when all documents established, with B80 = Description of Stream 0 (copied from word 10 of processed job description entry of this document).

Temporary working space: Document description area (6)
B99 - 101

Notes: Word 10.4 of the first section of the Job Description is assumed to hold the link to the first input document, and word 10.4 of each document entry to hold the link to the next entry. The last document is recognised as having a zero link or a link with digit 1 set to 1. It is assumed that word 10, digits 3-0 of the document entry holds the input stream number.

R287: Main Establish Input tream

Purpose: A Supervisor routine in main store obeyed under main control to set up an input stream, and to "Read" past titles and job description to the start of information.

Register of main store: 111

Instructions obyed: Mainly extracodes 1054, 1057 to read titles and Job Description - dependent on length of job description

Parameters Used: (1) to (25)

Main store working space:

(15) = One word for record of compiler name, set up by R287

Cross references:

(17)	=	(6/286)	Dump for document entry
(18)	=	(1/247)	Enter supervisor with full recovery
(19)	=		Set up private stream
(22)	=	(3/285)	Link for Create Input
(23)	=	(1/288)	Entry to call block from input well
(24)	=	(1/284)	Entry to create input/output

Connections with other routines:

Entry at (1) under main control from R286 with input stream description copied to the dump in the object program block *3403

Exit to 4(1/284) in extracode control to set up a new block and select the input stream with
B91 = (3/285) digits 2, 1, 0 = 0 Link and select input.
B97 = 0.1 : Return to main control
B90 = Word 10 of document description

Re-entry with B90 unaltered

Exit to (1/288) In Supervisor with full recovery switch set to move first block from input well with
B97 = 0.1 : Return to main control

Re-entry at (2) in main control with
B90 = Return address in extracode control (digits 23-3)
Digit 0 = 0 if no compiler name required
1 if compiler name to be assembled
B88 : Digit 23 = 0
Digit 0 = 0 Do not check title
1 Check title

Exit to return address in extracode control with input stream set up to read next character beyond title or job description.

R287: continued

Subroutines:

a) "Enter Supervisor with full recovery", entered at 1(1/247) with

B91 = Entry address
B97 = 0.1 (Return to main control)

b) Uses extracodes 1051, 1057, 1101, 1157, 1342, 1117

Temporary working space: B81, B83-90

Notes:

1. The input stream is read and analysed in order to pass over title, heading, and job descriptions. If required, the compiler name is copied to working store, (25), after the usual compression (ommission of shifts etc). Also if required the title is compared with that in the document description; if incorrect, the message TITLE ERROR INPUT (n) is output on the selected output stream, output 0, and the job is ended.

2. As a temporary measure to achieve simplicity, it has been assumed
a) Titles of input streams and tapes in the Job Description do not start on a new line - they are always preceded by the logical number

b) Following the heading DATA, the title or the document proper start on the next new line.

3. This routine occupies block *3403 in the object program area of store.

R290: Call Library Program N

Purpose: An S.E.R. in main store to implement the extracode 1147 "Call Library Program N and set c' = ba". If the program N is not listed in the current list, or is not assembled in store or is already in use, the object program is halted until the library program becomes available. If available, the program is copied to the object program store area, the supervisor copy being lost or retained according to the type of library program. Main control is set up and the object program resumed.

Registers of Main Store: 52

Instructions Obeyed: Maximum 38 plus entries to locate compiler, Assemble Compiler, Copy Compiler and Release Compiler.

Parameters Used: (1) to (24)

Subsidiary Store: (13) = Compiler record

One half word in current program working area holding zero if compiler has been retained in supervisor store; otherwise, compiler position in list, digits 23-2, for compiler copied into store.

Main Store Tables: Three compiler directories, each one holding information in successive half words relating to compilers currently in use.

(14) = Compiler Directory E — Digits 22-12 = Initial block label in Supervisor store. Zero if not assembled
Digits 23, 11-0 Action bits

(15) = Compiler Directory F — Digits 22-12 = Initial block label in Program store (also the standard entry address). Remaining digits unused by R290

(16) = Compiler Directory B — Digits 23 - 19 = a
18 - 11 = b
10 - 3 = c

where compiler occupies c blocks, and requires a x + b blocks to operate, x being the length of the input stream
Digit 1 = 1 if to be protected (can be damaged in use.
0 if no protection necessary (unaltered in use)
Digit 0 = 1 if common, small
0 if rare and/or large
Digit 2 = 1 if common
0 if rare.

R290: continued

Cross references:

(3)	=	(2/291)	Locate Compiler
(4)	=	(1/213)	Halt SER
(5)	=	(3/291)	Assemble compiler
(6)	=	(1/202)	Program scan
(7)	=	(1/249)	Halt program
(8)	=	(1/292)	Copy compiler
(9)	=	(1/293)	Release compiler
(10)	=	(4/247)	Exit to object program
(11)	=	(5/203)	Current program
(12)	=	(15/204)	Program switch directory

Connections with other Routines

Entered at (1) in Supervisor with full recovery switch set, via (1/247)
B119 = compiler number N, digits 14-3, remaining digits irrelevant.

Exit when compiler available to (4/247) with Main Control reset
when compiler not available, to (1/202) with program halted
when compiler directory full and compiler n not in list, to (1/202) with program halted for full list.

Alternative entries

At (23) for subroutine "Halt Program for full list"
Exit to program scan via (1/213)

At (24) for subroutine "Halt Program for compiler"
B106 = Directory E
B109 = Compiler number (position in directory)
Exit : To program scan via (1/249)

Subroutines:

a) "Locate compiler" entered at (2/291) with
B101 = Compiler number, digits 23-13. Rest zero
B110 = Return address
Return with B109 = 0 No entry available
B109 = n Compiler in entry n in directory (digits 23-2)

b) "Assemble compiler" entered at (3/291) with
B109 = Compiler directory position (digits 23-2)
B110 = Return address
Return with B109 even : Compiler assembled and ready for use
odd : Awaiting assembly or in use.

c) "Copy Compiler" entered at (1/292) with
B109 = Compiler directory position (digits 23-2)
B104 = Return address, digits 23-3
Digit 0 = 0 if lose supervisor copy
= 1 if retains supervisor copy
Return with compiler copied and supervisor version lost (if B104 even on entry). B101-103, 105-110 altered.

d) "Halt SER" entered at (1/213) with
B109 = 3.0 (Halt reason : Full list)
Exit to program scan with object program halted.

e) "Halt program" entered at (1/249) with
B106 = Return address : Program scan
B107 = Reason for halt, digits 20-12
(Compiler not available)
Exit to program scan with object program halted.

f) "Release compiler" entered at (1/293) with
B108 = Digit mask for program use of compiler
B109 = Compiler directory position (digits 23-2)
B110 = Return address
Exit to return address with B106-108, B110 altered.

g) "Exit to Main Program" entered at (4/247) with
B91 = Contents of ba.
Exit, after resetting full recovery switch, to main control

Temporary Working Space: B91, B100-110, Bt

Notes:

1. The compile switch (digit 19 of program switch directory) is forced to 1.

2. If the compiler does not occur in the current list, a routine is entered via "Locate compiler" to plant it in the list, reading its entry from a main list normally recorded on a library tape.

3. If the compiler is in the current list and is not in use elsewhere, it is transferred to the object program area of the block directory. The copy in Supervisor store is retained or lost block by block, according to the type of program as recorded in the compiler directory.

4. Main control is set to ba if non zero, or to the compiler initial address if ba is zero.

5. If the compiler does not require protection, no copy is kept in the Supervisor store. The compiler directory number is recorded in the current program working area of subsidiary store and the compiler will be returned to supervisor store on End Compiling. If a copy is retained in supervisor store, the record in the current program working area is set to zero.

R291: Assemble Compiler

Purpose: A supervisor subroutine to initiate assembly of a compiler or library program into supervisor store. If the compiler is not recorded in the current list, its entry to this list is initiated. Assembly in store is initiated unless it is already assembled or is in use elsewhere. Alternative entries to the routine are available for finding the location of the compiler in the current compiler list, and for starting of assembly when the compiler is located in the list.

Registers of Main Store: 73

Instructions Obeyed: Vary markedly with composition of list. Approximately 10 per list entry to locate the compiler if it is already on the list, together with 20 to initiate assembly.

Parameters Used: (1) to (24)

(10) = Number of entries in current compiler directories, digits 23-2

Main Store Directories:

(11)	=	Compiler directory A1	Last 4 characters of title. Successive half words for various compilers.	
(12)	=	Compiler directory A2	First 4 characters of title: Title of less than 8 characters filled out with zero characters.	
(13)	=	Compiler directory C	Digits 23-13	Compiler number
			Digits 12-0	Position on library tape
(20)	=	Compiler directory D	Digits 12-0	Dump tape position. Zero if not dumped.

Cross references:

(14)	=	(14/290)	Compiler directory E (state)
(15)	=	(15/290)	Compiler directory F (entry address)
(16)	=	(16/290)	Compiler directory B (description)
(21)	=	(5/201)	SER re-entry address
(22)	=		Routine : Enter to list
(23)	=	(1/466)	Routine : Start read compiler from tape
(24)	=	7(3/230)	Routine : Enter SER to slow queue

Connections with other routines:

Entry at (1) to locate and assemble
B110 = Return address
B101 = Number if known, digits 23-13
If not known, B101 = 0, in which case
B102 = First 4 characters of title
B103 = Last 4 characters of title (both filled with zero if less than 8 characters)

R291: continued

Exit: To return address with
B109 = 0 Not possible to enter compiler in list
B109 = n Digits 23-2 Compiler is in location n of list
Digit 0 = 0 Assembly complete, compiler not in use
Digit 0 = 1 Assembly requested but incomplete or compiler in use. In particular, the list entry may be incomplete, awaiting the calling of the main list from the library tape.

Alternative entries:

At (2) for subroutine "Locate Compiler"
B110 = Return address (see note 2)
B101-103 as for entry (1)

Exit : To return address with
B109 = 0 Not possible to enter compiler in list
B109 = n Digits 23-2. Compiler is in location n of list
If newly entered to list, compiler directory B is zero.
B105 - 109 altered.

