I.C.T. ATLAS COMPUTER

VOLUME 2

ROUTINE 400 ONWARDS

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).

This document is confidential and may not be given or lent to any person other than addressee without the authority of I.C.T. Ltd.

CONTENTS /3 (VOL. 2)
MAGNETIC TAPE ROUTINES

PERIPHERAL ROUTINES

MAGNETIC TAPE ROUTINES - ABBREVIATIONS

PBA Present Block Address, read from the magnetic tape V store

EBA Expected Block Address, used to check the PBA

WBA Wanted Block Address, used during search orders

SBA Stop Block Address, used to check if the tape has stopped

LBA Leading Block Address (always non zero)

TBA Trailing Block Address (always zero)

CU Current Order, or order being processed

FO Following Order, or order waiting to be processed when the CU is completed and the store is ready

DD Deck Directory, a flip flop register

ED Error Directory, a flip flop register for tape errors and faults

TCR Tape Control Register

TQ Tape Queue

Qm Location of most recent TQ entry for this channel.
m is contained in the DD

Qp Location next TQ entry to be processed for this channel.
p is contained in the CU.

1/10/63

R400: Tape Flip Flops in Subsidiary Store

The following two flip flops are contained in the CU locations:

<u>CU not complete flip flop</u>. [illegible]

Set when a tape order is placed in a CU location.
Reset when the order has been successfully completed.

<u>Supervisor exit flip flop</u> [illegible]

Set by the main supervisor.
Reset on successful completion of the order, on entry to R217.

The following two flip flops, in the DD location, are for the purpose of ensuring that the total number of possible long interrupts, which may have to be entered in an SER queue is not more than two entries per channel:

<u>Clear last flip flop</u>

Set when a tape order has been successfully completed, but the tape has not been stopped. Reset by the clear last long interrupt routine.

<u>Prepare next flip flop</u>

Set, if the clear last flip flop is set, when testing whether to enter the prepare next routine. Reset by the clear last long interrupt routine.

The following two flip flops in the DD locations are to test either whether to initiate the next tape order and start the tape after coming back from the Organise store subroutine R414, or whether to initiate the next tape order as soon as the current order has been successfully completed, without stopping the tape.

<u>Tape in use flip flop</u>

Set when the tape is started moving.
Reset when the tape has stopped, after successfully completing a tape order.

<u>Store ready flip flop</u>

Set when the store is ready for the next tape order.
Reset when the next tape order is initiated.

R400: continued

The following two flip flops are for the purpose of ensuring that if a LBA, forwards, or a TBA, reverse, is encountered, then another BA interrupt should occur within 0.1 sec:-

F1: Set at a LBA, forwards, or TBA, reverse

Reset at a TBA, forwards, or LBA, reverse

F2: Set by the clock interrupt, if F1 is set

Reset at a TBA, forwards, or LBA, reverse

Note that if F1 and F2 are both set when a clock interrupt occurs then the transfer is ended and control of the tape passes to the monitor. This, however, does not apply to Orion tape.

Location F3

This location is used to record the clock reading of the time when either a start tape order or a stop tape order is written to the TCR. It is reset as soon as a BA interrupt occurs. If the clock interrupt routine finds that F3 has not been reset after a certain interval of time, it is assumed either that the tape has failed to respond to start or stop order, or that BA interrupts are failing to register.

Tape Counters in Subsidiary Store

CE: This counter is used after slowing down a long search from fast speed to normal speed. It counts consequentively sequenced LBAs, in order to determine when normal BA interrupts are assumed to occur. The counter is also used when aligning the tape to the EBA. It counts the number of changes of direction of tape movement required to align the tape to EBA = PBA.

CT: Number of seconds on a fast speed search before it has to be slowed down to normal speed.