At (3) for subroutine "Assemble Compiler"
B110 = Return address
B109 = Compiler directory location, digits 23-2

Exit : Immediate if B109 = 0, to return address
Otherwise

a) To return address with B109 digits 23-2 unaltered
B109 even. Assembly complete, compiler not in use
B109 odd. Assembly requested but incomplete or compiler in use.

b) To re-entry address when assembly requested and compiler not otherwise in use. The routine will be re-entered and the compiler found to be in use, with assembly requested B105-109 altered.

Subroutines:

a) Enter to list:

Entered at (22) with
B110 = Return address
B109 = Compiler directory number

Exit to return address with entry completed in list.

This routine is only entered when the main list ("compiler 0") is assembled in main store. The return link is to the re-entry address so that assembly of the compiler is re-requested, this time with a completed list entry. The incomplete list entry holds the compiler number or, if this is zero, the compiling title. The library position is recorded as 4095 and the description is zero.

R291: continued

b) Call from system tapes

Entered via slow SER queue. The routine scans the compiler directories E for entries with both digits 8,9 set and enters a request to the system tape queue to read the title block of any such compiler. Digit 8 is reset to zero to indicate assembly started. After making the entry to the system tape queue, control is transferred to the re-entry address, resulting in an attempt to assemble the compiler again. The compiler is now found to be in use and exit is made normally from R291.

Temporary working space: B105 - 109

Notes:

1. It is assumed that the re-entry has been set before entry to this routine so that entry may be repeated after assembly the current compiler list or after making an entry to the system or dump tape queue. The re-entry address is not reset by this routine.

2. When entered at (2) and the compiler is not found in the list, an immediate exit is made if the return address in B110 is odd, Otherwise, a vacant entry is acquired, if there is no vacant entry, an entry corresponding to an "Uncommon" compiler is used. If there is such an entry for a compiler not in use, the entry is lost immediately, together with the copy of the compiler in supervisor store. If all such entries are in use, a marker is left requesting loss of the entry when all other activities are concluded. In this case the exit is with B109 = 0 (not possible to enter new compiler to the list).

R292: Copy compiler

Purpose: A supervisor subroutine in main store to copy a compiler or library program from supervisor store to the store of the current object program, and to lose or retain the copy in supervisor store. An alternative entry enables the routine to be used to lose the supervisor copy of the compiler without copying to object program.

Registers of Main Store: 48

Instructions obeyed: 4 + 14b if lose only
4 + 35b (maximum) if copy and lose
together with entries to Locate Block, Lose block, and entries to drum routines to duplicate.

Parameters Used: (1) to (19)

Cross references:

(7)	=	(2/630)	Locate Supervisor block
(8)	=	(1/203)	Locate Program block
(9)	=	(19/315)	Halt for full drum queue
(10)	=	(6/204)	Halt for program block
(11)	=	-3(14/314)	Enter to drum queue
(12)	=	6(14/317)	Lose block
(13)	=	-3(23/205)	Amend free block counter
(14)	=	(14/290)	Compiler directory E (state)
(15)	=	(15/290)	Compiler directory F (initial block)
(16)	=	(5/201)	Re-entry address
(17)	=	(3/203)	Block status directory
(18)	=	(4/315)	Entries in drum queue
(19)	=	(5/203)	Program number

Connections with other Routines

Entry at (1) with B109 = Compiler directory number (digits 23-2)
B104 = Return address (digits 22-3)
Digit 23 = 1 if lose only
= 0 if copy
Digit 0 = 1 if retain supervisor copy
0 if lose supervisor copy

Exit: To return address with B101 - 103, 105 - 110, Bt altered. If the compiler is copied, the object program is halted until the completion of all copy operations, which are carried out by drum transfer.

Subroutines:

a) "Locate Supervisor block" entered at (2/630)
with B108 = Supervisor block label, digits 22-12
Rest zero
B110 = Return address
Return with B105 = Location in block directory.

R292: continued

b) "Locate program block" entered at (1/203) with
B109 = Program block label, digits 22-12.
Digits 23, 0 = 1
B110 = Return address
Return with
B108 = Program number (digits 20-14)
Location in block directory relative to start of area for current program (digits 12-2)
Digits 23 = 1. Remaining digits zero.

c) "Halt for full drum queue" entered at (19/315)

Exit to program scan with object program halted, B100-104 preserved.

d) "Halt for program block" entered at (6/204) with
B110 = program number.
B109 = Block number, digits 22-12. Remainder irrelevant
B108 = Return address. Digit 0 = 1

Exit to return address with B100-104 preserved. Re-enter at re-entry address when block made free.

e) "Enter to drum queue" entered at -3(14/314) with
B100 = Program number (digits 20-14)
Location in block directory relative to start of area for current program (digits 12-2). Rest zero
B102 = Location of supervisor block relative to start of block directory (digits 12-2)
Digit 13 = 1 (duplicate)
Digit 21 = 1 (do not change timer)
Digit 0 = 1 (Operand)

Exit to program scan. Program resumed with supervisor block duplicated to object program area.

f) "Lose block" entered at 6(14/317) with
B100 = Return address
B105 = Location in block directory relative to start of block directory
Exit to return address, B106 - 110 altered.

g) "Amend counter of free blocks" entered at -3(23/205) with
B110 = Return address
Exit to return address with B105 - 110 altered.

Temporary Working Space: B101 - 103, B105-110, Bt.

R292: continued

Notes:

1. The compiler or program in supervisor store is recorded in the area of the block directory reserved for Program 0. The blocks have non-reserved block labels, generally corresponding to the location in the Block Directory. The initial block label is recorded in Compiler Directory E.
The block timers of these blocks are unaltered; digits 22-3 of the block status directory are used as follows:

   Digits 22-12 Block label of next block in supervisor area. Zero for last block.

   Digits 11-3 Block label used in Program store relative to start.

   The initial block label in program store is recorded in Compiler Directory F.

2. If any blocks of main program exist with block labels equal to those required for the compiler, they are overwritten by the copy operation. If insufficient space is available in the program allocation, more will be obtained, since the compiler switch will be set during this operation. Blocks are obtained in the program area and lost in the supervisor area one at a time, thus reducing the maximum demands on store space to a minimum.

3. When a block is lost from the supervisor area of store, the counter of free blocks is increased by 1.

4. The routine may be used to copy or lose any blocks of Supervisor store linked as described above by entry at -1(2) with

   B104 As on entry at (1)

   B102 Leading supervisor block label, digits 22-12. Remainder irrelevant.

   B101 Leading program block label, digits 22-12. Remainder irrelevant.

5. It is assumed that when copying is required, the SER base is zero on entry (i.e. the SER is based on the current main program). When entered for "Lose" only, any SER base is permissible for the routine will never be halted.

R293: Release compiler

Purpose: A supervisor subroutine in main store entered after completion of any action carried out on a compiler (assembly to list, assembly in store, copying to object program, dumping or losing from main store). The routine initiates the next action required, if any, on the compiler.

Registers of Main Store: 28

Instructions Obeyed: Between 5 and 23, depending on actions required.

Parameters Used: (1) to (15)

Cross references:

(5)	=	(13/291)	Compiler directory C (number)
(6)	=	(14/290)	Compiler directory E (state)
(8)	=	(20/290)	Action bit : Required for main program
(9)	=	(22/290)	Halt reason: Compiler in use (digits 20-12)
(13)	=	(3/291)	Assemble compiler
(14)	=	(2/249)	Free programs
(15)	=	(1/294)	Dump/lose compiler

Connections with other routines

Entered at (1) with
B110 = Return address
B109 = Compiler directory entry (digits 23-2)
B108 = Digit mask for directory E (see note (1))
Exit: To return address with next action initiated if any
B105-110 altered

Subroutines:

a) Assemble compiler : Entered at (3/291) with
B110 = Return address
B109 = Compiler directory entry
Exit : To return address or to re-entry address

b) Free program : Entered at (2/249) with
B106 = Return address
B109 = Reason for making free
Exit to return address with all programs made free which were halted for this reason.

c) Dump/lose compiler : Entered at (1/294) to remove the compiler from store.
B110 = Return address
B109 = Compiler directory entry
Exit to return address with compiler lost or dumping initiated.

Temporary Working Space: B105 - 110

R293: continued

Notes:

1. Compiler directory E holds bits in digit positions 23, 11-0 which denote actions required and action being taken. On conclusion of any action, R293 is entered with B108 set to the mask causing the current bits corresponding to the action completed to be removed. The table below gives the mask for the various actions.

2. After masking, if the compiler is still recorded as in use, immediate exit is made. Otherwise, the actions required are scanned in the following order, and the actions described taken

   a) Assemble in store:- If already assembled in store, cancel this request and proceed. Otherwise start assembly, setting a bit to indicate assembly in progress.

   b) Required for object program:- Cancel this bit, free all object programs halted for compiler availability, and exit.

   c) Lose compiler from store:- Cancel this bit and enter R294 to dump or lose the compiler. If lost immediately, this routine will re-enter R293.

   d) Lose list entry:- The entry is vacated by clearing directory C to zero. Any programs awaiting a vacancy will be made free by the one second SER R230 which makes free all programs halted for reason 3.0.

3. On conclusion of actions, R293 is entered with the following masks in B108

Action Ended	Mask in B108
Entry to list	*00000017
Assembly in store	*77776777
Copying to Object Program	*77775777
Restoring from object Program	*77777737
Losing or dumping	*40007375

4. If the compiler is assembled and assembly is required, the scheduler is activated (through use of program scan exit) to use the compiler in pre-assembly of object programs.

R294: Dump or Lose compiler

Purpose: A supervisor subroutine in main store entered from space allocation routines when a compiler or library program is to be deleted from store. The routine decides whether to lose or to dump, and if the latter, calls for the dump tape.

Register of Main Store: 34

Instructions obyed: 10 to 16

Parameters Used: (1) to (18)

Cross refernces:

(6)	=	(11/244)	Dump tape record
(7)	=	(20/291)	Compiler directory D (dump record)
(8)	=	(16/290)	Compiler directory B (description)
(9)	=	(15/229)	Miscellaneous marker
(10)	=	(14/290)	Compiler directory E (state)
(11)	=	(1/292)	Copy compiler
(12)	=	(1/293)	Release compiler
(13)	=	(13/230)	Insert SER to tape queue
(14)	=		Write compiler to dump tape

Connections with other routines:

Entered at (1) with
B110 = Return address
B109 = Compiler directory number (digits 23-2)
Exit to return address with compiler lost or waiting to be lost or dumped. B101 - 108 altered

Subroutines:

a) "Copy compiler" entered at (1/292)
B109 = Compiler directory number (digits 23-2)
B104 = Return address digits 22-3. Digits 23 = 1. Digits 2-0 = 0
Exit to return address with B101 - 110 altered.

b) "Release compiler" entered at (1/293) with
B110 = Return address
B109 = Compiler directory number, (digits 23-2)
B108 = Mask *40007375
Exit to return address with next action, if any, initiated

c) "Insert SER to tape queue" entered at (13/230) with
B110 = Return address
B109 = Address of SER to start dumping compiler.