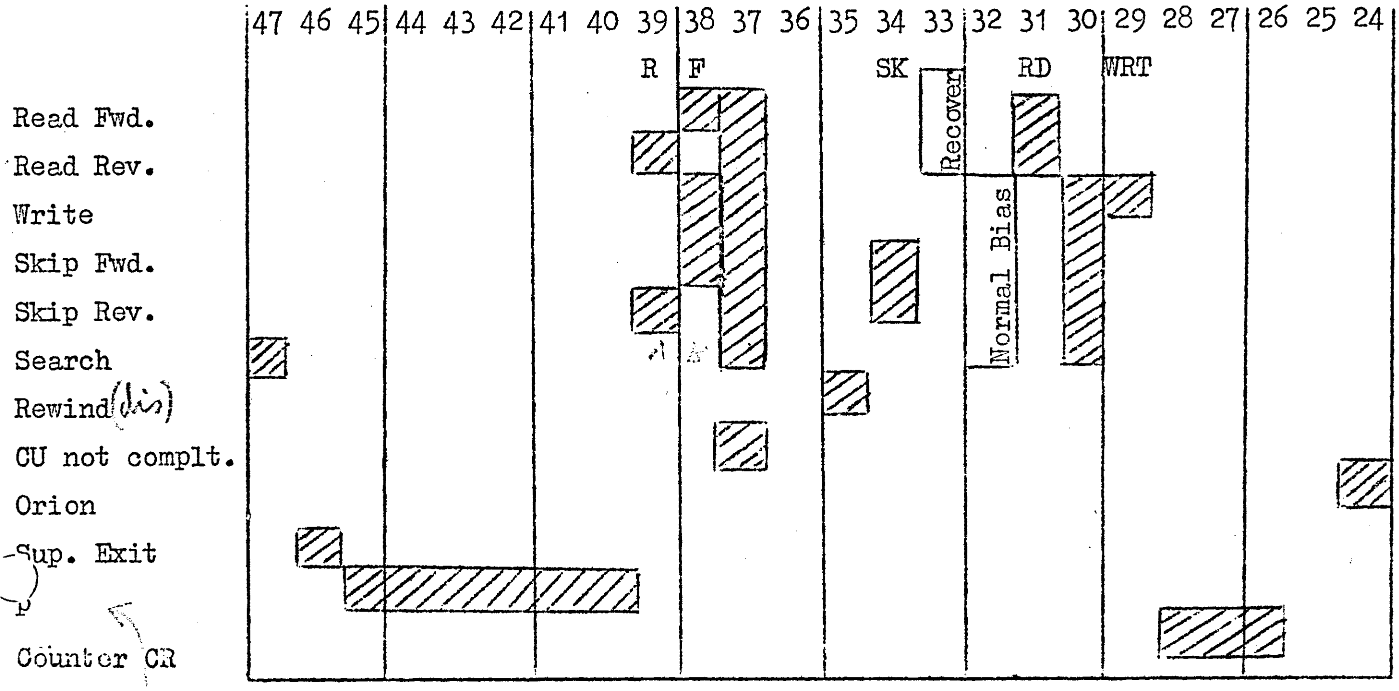
CQ: Number of blank spaces in the tape queue. Note that this counter must be preset, probably at 15.4. Otherwise, no tape order can be processed.

CR: This counter is contained in the CU location. It counts the number of times the current order has been repeated, due to error conditions.

R400: Digits in Tape Working Store (continued)

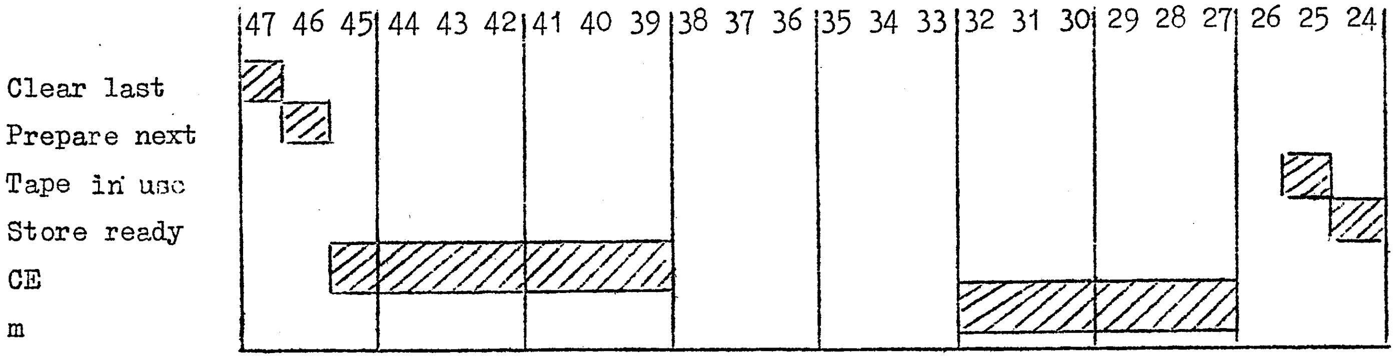
A [illegible] R [illegible]