Temporary Working Space: B107, 108

R294: continued

Notes:

1. If the compiler is in use, or is required for a main program or for assembly, a record is left that the compiler is to be dumped, and an immediate exit is made.

2. The compiler is lost in store without dumping if there is no dump tape, if dumped already, or if it is classed as "rare" and the library tape is loaded.

3. Dumping is initiated if the dump tape is loaded, the compiler has not previously been dumped and if the library tape is not loaded or the compiler is classed as "common".

4. If dumping is required, the compiler is marked as being dumped. If the dump tape is free, an entry is made to the tape queue to initiate writing. Otherwise the tape is recorded as "required for dumping compiler".

R295: Record compiler from object program

Purpose: A supervisor routine in main store based on the current object program which copies a compiler back from object program store to supervisor store, losing the copy in the object program store and reducing the program store allocation. Used on End Compiling or Define Compiling. The routine exits to repeat the calling sequence, with switches set to prevent repeat entry to R295.

Register of Main Store: 54

Instruction obeyed: Dominated by N entries to R260 to reduce program store allocation by one block, where N is the number of blocks in the compiler.

Parameters Used: (1) to (28)

Cross references:

(6)	=	(1/203)	Locate block
(7)	=	(1/630)	Aquire one block
(8)	=	(1/312)	Set PAR
(9)	=	(2/630)	Locate Supervisor block
(10)	=	(21/261)	Entry to reduce program store
(11)	=	(2/350)	Find next block label
(12)	=	(4/247)	Return to Main Control
(13)	=	(1/293)	Release compiler
(14)	=	(14/290)	Compiler directory E (state)
(15)	=	(15/290)	Compiler directory F (first block)
(16)	=	(16/290)	Compiler directory B (description)
(18)	=	(1/203)	Block directory
(19)	=	(3/203)	Block status directory
(20)	=	(35/203)	Block location table
(21)	=	(5/201)	SER re-entry address
(22)	=	(6/203)	Page directory
(23)	=	(5/203)	Program number
(24)	=	(4/203)	Program store directory
(26)	=	(13/290)	Compiler record in program working area
(27)	=		Store for Ba in Extracode working space

Connections with other routines:

Entry at (1) with

B95 = compiler directory number, digits 23-2
digit 0 = 0 : Return to R261 (End compile)
digit 0 = 1 : Return to R719 (End program)

R295: continued

Subroutines:

a) "Locate Block" entered at (1/203) with
B109 = block label digits 22-12. Digits 23 = 1
(Halt if lock out)
Remaining digits zero
B110 = return address
Exit to return address with block location table holding position in block directory relative to the start of the block directory.
Digits 23-2
B105 - 110 altered

b) "Aquire one block" entered at (1/630) with
B101 = Reason for request
B110 = Return address
Exit to return address with
B105 = Location of new entry in block directory

c) "Set PAR" entered at (1/312) with
B108 = Block label (*4)
B109 = Page number
B110 = Return address
Exit to return address. B107 altered

d) "Locate Supervisor block" entered at (2/630) with
B108 = block label digits 22-12. Rest zero
B110 = return address
Exit to return address with B105 = Location in block directory relative to the start.

e) "Change program store allocation" entered at (21/261) with
B102 = new store allocation, digits 23-13
B100 = Return address
Exit to return address with B91-99 unaltered

f) "Find next block label", entered at (2/350)
B100 = block label n digits 22-12. Digit 0 = 1
B101 = Return address
Exit to return address with B91 = next block label > n or *4 if no such block label. B107-110, Bt altered.

g) "Release compiler" entered at (1/293) with
B108 = Mask (*77777737)
B109 = Compiler directory number, digits 23-2
B110 = Return address
Exit to return address with next action, if any, initiated.

R295: continued

Temporary working space: B91-97, B100-110, Bt

Notes:

1. The compiler directory holds the initial block label in program store, B and the number of blocks, N. The first N blocks from B upwards are removed from the program store area and inserted in the supervisor store area. After each block is transferred, the program store allocation is reduced by 1.

2. The block timers in the supervisor area are not used, and the block status directory of each block is set to contain
   Digits 22-12 next block label (zero for last block)
   Digits 11-3 program block label relative to start
   The initial block label is recorded in compiler directory E,
   Digits 22-12

3. If any of the blocks is in core store, the page address register is set to *4 and the page directory set to correspond to the new entry, with digit 21=1 (no change of block timer)

4. It is assumed that the calling routine has recorded the contents of ba in store address (27) of subsidiary store (a half word in extracode working space) before entry, and that entry to supervisor was earlier made via (1/247).

Purpose: To enter a drum transfer routine following a non-equivalence interrupt and to distinguish an instruction and operand request.

Registers of Fixed store: 18

Number of Instructions Obeyed: 8 for exit (a)
16 (maximum) for exit (b)

Parameters Used: (1) to (11)

(6) = V store address of "block non-equivalence address" = 34*6001
(8) = Address of instruction (113 0 0 3*6) to switch to M or E
(11) = Lowest reserved block label, digits 22-12 = *34

Cross references:

(2) = (2/201) Entry B to supervisor
(3) = (1/314) Entry to drum transfer routine
(4) = (10/201) In supervisor switch
(5) = (5/201) Current SER entry address
(9) = (5/203) Current program number
(10) = (1/318) Entry to Call to Cores

Connections with other routines:

Entry at (1) from R500

Exit (a) to "Enter Supervisor" using interrupt control
If operand, B111 digits 22-12 = block number b
digit 23 = 0 (operand)
digit 1 = +1/+2 (see note 2)
If instruction B111 digit 23 = 1 for instruction
digit 1 = irrelevant
Remaining digits zero
B112 = address of entry to "drum transfer routine"

(b) To "call to cores" using extracode control

B109 digits 22-12 = block number b
digit 0 = 1 (operand) 0 (instruction)
digit 1 irrelevant
digits 8-2 = 0 ($b \geq *3400$)
= current program number ($b < *3400$)
Remaining digits zero
B110 Current SER re-entry address, digits 23-0

Temporary Working Space:

B111 - 112 for exit (a)
B111 - 112, 109 - 110, Bt for exit (b)

16.7.62

Notes:

1. The "In Supervisor Switch" is examined to determine whether the interrupt occured in supervisor or not. If digit 0 = 0, the interrupt occurred in supervisor.

2. The current control is not reduced to the address which caused the interrupt.

3. Exit (b) causes the block of supervisor or current program to be called to core store and the current SER to be resumed at the re-entry address. If the block is already in core store the SER will still be resumed at the re-entry address.

R312: Change Page Address Register

Purpose: A supervisor extracode routine to change the contents of a page address register without setting up a spurious drum or tape address in transit.

Registers of Fixed Store: 8

Number of Instructions Obeyed: 8

Parameters Used: (1) to (2)

(2) = "Contents of page address registers"
Entries are in alternate half words, entry P corresponding to page P. Digits 23-12 are the corresponding digits of the PAR. (Digits 11-0 are the corresponding digits of B108 the last time this routine was obeyed).

Connections with other routines:

Entry at (1) with B108 digits 23-12 new contents of PAR
B109 digits 23-12 & 2 zero
digits 10-3 page number
B110 return address

Temporary Working Space: B107

Purpose: A supervisor extracode routine to determine the next empty sector on fast drums and to initiate a write to drum transfer to that sector.

Register of Fixed Store: 55

4 drums: minimum 45, maximum 328
32 drums: minumum 66, maximum 2288

Part of the time interrupts are inhibited and a maximum of 74 instructions for 4 drums (96 for 32 drums) are obeyed with interrupts inhibited.

Parameters Used: 1 - 10

(1) = entry
(2) = table of empty sectors
(8) = address of drum V-store = *6002
(9) = number of drums in digits 7-3
(10) = drum check location C.

Subsidiary Store: "Table of empty sectors".

The entry of drum d and angle θ is in half line $3d + \theta/2$.
Digits 7-0 indicate the state of the relevant sectors on bands 7-0 and are 1 if empty, 0 if not empty. Digits 23-8 are zero.

Drum check location : one half word used to check completion of a drum transfer.

Temporary Working Space: B100 - 107, Bt

Entry : B110 = return address

Exit : B100 = cabinet in digits 4-2
B101 = drum in digits 3-2
B102 = band in digits 4-2
B103 = θ in digits 5-3
B104 = $6d + \theta$ in digits 23-2

None of the subsidiary store directories or page address registers are changed in this routine.

Notes:

1. Method

The drums are considered in turn and for each drum the 'wait' for the next empty sector is determined (this wait is the miniumum angle between the present θ and an empty sector - 1). If the wait is zero for any drum no further drums are considered.

2. Inhibit Interrupts

In order to get the drum transfer started soon after reading the present θ, interrupts are inhibited when a drum is first considered. If the wait is zero, interrupts are permitted when the transfer is started but if the wait is not zero interrupts are permitted during the further consideration of the drum.

3. Checks

No checks are included to see if the present θ has changed significantly between first reading it and starting the transfer. Also, no check is made that the present θ is read incorrectly while it is actually chaning.

4. Drum Check

When the drum is started, the drum check location in subsidiary store is set to zero.

R314 Drum transfer routine

Purpose: This is a sequence of SER's to enter a non-equivalence or read to cores drum transfer in the drum queue and perform the transfer. Alternative entries are provided from "call to cores", "read to page P" "write to drum", "duplicate", and to update empty sectors.

Registers of Fixed Store 190

Instructions obeyed: 150 normally
1148 if the block is not allocated

Subsidiary store:

(25) = Drum transfer working space

Two half words of private working space
(25) = number of empty page in digits 23-3
0.4(25) = number of selected page in digits 23-3
This is used while the drum transfer is in progress but not while the drum transfer is dormant in the drum queue.

(45) = Number of drum transfers

One half word in the current program working area holding number of non-equivalence or extracode read to cores drum transfer obeyed, digit 23-3.

Fixed store table:

(23) = "Table of powers of 2"
Eight consective half words, entry n containing 2^n in digits 23-0.

Parameters used: (1) to (47)

Cross references

(4)	=	(5/201)	Current SER entry address
(5)	=	(2/315)	Address of first entry in drum queue
(6)	=	(11/304)	Upper page limit for learning program
(7)	=	(6/204)	Halt main program
(9)	=	(1/203)	Store location and lock out
(11)	=	(35/203)	Block location table
(12)	=	(1/315)	Drum queue routine
(13)	=	(4/203)	Program store directory
(15)	=	(31/303)	Alternative entry to learning program
(16)	=	(1/302)	Entry to Page Selection routine
(17)	=	(1/312)	Entry to change page address register
(18)	=	(1/203)	Block directory
(19)	=	(20/315)	Drum queue
(22)	=	(2/313)	Table of Empty Sectors
(24)	=	(6/203)	Page directory
(26)	=	(1/202)	Program scan
(27)	=	(2/205)	Unlock store block (cores)
(28)	=	(6/315)	Alternative entry to drum queueu routine
(29)	=	(1/313)	Write to next empty sector
(33)	=	(5/203)	Current program number
(34)	=	(1/327)	Entry to preserve and reserve accumulator

(35)	=	(3/327)	Alternative entry to preserve and restore accumulator.
(38)	=	(10/323)	Alternative entry of "duplicate block b to cores"
(39)	=	(27/323)	Alternative entry of "duplicate block b to cores"
(40)	=	(7/ 201)	Current SER base
(46)	=	(10/313)	Drum check location
(41)	=	(6/201)	Main program controls
(47)	=	(1/201)	Enter supervisor

Other parameters:

(1)	=	Entry address from one equivalence interrupt
(8)	=	Entry address from extracode "Read to Cores"
(42)	=	Address of drum V store = *6002
(32)	=	Entry to record empty sector
(36)	=	Entry to find block location
(14)	=	Entry from Call to Cores and from drum queue
(43)	=	Entry from Read to Page P
(44)	=	Entry from Write to drum

Connections with other Routines

Entry a) at (1) from "Non-eqivalence Interrupt", R311, via R201
B100: Digit 23: 1 if instruction, 0 if operand
Digits 22-12: block number b (operand);
zero if instruction
Digit 1 = +1/+2 (irrelevant if instruction request)
Remaining digits zero.

Entry b) at (8) from extracode jump table "Read block b to cores"
B119: Digit 23: 1 if instruction, 0 if operand
Digit 22-12: block number b
Digit 11-0 irrelevant

Entry c) at (14) from drum queue routine on "Call to Cores" when drum transfer is first in the drum queue
B102 = Block specification
B104 = Address of first entry

Entry e) at (36) to find the block location D
Entry: B102 digits 20-14 = program number P
digits 12-2 = block directory location relative to start of area reserved for program P
B110 = return address
B108 digits 12-2 = block directory location
Exit: B108 digits 12-2 = block directory location relative to start of block directory. Remaining digits zero.
B107 used as temporary working space.

Entry f) at 15(21) to initiate drum transfer
Entry : B100 = Sector number, digits 11-1
Remainder irrelevant
B108 = Return address

Exit: Read to cores drum transfer initiated and drum check location cleared. B103, 106, 110 used as temporary working space.

At 16(21) with B103 = number of sectors.
Otherwise as (f)
At 17(21) with B103 = number of sectors.
B110 = 7.0(read), 6.4(write)
otherwise as (f)

Subroutines:

(a) "Halt main program"
Entry (6/204) for block b not available
B108 = return address 3(8)
B109 = digits 22-12 = b
Remaining digits zero.

Exit to 3(8/314)

(b) "Store location and lock out"
Entry at (1/203) with
B109 digits 22-12 = block number b
digits 2-0 = 1
remaining digits zero
B110 return address

Exit: Either to return address with
B108 digits 23 = 1
digits 20-14 = Program number P
digits 12-2 = block directory location D of block b relative to the start of the area reserved for program P.

remaining digits zero or to program scan; re-enter this SER at 7(8)

(c) "Drum Queue Routine"
Entry at (1/315) B101 = 0
B102 = P/D
B103 = Entry of drum transfer routine (14/314)
Exit B104 = address of first entry
B102 = block specification

Entry at (6/315) to next drum transfer routine or to program scan.

(f) "Change page address register"
Entry at (1/312)

B108 digit 23 = lockout digit
digits 22-12 = block number
remaining digits zero.

B109 digits 23-3 = page number
Remaining digits zero

B110 Return address

(g) "Program scan"

(h) "Unlock store block"
Entry at (2) after drum transfer to core store
B109 digits 23-3 = page number
B110 return address

(i) "Preserve and restore accumulator"
B110 = return address.

Other connections:

a) Enters "duplicate block b to cores" at (10/323)
B110 = entry in block directory for block b_1
B102 = digits 20-14 = program number P
digit 13 = 1 for duplicate
digits 12-2 = directory position of block relative to start of area for program P
B108 digits 12-2 = directory position relative to start of block directory
B109 digits 11-3 = number of empty page P.
Remaining digits zero
B103 = 0 if b_1 not allocated, non zero otherwise

b) Enters "duplicate block b to cores" at (27/323)
B102 as above
B109 digits 11-3 = number of page just read, remaining digits zero.

c) Entered at 2(20/314) from "duplicate block b to core"
B109 digits 1103 = number of page just read, remaining digits zero.

d) Entered at -5(14/314) from "duplicate block b to cores" to enter request to drum queue.

B100	=	P/D for b_2	
B102	=	Digits 20-14	Program number P
		Digits 13	1 for duplicate
		Digits 12-2	Directory position of block b, relative to start of area for program P.
		Digit 0	operand/instruction
			Remaining digits zero

c) Entered at 3(14) from "Read to page P" with
B104 = address of current entry in drum queue
B109 = page number P digit 11-3. Remainder zero

Exit to program scan after initiating transfer.

f) Entered at (43) from "Read to page P with 0.4(25/314) = page number of page occupied by block b.

g) Entered at (44) from "Write block B" via learning program with
B109 = page number of block, digits 11-3.
Digit 0 = 1

Temporary working space: Entry a, b, c: B100-110
Two half words of subsidiary store
Block location table

Entry d : B106-108
Entry e B107-108

Notes:

1. When the drum transfer routine is entered from the drum queue there is always at least one empty page to which the required block is read. If the page is in the operand half and an instruction block is to be read, the learning program is entered to select a page in the instruction half and after writing this page to the drum the transfer proceeds. While the transfer is in progress, a page is chosen by the learning program. When the read transfer is complete, the chosen page is normally written to the drum to leave an empty page.
2. If when the read transfer is complete the chosen page is empty, the write to drum transfer is omitted. If the chosen page had been lock down another page is chosen by the learning program.
3. If a block had not been allocated previously, an empty page is assigned, and is cleared to floating point zero (using the accumulator, which is preserved and restored) or is left containing the previous contents, depending on a switch set by an extracode instruction or, in the case of supervisor transfer, depending on a digit recorded in the drum queue entry.
4. If the block is in the core store on entries (a) and (b), exit from this routine without entering the drum queue. If the block is in core store when the transfer is to be initiated, the block is unlocked and exit is to the next drum transfer routine or to program scan.
5. The "Current SER base" is changed to main program after entry from a non-equivalence interrupt.
6. For non-equivalence, the main program is halted for block b not available, and then block b is locked out. If there is also a halt for drum queue full, the drum queue release will occur first. The transfer is entered in the drum queue and the program restarted, and produces another non-equivalence. This time, since block b is locked out for read to cores, it exits to program scan and the program is finally released when the drum transfer (read) is completed and block b is unlocked.

   Similarly if the program is halted for block b locked out, it is resumed "in supervisor" and block b is then locked out for read to cores.

Purpose: To enter an item in the drum queue and begin dealing with it. An alternative entry steps on the drum queue to deal with the next item when one has been completed and there is a special entry for when the drum queue if full. The routine deals with both ordinary and priority entries.

Registers of Fixed Store: 62

Instructions Obeyed: 45 - 49 normally

Subsidiary Store:

Drum Queue: Each item occupies 5 consecutive half words.
The first four are in B100-103 on entry.

(B100) 0.0 Information on return, I/sector number/duplicate block/ page number

(B101) 0.4 Return address, digits 23-3 = address for return
digits 2-1 = SER queue (10 top, 01 tape, 11 slow)
digit 0 = zero
Return address = 0 indicates not required.

(102) 1.0 Block specification, digit 23 = 1 if clear new block
digit 22 = 1 if lockdown required
digit 21 = 1 if don't change timer
digits 20-14 = program number P
digits 12-2 = entry in block directory relative to start of area reserved for program P.
digit 0 = operand/instruction (1/0)

(103) 1.4 Address of drum transfer routine, the entry of the routine to deal with the next stage of this item.

2.0 Address of next entry
digits 23-2 = address of word 0.0 of the item to be done after this item is completed relative to the start of the drum queue.
digit 0 = 1 if there is an item in this space
= 0 if there is no item in this space
The last item contains the address of itself.

Address of first entry, one half word
digits 23-2 = address of word 0.0 of the item which is in progress, relative to the start of the drum queue.

Address of last entry, one half word
digits 23-2 = address of word 0.0 of the last item to be insterted in the drum queue, relative to the start of the drum queue.
digit 0 = 1 if the drum queue is not empty
= 0 if the drum queue is empty

Number of further entries, one half word
digits 23-3 = number of spaces for further items to be inserted in the drum queue
digits 2-0 = zero.
If the "number of further entries" $\leqslant 0$, the drum queue is full.

Parameters Used: (1) to (20)

(2) = address of first entry
(3) = address of last entry
(4) = number of further entries
(5) = end of drum queue i.e. word 0.0 of the end space in the drum queue relative to the start of the drum queue.
(20) = start of drum queue

Cross References

(9)	=	(1/202)	entry of "program scan"
(11)	=	(1/206)	alternative entry of "enter SER to queue"
(12)	=	(1/214)	entry of "free program"
(15)	=	(1/213)	entry of "halt SER"
(17)	=	(5/201)	"current SER entry address"

Connections with other routines:

Entry at (1) to enter item in the drum queue with
B100-103 = item (as above)

(6) when an item in the drum queue is completed.