Tape Order (Qp, FO, CU)



always in stored order
link to next unit in Tape Q. Lifted from CU

Deck Directory (DD)



R400: Digits in Tape Working Store (continued)

Error Directory (ED)

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write Crisis								▨												▨				
Read Crisis								▨										▨						
Parity 3								▨								▨								
Parity 6														▨										
Checksum																								▨
512 Words																							▨	
Write not reset																			▨					
Read not reset																	▨							
LBA Error						▨																		
TBA Error					▨																			
Beg. Tape				▨																				
End. Tape			▨																					
Dir. fault									▨															
Spd. fault							▨																	
Read Bias fault															▨									
Deck Fail Reset																					▨			
Non Equivalence													▨											
Counters										▨	▨	▨												

R401: B.A. Interrupt Entry

Whenever a B.A. interrupt occurs from magnetic tape this routine is entered. It stores the following information from the subsidiary store and the tape V-store into B registers: PBA, EBA, TCR, CU, DD. It also resets location F3 and the BA interrupt LAM. The routine then transfers control to an interrupt routine, which has been preselected by the supervisor or monitor programs. The main B.A. interrupt routines include the following:

1. R402 B.A. interrupt routine
2. R403 Tape stopped interrupt routine
3. R404 Alignment of tape to EBA
4. R405 Calculation of EBA after an off channel search
5. R500 (Exit) if BA interrupts are to be ignored

Registers of fixed store	10
Instructions obeyed	10
B registers used	111-116, 123

R402: B.A. Interrupt Routine

The B.A. Interrupt routine deals with the B.A. interruptsas they occur from a magnetic tape, which is moving under normal operation. It is entered from the B.A. interrupt entry routine R401 at one of two entry locations. One entry corresponds to the LBA, and the other entry corresponds to the TBA

The routine accomplishes the following:

1. Checks the PBA = EBA. If there is any discrepancy, the routine records an error in the error directory, and stops the tape.

2. At a LBA, forwards, or TBA, reverse, it sets flip flop bit F1 to indicate that another BA interrupt is expected within 0.1 sec. At a TBA, forwards or LBA, reverse, it resets flip flop bit F1. It also resets flip flop bit F2, which has been set by the clock interrupt. Note that if both F1 and F2 are set when a clock interrupt occurs, it indicates that a BA interrupt has failed to register. If this should occur, the transfer is ended and control of the tape is passed to the monitor. This, however, does not apply to Orion tape.

3. At a LBA, forwards, or TBA, reverse, the CU is tested. If the CU is a write order, then the "Write" digit is set in the TCR. If the CU is a read order, the TCR is tested to ensure that the "Start Read at next BA" digit is set. If it is set, then the "End Read at Next BA" digit is written to the TCR, otherwise, an error is recorded in the error directory and the transfer is ended, but the tape proceeds to the next B.A. interrupt before further action is taken. If the CU is a search, the WBA is compared with the EBA, and action to proceed, or stop the tape is taken accordingly.

4. At a TBA forwards, or LBA, reverse, the TCR is checked for certain error conditions, such as check sum error, not 512 words in a block, write digit not reset, according to the order that has just been completed. Also the parity 3 and parity 6 errors are examined. Any errors which are indicated are recorded in the error directory, and the tape is stopped.

5. At a TBA, forwards, or LBA, reverse, if the tape order just completed was a read or write, but not on Orion tape read order, and no errors were found, then the corresponding entry in the PAR is altered to prevent it from interfering with any following tape transfer order.

6. At TBA, forwards, or LBA reverse, if the next tape order is ready then it is moved into position to be processed, and the corresponding entry in the PAR is set to enable the page to accept the transfer. Hence, successive tape orders can be processed without stopping the tape. This, however, is not done when the tape either is in monitor control, or is an Orion tape.

R402: continued

7. At a TBA, forwards, or LBA, reverse, either the routine exits to the clear last long interrupt, R412, if the previous tape order was a read or write order and the tape is proceeding to the next tape order without stopping, or the routine exits to main control.

8. At a LBA, forwards, or TBA reverse, either the routine exits to the prepare next long interrupt, R411 if there is space in the SER tape queue, or the routine exits to main control, setting a flip flop bit in the DD to indicate that as soon as there is space in the SER tape queue, the prepare next long interrupt can be entered.