(13) for priority entry with
B100-103 = item
B102 digit 1 = 1
B105 digit 5-3 = measure of priority

(19) to halt SER or main program for drum queue full

Exit (a) to the drum transfer routine for the first item with
B102 = block specification
B104 = address of first entry

(b) to "program scan"

(c) to "halt SER" with
B109 = 0 for drum queue full

Subroutines

(a) "Enter SER to queue"

B108 = information on return
B109 = starting address of new SER recorded in the drum queue
B110 = return address
B126 odd

(b) "Free program"

B109 = 0 for drum queue full
B110 = return address

Temporary Working Space: B104-110, Bt

Notes:

1. When the drum queue is empty (indicated by zero in digit 0 of the the "address of last entry") digit 0 of half word 2.0 of each space is zero to indicate no item in this space. The "number of further entries" is then 64 minus the number of allowable priority entries.

2. When the drum queue is empty the first item is inserted in the end space of the drum queue and thereafter in spaces where there are no items already working backwards through the queue.

3. If the drum queue is full the "current SER entry address" is changed and the SER main program is halted.

4. The priority entry to the routine occurs immediately after the drum queue full test. If the drum queue is empty the instructions obeyed are the same as those obeyed in the ordinary case. An item with priority n is inserted after entry n in the queue, the appropriate links being set. If there are less than n items in the queue the new entry is put at the end of the queue.

R316 : Read block b

Purpose : To do extracode drum transfers "read drum block b" by entering the non-equivalence drum transfer routine.

Registers of Fixed Store : 4

Number of instructions obeyed : 4

Parameters used : (1) to (3)

Cross references

(2) = (1/201) = entry address from extracodes of "enter supervisor" routine

(3) = (8/314) = Alternative entry to Drum Transfer Routine

Connections with other Routines

Entered with B119 = block label b, digits 22-12
Digit 23 = 0 operand
1 instruction
Remaining digits irrelevant

Exit to Drum Transfer routine with B100 = digits 23-12 of B119 on entry.
remaining digits zero.

Subroutine : Enter Supervisor, entered at (1) with B96 = return address

Notes : The drum transfer is dealt with in the drum queue as a non-equivalence drum transfer except that current program is not halted.

Purpose: To do the extracode "lose block b" by vacating the entry in the block directory and making the relevant page or sector empty. Alternative entries lose block b as a subroutine of an SER.

Registers of Fixed Store: 30

Number of Instructions Obeyed: 15 - 22

Parameters Used: (1) to (15)

Cross References:

(2)	=	(12/203)	alternative entry of "store location and lockout"
(4)	=	(1/202)	"program scan"
(5)	=	(35/203)	block location table"
(6)	=	(3/203)	block status directory
(7)	=	(2/203)	block directory
(8)	=	(1/312)	"change page address register"
(9)	=	(6/203)	page directory
(10)	=	(32/314)	alternative entry of "drum transfer routine
(12)	=	(5/203)	current program number
(15)	=	(1/201)	enter supervisor
(11)	=	(5/201)	SER entry address

Connections with other Routines:

Entry(a) At (1) for extracode: lose block b.
B119 digits 22 - 12 = block number, b
remaining digits irrelevant.

Exit to program scan.

Entry(b) At (3) for subroutine; lose block b of the current program
B100 = return address, E
B101 digits 22 - 12 = block number, b
remaining digits zero.

On exit B102 digits 8-2 = program number P.
remaining digits zero.

Entry(c) At 1 (3) for subroutine: lose block b of program P.
B100, 101 set as for entry (b)
B102 digits 8 - 2 = program number P.
remaining digits zero.

Exit to return address E.
B105 digits 12 - 2 = D, entry in block directory
relative to start of block directory.

Subroutines: (a) "enter supervisor"

Entry: B96 digits 23-3 = entry address, E
digits 2 - 0 = zero.

(b) "Store location and lock out"

Entry: B108 digits 8-2 = program number, P
remaining digits zero.
B109 digits 22-12 = block number, b
remaining digits zero or digit 23 = 1
digit 0 = 1 and remaining digits zero.
B110 = return address.

(c) "change page address register"

Entry: B108 = *4
B109 digits 11 - 3 = page number, P
remaining digits zero.
B110 = return address

(d) "drum transfer routine"

Entry: B100 digits 11-1 = sector number, s
remaining digits zero
B110 = return address.

Temporary Working Space:

Entry (a) B96 B100 - 102, B105 - 110
Entry (b) B100, B102, B105 - 110
Entry (c) B100, B105 - 110

Notes:

1. If the block is locked out the "lock out status", S is examined.
If S = 3, 4, 5, 6, 7 it is changed to 7 and exit.
If S = 1, 2 the "current SER entry address" is changed
and the program is halted in supervisor until block b is unlocked.

2. If the block is not allocated and not locked out, exit without doing anything.

3. If the block is not locked out and having been chosen by the "learning program" is being written to the drum, the block directory and sector are vacated immediately. The "write to drum" transfer is allowed to continue and when completed the page is vacated in the normal way.

R318: Call to Cores

Purpose: A supervisor subroutine to read down block b of program P and lock it down if required.

Registers of Fixed Store: 49

Number of Instructions obeyed: 12 to 32

Parameters Used: (1) to (19)

Cross References:

(4)	=	(11/213)	alternative entry of "halt SER"
(5)	=	(12/203)	alternative entry of "store location and lock out".
(6)	=	(14/314)	alternative entry of "Drum Transfer Routine"
(7)	=	(17/204)	special entry of Halt Main program
(8)	=	(35/203)	block location table
(9)	=	(6/203)	page directory
(10)	=	(8/303)	page timers
(11)	=	(2/205)	entry of "Unlock Store block"
(12)	=	(4/315)	number of further entries in drum queue
(13)	=	(19/315)	alternative entry of "drum queue routine"
(14)	=	(5/201)	current re-entry address
(15)	=	(7/201)	current SER base
(16)	=	(2/203)	block directory
(17)	=	(5/203)	current program number

Connections with Other Routines:

Entry at (1) with B109 digit 23 = 0 if clear new block
digits 22-12 = block number, b
digit 10 = 1 if lock down
digit 9 = 1 if don't change timer
digits 8-2 = program number P
digit 0 = 1 if operand, 0 if instructions
B110 = Return address E, digits 23-0, if in core store.

Entry at (2) with B108 Digit 23 = 1 if clear new block
Digit 22 = if lock down
Digit 21 = if don't change timer
Digits 20-14 = Program number P
Digit 13 = 0
Digit 12-2 = directory location of block relative to the start of the area for program P.
Digit 1 = 0
Digit 0 = 1 if operand, 0 if instruction
Block location table set up
B110 = Return address E, digits 23-0, if in core store.

Entry at (3) with Block location table set up
B108 as above only digit 1 = 1 (priority entry)
B110 as above
B109 = priority number

Exit: a) To "Drum queue routine" at (13)
B100 = information on return (dump address or initial B100)
B101 = re-entry address (digits 23-3,0)
SER base (digits 2,1)
B102 = block specification
digit 23 = clear new block
digit 22 = lock down
digit 21 = don't change timer
digits 20-14 = program number P
digit 13 = 0
digits 12-2 = entry in block directory relative to start of area reserved for program P
digit 1 = 1 if priority entry
digit 0 = operand/instruction
B103 = entry (14) of drum transfer routine R314

b) To "Halt SER" if drum queue full via entry (19) of drum queue routine with B100-104 unaltered.

c) To Return Address E if block is in core store with
B100-104, B110 unaltered
B109 = Page number digits 11-3, rest zero
On entry at (2) or (3) only B107, B109 are altered.

Subroutines

(a) "Store location and lock out"

Entry: B108 digits 8-2 = program number P,
remaining digits zero.
B109 digits 23=1 for halt requirement H
digits 22-12 = block number b
digits 2-0 = 1 for read to cores
remaining digits zero
B110 = return address

(b) "Halt SER" alternative entry (11/213) to preserve B100-104
Entry : B110 = Return address
Exit : B107 = Dump address

(c) "Unlock store block", entry (2/205), to unlock block in cores
Entry : B109 digits 11-3 = page number
remaining digits zero
B110 = return address E
Exit : To E with B109 preserved.

Temporary Working Space: B100-110, Bt, Block location table
On exit to E when block in cores,
B100-104, B110 are unaltered.

Notes:

1. When R318 entered at (1), the SER may be halted for lock out of store block b when P $\neq$ 0, provided the SER is based on the main program. If P = 0, the SER may be halted if block b is not in the one-level store, whatever the SER base.

2. When R318 is entered at (2) or (3), there is no halt for lock out etc.

3. When entry is at (1) or (2), the SER may be halted for drum queue full.

4. For all entries, the SER is halted in the drum queue if the block is not in core store.

5. After all such halts, the SER is resumed at the re-entry address. Digit 0 of the "current SER entry address" indicates whether B100-104 or only B100 are to be preserved.

6. Entry 3 causes a priority entry to be made in the drum queue, B109 on entry giving the priority.

   B109 = 1.0 for IBM tape
   2.0 for Program change dump block
   3.0 for N.E.P. tape
   4.0 etc.

7. Only if the block is in core store is return made to address E specified in B110 on exit. The extra digits are inserted in the page directory and the page timers are set (T=-1, t=0) if lock down. However, the operand/instruction digit is not changed in this case. The drum queue is not entered. The lock out digit and clock digit in the page directory, and the page address register are set up by R205.

R319 : Set page address register

Purpose: To set the contents of a page address register of a supervisor block to a reserved block number.

Registers of fixed store: 9

Instructions obeyed: 9

Parameters used: (1) to (5)

(1) = entry address (see below)

Cross References:

(2) = (12/203) alternative entry to "store location and lock out"
(3) = (35/203) block location table
(4) = (2/203) Block directory
(5) = (1/312) entry to "change page address register"

Connections with other routines:

Entry at (1)

B104 = return address
B103 digits 23 = lock out
digits 22 - 12 = reserved block number
remaining digits irrelevant.
B109 digits 22 - 12 = supervisor block number
remaining digits zero.

Alternative Entry at 1(1)

B108 digits 8-2 = program number P
remaining digits zero
B103, 104, 109 set as for entry at (1)
(B109 = block number)

Exit

B102 digits 11-3 = page number
remaining digits as in block directory
B109 digits 11-3 = page number
remaining digits zero

Subroutines

(a) Store location and lock out
Entry B108 digits 8-2 = program number,
remaining digits zero.
B109 digits 22 - 12 = block number
remaining digits zero
B110 = return address

(b) Change page address register
Entry B108 digits 23-12 = new contents of PAR,
remaining digits zero.
B109 digits 11-3 = page number,
remaining digits zero.
B110 = return address.

16.7.62.

Temporary Working Space : B102- 110 Bt

Notes :
1. It is assumed that on entry to R319 the block is in core store.
2. The block directory, page directory and block status directory are not changed by R319

R320 : Lose band b.

Purpose: To perform the extracode lose band b.

Registers of Fixed Store: 45

Number of Instructions Obeyed:

Parameters used (1) to (15)

Cross References:

(3)	=	(1/321)	Start of "Read/Write sectors"
(4)	=	(3/315)	Last entry in Drum Queue
(5)	=	(2/315)	First entry in Drum Queue
(6)	=	(20/315)	Drum queue
(7)	=	(2/203)	Block directory
(8)	=	(5/203)	Current program number
(9)	=	(4/203)	Program store directory
(10)	=	(6/321)	Bank directory
(11)	=	(36/314)	Alternative entry to Drum Transfer routine
(12)	=	(5/204)	Entry to Halt Main Program
(13)	=	(14/317)	Entry to Lose Block
(14)	=	(4/247)	Alternative entry to "Pass information to main program"

Connections with others:

Entry at (1) from Extracode "Lose band b" via fixed store sequence

121	91	0	(1/320)
121	126	0	(1/247)

"B122" = Band number, digits 10-3. Digit 23-11 zero½

Exist a) to Re-entry program, jump to (4/247) with
B121 = 0
Exit from there to reset full recovery switch and exit from Supervisor.

b) to Halt Main Program at (5/204) with
B109 = block label b digits 22-12 Remainder irrelevant
B110 = current program no. digits 8-2
if sector in use in the drum queue.

Subroutines: a) "Read/Write Sector B"
Entry at 2(1/321) with
"B122" = Band number digits 10-3.
Digit 23-11 zero.
B126 digit 1,0 = 1 0

Exit a) To Monitor (see R321)
b) To 1(1/320) with
B102 = Absolute band number, digits 11-4.
Remainder irrelevent

16.7.62.

b) Drum transfer routine to find block directory location
Entered at (36/314) with
B102 = Prog, no. (20-14)
Direct locate rel. to start of area reserved for P (12-2)
B110 = Return address
Exit
with B108 = Directory location rel. to start of block directory.

c) Lose block
Entered at 7(14/317) with
B105 = Block directory location rel. to start of block directory.
B106 = Contents of block directory location
B100 = Return address
Return with B105 unaltered.

Notes:

1. Routine is in main store

2. Entry is via R247 which sets full recovery switch in case of halts and sets re-entry address; any halt causes repeat of entire routine.

3. Exit is via R247 to reset the full recovery switch. As R247 also loads "B122", B121 is set to zero before exit.

4. If the band is involved in the drum queue, the program is halted for the first one-level store block involved in this transfer. When this is unlocked, the routine is repeated.

5. Entries in the block directory of this program referring to this band are vacated, the sectors are declared empty, and the band directory entry of this band is set to zero.

16.7.62.

R321.: Read/Write K+1 Sectors

Purpose: To perform the extracode read and write K+1 sectors

Registers of fixed Store: 148

Number of Instructions Obeyed:

Subsidiary Store: The band directory consists of consecutive half-words, one per band. Entry m for band m contains
Digits 20-14 program number
Digits 10-3 program band number

Parameters used: (1) to (36)

(6) = band directory
(33)= error marker
(34)= $\frac{N-1}{2}$ where N is the number of bands

Cross References

(2) = (1/201) Enter Supervisor
(3) = (5/201) SER re-entry address
(5) = (5/203) Current program number
(7) = (14/702) Monitor
(9) = (1/203) "Store location and lockout"
(10)= (4/315) Number of further entries
(11)= (19/315) Drum queue full entry to "drum queue routine"
(12)= (1/315) "Drum queue routine"
(13)= (7/203) Working space
(14)= 0.4(7/203) Working space
(15)= (8/203) Working space
(16)= (35/203) Block location table
(17)= 0.4(8/203) Working space
(18)= 0.4(11/226) Working space
(19)= (5/202) Program scan exit
(20)= -3(14/314) Alternative entry "drum transfer routine"
(21)= (3/315) Last entry in drum queue
(22)= (5/315) End of drum queue
(23)= (20/315) Start of drum queue
(24)= (2/203) Block directory
(26)= (36/314) Alternative entry "drum transfer routine"
(27)= 16(21/314) Alternative entry "drum transfer routine"
(28)= (30/314) Alternative entry "drum transfer routine"
(29)= (1/312) "Change page address register"
(30)= (2/329) Alternative entry "remove lock down"
(31)= (1/205) "Unlock store block"
(32)= (6/315) Alternative entry "drum queue routine"
(35)= 1(1/320) Lose band
(36)= (2/315) Alternative entry "drum queue routine"

Connections with other routines

Entry at (1) from read/write K+1 sectors with
B119 digits 22-12 = b
2-0 = K
B122 digits 10-0 = S
B126 digits, 1, 0 = 00 read or
= 01 write
Entry at 2(1) from lose band (R320) with B119, 122 as above
B126 digits 1,0 = 10 for lose band.

Subroutines:

(a) "Monitor", enter with
B100 = (33), error marker parameter
and B126 odd

(b) "Store location and lockout" enter with
B110 = return address
B109 digits 22-12 = b, first block label
5-3 = K, one less than number of block
2-0 = L lockout reason
23 = H, halt requirement

Exit with
B109 digits 22-12 = b+K
B108 digits 20-14 = Main program number P
12- 2 = Block directory location of block b
relative to start of the area reserved
for program P
23 = 1
remaining digits zero
B107 digits 2-0 = S previous lockout status of b.
remaining digits zero

"Block location table" Half word i holds location in block directory of block b+i relative to the start of the block directory in digits 12-2, remaining digits zero.

(c) Enter at (18/315) of drum queue routine to halt SER for drum queue full

(d) Enter at -3(14/314) of drum transfer routine to enter item in drum queue and return to program scan
B102 = block specification
digit 23 = 0 do not clear new block
22 = 1 lockdown required
21 = 0 change timer
20-14 = program number P
12- 2 = entry in block directory relative to start of area reserved for program P
0 = 1 operand

16.7.62.

(e) Enter at (36/314) of drum transfer routine to find block location with
B102 digits 20-14 = program number P
12- 2 = block directory location relative to start of area reserved for program P
B110 = return address

= Exit with B108 digits 12-2 = block directory location relative to start of block directory.
Remaining digits zero.

B107 used as temporary working space.

(f) Enter at 16 (21/314) of drum transfer routine to read K+1 sectors with
B100 = S (11-1), other digits irrevelant
B103 = K+1 (4-2), remaining digits zero
B108 = return address
and at 17(21/314) to write K+1 sections with
B110 = 6.4
and B100, 103 and 108 as above.

(g) Enter at (30/314) of drum transfer routine to exit to drum queue routine with B103 = return address

(h) Change page address register with
B108 digits 23-12 = new contents of PAR
B109 " 10-3 = page number
B110 = return address

(i) Enter (2/329) of remove lock down with
B104 = return address
B109 digits 11-3 = page number
Remaining digits zero
Exit with B109 digits 11-3 = page number
Remaining digits zero

(j) Enter unlock store block at (1/205) with
B109 digits 11-3 = page number
digits 23-12,0 = zero
B110 = return address
Exit with B109, 110 preserved.

Notes:

1. K+1 entries are inserted in the drum queue to read the blocks to store then a further entry deals with the multiple transfer; associated with this entry is a dummy entry in the queue which contains the number of blocks to be transferred and their locations in the block directory (D).

2. No account is taken of whether the band is already in use. Before losing a band however, the drum queue is checked for multiple sector transfers. If these concern the band to be lost then the band is not lost till the transfers are completed.

16.7.62.

3. The three types of entry in the drum queue are:

a) Ordinary Entry

Word	0.0	Zero
	0.4	Zero
	1.0	Block specification (see above)
	1.4	Return address
	2.0	Link to next queue entry

b) Multiple Sector Entry

Word	0.0	Read/write (23), Sector number (11-1)
	0.4	Zero
	1.0	Link to dummy entry
	1.4	Return address + 0.1
	2.0	Link to next queue entry

c) Dummy Entry

Work	0.0	D_0 (0,23-13), D_1 (12-2)
	0.4	D_2 (0,23-13), etc.
	1.0	etc.
	1.4	etc.
	2.0	Program number (20-14), K(4-2), 1(0).

16.7.62.

R322: Drum transfer complete interrupt

Purpose: To enter a drum transfer routine at the address specified in the first entry of the drum queue.

Registers of fixed store: 7

Number of instructions obeyed: 7

Parameters Used: (1) - (5)

Cross References

(2) = (2/315) = "address of first entry"
(3) = (20/315) = start of drum queue
(4) = (2/201) = entry address from interrupt of "enter supervisor"
(5) = (10/313) = drum check location.

Connections with other routines:

Exit to drum transfer routine with
B104 digits 23-2 = address of first entry

Subroutine "Enter supervisor"
Entry with B112 = entry address (using interrupt control)
Exit to entry address (using extracode control)

Temporary Working Space: B104, 112

Note: The "drum check location" is set negative by this routine.

16.7.62.

R323 : Duplicate block to core

Purpose: To perform the extracode to duplicate store block b to form block ba in the core store.

Register of Fixed Store: 72

Number of instructions obeyed:

Parameters Used: (1) to (31)

Cross References:

(4)	=	(35/203)	"block location table"
(5)	=	(3/317)	alternative entry of "lose block b"
(6)	=	(6/201)	"Current SER entry address"
(7)	=	(12/203)	alternative entry of "store location and lock out"
(9)	=	(14/314)	alternative entry of "drum transfer routine"
(11)	=	(3/205)	alternative entry of "unlock store block"
(12)	=	(20/315)	start of drum queue
(13)	=	(6/315)	alternative entry of "drum queue routine"
(14)	=	(36/314)	alternative entry of "drum transfer routine"
(16)	=	(21/314)	alternative entry of "drum transfer routine"
(17)	=	(1/327)	"preserve and restore A routine
(18)	=	(3/327)	alternative entry of "preserve and restore A"
(19)	=	(1/312)	"change page address register routine
(20)	=	(2/205)	"alternative entry of "unlock store block"
(21)	=	(3/303)	"page timers"
(22)	=	(6/203)	"page directory"
(23)	=	(2/203)	"block directory"
(24)	=	(25/314)	drum transfer working space
(25)	=	(30/314)	alternative entry of "drum transfer routine"
(26)	=	(31/303)	alternative entry of "Learning program"
(31)	=	(1/201)	"enter supervisor"

Connections with other routines

Entry At (1) with

B119 digits 22 - 12 = block number of b_1

Ba digits 22 - 12 = block number of b_2

digit 23 = instruction/operand

Exit to main program via "program scan"

16.7.62.

At (10) with

B100 = block directory entry for b_1

B102 digits 20-14 = program number P
digit 13 = 1 for duplicate
digits 12-2 = block directory position of b_1 relative to start of it's program area (BD of b_1 relative self)

B108 digits 12 - 2 = blocking directory position of b_1 relative to start of block directory (BD of b_1 relative BD)

B109 digits 11-3 = number of empty page, remaining digits zero.

At (27) with

B102 digits 20 - 14 = program number P
digit 13 = 1 for duplicate
digits 12 - 2 = BD of b_1 relative self

At (30) from R331 to copy P_1 to P_2 (P_2 empty) with

B109 digits 11 - 3 = page number P_2
remainder zero
B108 digits 12 - 2 = block directory location of P_1 relative to start of block directory
remainder zero
B102 digits 20-14 = program number of block occupying page P_1
B100 digits 11 - 3 = page number P_1
remainder zero
B106 = return address

Return with
B109 = page number P_1
B100,102 unaltered
(25/314) = page number P_2

Subroutines

(a) "Enter Supervisor"
with B96 = entry address

(b) "Lose block b"

Entry with B100 - return address
B101 digits 22 - 12 = block number b_2
digit 23 = instruction/
operand remaining digits zero

Exit with B102 digits 8 - 2 = program number P
remaining digits zero

(c) "Store location and lock out"

with B108 digits 8 - 2 = program number P
B109 digit 23 = 1 for halt requirement
digits 22-12 = block number b_1 or b_2
digits 11 = 1 for duplicate
digits 2-0 = 1 for read to cores
remaining digits zero
B110 = return address.

(d) "Unlock store block"

Entry At (3) for block on drum or not allocated with
B108 digits 20-14 = program number P
digits 12-2 = BD position relative BD
remaining digits zero
B110 = return address.

(e) "Preserve and restore A"
with B110 = return address

(f) "Change page address register"
with B108 digits 22-12 = block number
remaining digits zero
B109 digits 11-3 = page number
remaining digits zero
B110 = return address

(g) "Drum transfer routine"

Enter at (36) to find block directory position
with B102 digits 20-14 = program number P
digits 12-2 = BD position relative self
B110 = return address

Exit with B108 digits 12-2 = BD position relative BD
remaining digits zero.

(h) "Learning Program"

Entry with B109 digits 11-3 = page just read for b_2
remaining digits zero
B110 digits 23-3 = return address
digit 0 = just read
Exit with B109 digits 11-3 = chosen page
remaining digits zero

16.7.62.

Other Connections: Drum Transfer Routine

At -5 (14) to enter in drum queue
with B100 digits 20-14 = program number P
digits 12-2 = BD position for b_2 relative self.
remaining digits zero.
B101 digit 23 = operand/instruction
remaining digits zero
B102 digits 20-14 = program number P
digit 13 = 1 for duplicate
digit 12-2 = BD position of b_1 relative self
digit 0 = operand/instruction
remaining digits zero

At (21) with B100, 102, 109 as for entry at (10) to R323

At -2 (30) after copying via the accumulator
with B109 digits 11-3 = page number just allocated for b_2
remaining digits zero
B110 digits 23-3 = (30)
digit 0 = 1 for just allocated.

At (30) after reading from drum (via learning program)
B109 digits 11-3 = chosen page
remaining digits zero.

Temporary Working Space B96, B100-103, B108-110
Drum transfer working space (25/314)

Notes

1. The main program will be halted "in supervisor" if either blocks b_1 or b_2 are locked out it may also be halted by the drum queue routine and "drum transfer routine"

2. Block b_2 is lost and then "duplicate b_1" is always obeyed via the drum queue.

3. If b_1 is not allocated this routine leaves both b_1 and b_2 not allocated. If b_1 is on the drum it is read to core store to form b_2. If b_1 is in core store it is compiled to the empty page via the accumulator to form b_2.

R324: Rename block b_1 as block b_2

Purpose: To perform the extracode "rename b_1 as block b_2"

Registers of Fixed Store: 36

Number of Instructions Obeyed: 29-36

Parameters Used: (1) to (14)

Cross References:

(2)	=	(1/203)	Entry of "store location and lock out"
(5)	=	(5/203)	Block location table
(6)	=	(5/203)	Current program number
(7)	=	(5/201)	Current SER entry address
(8)	=	(3/317)	Alternative entry of "lose block b"
(9)	=	(2/203)	Block directory
(10)	=	(1/202)	"Program scan"
(11)	=	(6/203)	Page directory
(12)	=	(1/312)	"Change page address register"
(13)	=	(3/203)	Block status directory
(14)	=	(1/201)	"Enter supervisor"

Connections with Other Routines:

Entry: B119 digits 22-12 = block number b_1
remaining digits irrelevant
ba digits 22-12 = block number b_2
remaining digits irrelevant

Exit: To main program using main control, via program scan.

Subroutines: (a) Store location and lock out

Entry: B109 digits 22-12 = block number b_1 or b_2
remaining digits zero (or digit 23 = digit 0 = 1 if locked out in order to halt in supervisor)

Exit: B107 = lock out status, S
B109 = unchanged

(b) "lose block b" at (3)

Entry: after checking that b_2 is not locked out.
B100 = return address
B101 digits 22-12 = block number b_2
remaining digits zero.

(c) Change page address register

B108 digit 23 = 0
digits 22-12 = block number, b_2
remaining digits irrelevant
B109 digits 11-3 = page number
remaining digits zero.

Temporary Working Space: B96, B100-105, B108-110

Notes:

1. If either b_1 or b_2 are locked out the program is halted in supervisor.

2. If b_1 is not allocated, b_2 is lost and nothing further is done.

3. The timers for the "learning program" for b_2 are set to the timers for b_1

R327 : Preserve and restore accumulator. R327/1

Purpose : A subroutine of an SER to preserve the double length accumulator and clear it to floating point zero. An alternative entry restores the double length accumulator.

Registers of fixed store : 10

Number of instructions obeyed : 5

Parameters used : (1) to (3)

(3) = alternative entry address (see below)
(2) = address of "block location table" = (35/203)

Connections with other routines :

Enter with B110 = return address

(a) at beginning to preserve and clear A
(b) at (3) to restore A

Notes:

1. The accumulator is stored in three words beginning at full word address, 1(2)

2. The SER using this subroutine must not be halted between preserving and restoring the accumulator.

16.7.62

R328 : Duplicate write

Purpose : To perform the extracode "duplicate write"

Registers of fixed store : 8

Number of instructionsobeyed : 8

Parameter Used : (2) and (99)

(2) = (1/324) "rename block b"

(99) = extracode working space

Connection with other routines :

Entry with B119 digits 22-12 = block number b_1
digit 23 = instruction/operand
Ba digits 22 - 12 = block number b_2
remaining digits zero

Extracode (a) "rename block b"
Entry with B119 & Ba as above
B127 = return address
Exit on main control

(b) "duplicate block b to cores"
Entry by extracode instructions with
B119 digits 22 - 12 = block number b_2
Ba digits 22 - 12 = block number b_1
digit 23 = instruction/operand

Temporary Working space: one half word of extracode working space
B91 - 93

Notes:

1. The extracode (a) is entered by a direct jump on extracode control, the extracode (b) is entered from main control by the extracode instruction.

2. On exit B121 digits 8-2 = 93 and B119 = Ba

16.7.62

R329 : Remove lock down.

Purpose : To remove the lock down digit from the page directory and set the page timers T = 3, t = 0.

Registers of fixed store : 13

Number of instructions obeyed : 13

Parameters used : (1) to (7)

(1) = entry address (see below)

Cross references :

(3)	=	(12/203)	Alternative entry of "store location and lock out"
(4)	=	(35/203)	Block location table.
(5)	=	(2/203)	Block directory
(6)	=	(6/203)	Page directory
(7)	=	(8/303)	Page timers

Connections with other routines :

Entry at (1)

B104 = return address
B109 digits 22-12 = supervisor block number
remaining digits zero.

Alternative entry at 1(1)

B108 digits 8-2 = program number P
remaining digits zero
B109 digits 22-12 = block number.

Alternative entry at (2)

B104 = return address
B109 digits 11-3 = page number
remaining digits zero.

Exit :

B109 digits 11-3 = page number
remaining digits zero.

Subroutine :

Store location and lock out.

Entry :

B108 digits 8-2 = program number
remaining digits zero
B109 digits 22-12 = block number
remaining digits zero.
B110 = return address.

Temporary Working Space :

B108 - 110

Purpose: To perform the extracode instruction 1170 "Read to page P"

Registers of Fixed Store: 57

Number of Instructions Obeyed: Maximum 52

Parameters used: (1) to (24)

Cross References:

(2)	=	(35/203)	Block location table
(3)	=	(5/201)	SER Re-entry Address
(4)	=	(1/203)	Entry to "Store location and Lock Out
(5)	=	(1/315)	Entry to Drum Queue Routine
(6)	=	(20/315)	Drum queue
(7)	=	(36/314)	Alternative entry to Drum Transfer routine
(8)	=	(2/203)	Block directory
(9)	=	(6/318)	Alternative entry to Drum Queue Routine
(10)	=	(2/205)	Entry to Unlock Store Block
(11)	=	(6/203)	Page Directory
(12)	=	(1/711)	Entry to Monitor Page Locked down
(13)	=	(3/205)	Alternative entry to Unlock Store Block
(14)	=	(1/302)	Entry to Find Empty Page
(15)	=	(30/323)	Alternative entry to "Duplicate"
(16)	=	(14/314)	Alternative entry to Drum Transfer Routine
(17)	=	(1/201)	Enter Supervisor
(19)	=	(25/314)	Drum queue working space
(20)	=	(1/312)	Entry to Change Page Address Register
(21)	=	(43/314)	Alternative entry to Drum Transfer Routine
(22)	=	(3/303)	Alternative entry to Update Timers
(23)	=	(8/303)	Page Timers

Connections with Other Routines

Entry : At (1) with B119 = b digits 22-12 Remainder irrelevant
/B121\ p digits 11-3 Remainder irrelevant

Exit (a) To drum queue routine (1/315) with

B100 = p digits 11-3. Remainder zero
B101 = 0
B102 = Digit 23 = 1
Digits 20 - 14 = Program number
Digits 12 - 2 = Directory location of block b relative to start of the area in the block directory reserved for the current program
Digit 0 = 1
Remaining digits zero
B103 = Link address (18)

Entry at (18) from drum queue routine with
B104 = Address of current entry in drum queue
B103 = Block specification (as B102 above)

Exit (b) To next drum transfer, entering R315 at (6), via R205, if block b already in page p

(c) To "monitor page locked down" if page p locked down, with
B109 = Page number p digits 11-3. Remainder zero
B107 = Program number P digits 8-2
Block b is unlocked and the program made free if it was halted for block b before exit.

(d) To drum transfer routine R314 at 3(14) if block b on drum, with
B104 = Address of current entry in drum queue
B109 = page number p digits 11-3. Remainder zero.
Page p is empty at this stage, but the page directory and page address register are <u>not</u> reset.

(e) To make empty page and continue to next drum transfer if b in cores, with subsidiary store 0.4 (25/314) holding page number of page originally occupied by block b.

<u>Subroutines</u>:

"Enter Supervisor" : Entered at (1) with B96 = 2(1/331)
Exit to this address

"Store location and Lock Out" : Entered at (1) with
B110 : Return address
B109 : Digit 23=1, digits 22-12 = b, digit 0=1 remainder zero
Return with B108 : Digit 23=1, digits 20-14 = Current program No.,
digits 12-2 = block directory location of b relative to start of area reserved for current program
Remaining digits zero.
<u>or</u> to program scan
<u>or</u> to halt main program in supervisor for block b

"Drum transfer routine" : Entered at (36) to find block directory location of block b
with B102 as for exit (a)
B110 return address
Return with B108 = block directory location relative to the start of the block directory digits 12-2.
Remaining digits zero.

"Unlock Store Block" : Entered at (2) to unlock block in cores with
B109 = Page number P or page number of block digits 11-3.
B110 = Return address
Entered at (3) to unlock block on drum with
B108 = Program number digits 20-14
Location of b in block directory relative to start of block directory, digits 12-2
B110 = Return address
Return with B109 unaltered.

16.7.62

"Duplicate" : Entered at (30/323) to copy from one page to another

a) To copy P to an empty page P' if P is not empty

Entered with B109 = page number P' digits 11-3
remainder zero.
B108 = block directory location of P
relative to the start of the block
directory, digits 12-2. Rest zero.
B102 = program number of block occupying
page P digits 20-14.
Rest irrelevant.
B110 = page number P digits 11-3. Rest zero
B106 = return address

Return to return address with
B109 = page number P
B100, 102 unaltered
Block directory of page P updated, now referring to P'

b) To copy b to empty page P.

Entered with B109 = page number P digits 11-3 rest zero
B108 = block directory location of b
B102 = program number of b
B100 = page number of b
Return to return address with
B109 = page number of b
(25/314) = page number P

"Find Empty Page": Entered with B110 = Return address
B108 digit 0 = 1 (any stack)
Return to return address with B109 = page number P'
of empty page, digits 11-3
B100, 102, 104 unaltered.

"Update Timers": Entered at P(3/303) with B110 = Return address
Return with B104 unaltered.

Notes:

1. The program will be halted if block b is already locked out; if the block is not locked out already, it is locked out with S = 1 (read to cores). The program may then be halted if the drum queue is full.

2. An entry is always made to the drum queue, irrespective of the initial location of block b.

3. When the item is obeyed in the drum queue the following action occur

   a) Immediate exit if b already in page P
   b) Exit to monitor if page P is locked down
   c) If page P is not empty, it is copied to an empty page via the accumulator
   d) If block b is on the drum, it is read to page P
   e) If block b is in core store it is copied to page P via the accumulator and the previous page occupied is made empty.

4. If action 3(c) occurs, the page address register of the previous empty page is set to *4, whatever the initial contents of the P.A.R. of page P. The page directory of P is copied across.

5. The page timers are updated when the entry is obeyed from the drum queue. When blocks are copies to or from page P in core store, the timers T are copied, and t = 0 for the new block.

R332 : Clear blocks.

Purpose : To set the "clear blocks" switch.

Registers of fixed store : 5

Number of instructions obeyed : 5

Parameters Used : (1), (2) and (3)

Cross References :

(2) = (5/203) program number
(3) = (4/203) program store directory

Connection with Other Routines:

Entered at (1) from extracode 1167 with

B119 digit 0 = 0 Clear store blocks not required
= 1 Clear store blocks required

Remaining digits irrelevant.

R333: Write/Release block b

Purpose: To perform extracode "Write block b to drum" and "Release block b" by entering an item to the drum queue or setting a digit in the page timers for the "learning program".

Registers of Fixed Store: 41

Number of Instructions Obeyed: 15 to 58

Parameters Used: (1) to (19)

Cross References:

(2)	=	(1/201)	Enter Supervisor
(3)	=	(5/201)	SER Entry Address
(4)	=	(1/203)	Store location and lock out
(5)	=	(35/203)	Block location table
(6)	=	(2/203)	Block directory
(7)	=	(3/203)	Block timers/status directory
(8)	=	(1/202)	Program Scan
(9)	=	(4/315)	Number of further entries in drum queue
(10)	=	(1/315)	Entry of Drum Queue Routine
(11)	=	(2/304)	Alternative entry to learning program
(12)	=	(1/205)	Unlock block in cores
(13)	=	(36/314)	Alternative entry to Drum Transfer routine
(14)	=	(6/315)	Alternative entry to Drum Queue routine
(15)	=	(3/205)	Unlock block on drum
(16)	=	(9/313)	Number of drums
(17)	=	(2/313)	Table of empty sectors
(18)	=	(44/314)	Alternative entry to Drum Transfer routine

Connections with Other Routines

Entry at (1) with B119 = b digits 22-12. Remaining digits irrelevant
B126 digit 0 = 0 (write), 1 (release)

Exit (a) to program scan if block on drum

(b) to "unlock store blocks" at (2/205) with
B109 = page No. of block b digits 11-3 remainder zero
B110 = program scan
if realease or dum queue full.

(c) to "Drum Queue Routine" at (1/315) otherwise with
B101 = 0
B102: Digit 23 = 1
Digits 20 - 14 = Program number P
Digits 12 - 2 = Block directory location of b relative to the start of the area reserved for program P
Remaining digits zero.
B103 = Return address (19/333)

16.7.62

Entry at (19) from drum queue routine with

B104 = Address of current entry in drum queue
B102 as on exit to drum queue routine.

Exit (d) to "Unlock Store Block" at (3/205) if block on drum with
B108 = Program number P digits 20-14
Location of b in block directory relative to the start of this directory. Digit 12-2.
B110 = Alternative entry to drum queue routine (6/315)

(e) to (6/315) after setting write digit if there is no empty sector.

(f) to drum transfer routine at (44/314) otherwise with block unlock and B109 = Page number of block (digits 11-3) via learning program.

Subroutines:

"Store location and lock out"

Entry at (1/203) with
B109 = block label b digits 22-12
digits 1 = 1 (L = 2)
remaining digits zero
B110 = Return address
Return with B108 = Program number, digits 20-14
Block directory location relative to the start of the area reserved for this program, digits 12-2
digit 23 = 1
Remaining digits zero.

The program may be halted in supervisor, and exit may be made to program scan.
"Drum transfer routine" to find block location.

Entry at (36/314) with
B102 as in exit (c)
B110 = Return address

Exit with B108 = location in block directory relative to the start.

"Unlock store blocks"

Entry at 1/205 with B109 = Page number
B110 = Return address

Exit with B109 preserved.

Temporary Working Space: B96, 100-103, 105, 108-110

Notes:

1. If the block is on the drum, nothing is done.

2. If "release" or if the drum queue is full, no entry is made to the drum queue, but the "Write to drum" digit in the Page Timers is set to 1.

3. When the item is dealt with in the drum queue, no transfer takes place if the block is on the drum. If there is no empty sector, the write to drum digit is set to 1. Otherwise the drum transfer routine is entered to write this page to the drum, after unlocking the block.

4. If the block has not previously been defined, it becomes defined but unallocated to a sector or page.

5. The "write to drum" digit is set by entry to R205 at (1) with lock out status S = 2.

6. The block timer Y is reset when the block is written to the drum by a special entry to R304 without first updating the timers, and may therefore be in error.

R350: Locate next program store block

Purpose: A supervisor routine in main store to implement extracode"ba' = next block label defined $\geqslant$ n". An alternative entry allows for use as a supervisor subroutine.

Registers of Main Store: 20

Instructions obeyed: 9 + 5 to 10 per store block reserved

Parameters Used: (1) to (7)

Cross references:

(4)	=	(5/203)	Program number
(5)	=	(4/203)	Program store directory
(6)	=	(2/203)	Block directory
(7)	=	(4/247)	Return to Main Control

Connections with other Routines

Entered at (1) from extracode via (1/247) with full recovery switch set.
B119 = block label n digits 22-12. Remainder irrelevant
Exit to (4/247) to return with
B91 = next block label $\geqslant$ n, digits 22-12
B91 = *4 if no next block < *3400
Alternative entry at (2) as a subroutine
B100 = n digit 22-12. Digit 0 = 1
B101 = Return address
Exit to return address with B91 = next block label $>$ n, digits 22-12 or *4
(note block label n ignored)

Temporary working space: B91, B100, B107-110, Bt

Notes:

1. When used as a subroutine, it is assumed that the calling routine is based on the current object program and that the full recovery switch is set.

2. Any block label in the supervisor range ($\geqslant$ *34) is neglected.