9. If the tape has to be stopped, then the routine writes the clock reading into F3 location, to indicate that a stop B.A. interrupt is expected within a certain period of time.

Entry R401 BA Interrupt entry

Exit 1. R411 Prepare next long interrupt

2. R412 Clear last long interrupt

3. Main control

Fixed store registers 124

Instructions obeyed varies from 15 to 50 in normal operation

B Registers used 111-117

R403: Tape Stopped Interrupt Routine

The tape stopped interrupt routine is entered from the BA interrupt entry R401 at the next BA interrupt after a stop order has been given to the tape. To determine whether the tape has stopped it compares the PBA with the value stored from the previous BA interrupt. If there is no change, it is assumed that the tape has stopped.

If it is assumed that the tape has stopped, the routine then enters the long interrupt start tape routine R413. This does not apply, however, if the tape is endeavouring to align itself to stop before EBA=PBA. In this case, the tape is either restarted again moving in the opposite direction, or the monitor routine is entered, depending on whether the tape has been restarted less than or more than five times, respectively.

If it is indicated that the tape has not stopped, the routine checks to determine if the TCR is set to stop, and informs the operator accordingly. BA interrupts, which may occur after the tape has indicated that it has stopped, are ignored.

Entry:		R401	BA Interrupt entry
Exit:	1.	R419	Start tape long interrupt
	2.	-	Failed to find EBA after 5 attempts
	3.	-	Failed to stop
	4.	-	TCR not set to stop
	5.		Main control

Number of fixed store registers	30
Maximum number of instruction obeyed	16
B register used	111-116, 123

R404: Alignment of Tape to E.B.A.

This B.A. interrupt routine is used in conjunction with the tape stopped interrupt routine by the main store monitor program when it is desired to align the tape to stop just before PBA = EBA, where the EBA remains unchanged.

Successive PBA's are checked at each BA interrupt until one is found equal to the EBA, when the tape is moving in reverse. The tape is then stopped. This search for the required PBA is repeated, if necessary, up to five times before indicating that the required PBA cannot be found.

Entry: R401 BA interrupt entry

Exits: 1. (5/402) of BA interrupt routine

2. Main

Number of fixed store registers 17

Maximum number of instructions obeyed 12

B register used 111-116, 123

R405: Calculation of E.B.A.

This BA interrupt routine is used after the tape has been brought back to normal speed from fast speed towards the end of a long off channel search. It determines when BA interrupts occur normally, and calculates the EBA.

The value of the EBA is taken as the value of a certain number of consecutively sequenced, and checked PBA's. After having determined this value for the EBA, then BA interrupts will normally be processed by routine R402.

Entry: R401 BA interrupt entry

Exit Main control

Number of fixed store registers 24

Maximum number of instructions obeyed 10

B registers 111-116, 123

R406: Deck Failure Interrupt Routine

The deck failure interrupt routine identifies whether the deck failure was caused by a read or write crisis, or by the tape being stopped on the metal backing, or by a mechanism failure. It then deals with the interrupt as follows:

If the interrupt was caused by a read or write crisis, the routine exits to (1/407). Otherwise it writes "End Transfer" to the TCR of the effected channel, resets the timers and tries to reset the interrupt.

If the interrupt cannot be reset, the tape is stopped and disengaged, and the operator is informed. There should be no further BA interrupts, but if any occur, they will be ignored.

If the interrupt has been reset, and it is found that the metal backing digit is set in the TCR, then the routine exits to (1/416), the metal backing long interrupt routine.

If the interrupt has been reset, and the tape is not at the metal backing, then the tape will not be stopped. An error mark will be written to the ED of the channel concerned, and the tape continues moving until the next BA interrupt occurs, before further action is taken.

Entry: from deck failure interrupt

Exits: 1. 1/407 Read and Write Crisis

2. 1/416 Metal backing long interrupt

3. R419 Deck failure cannot be reset

4. Main

Number of registers of fixed store 34

B registers used 111-113, and 123

R407: Parity 3 & 6 Read & Write Crisis Time Interrupt

Parity 3 interrupt occurs if an incorrect parity is detected in the core store page in the directory, when reading from tape, or writing to tape. There is no indication given as to which channel was affected.

The routine, therefore, has to examine the TCRs of all channels to determine which are processing read or write orders. If a channel is processing a read or write order, the transfer is ended, and an indication is made in its respective ED that there has been a parity 3 fault. If a channel is not processing a read or write order, and indication is made to its respective ED to stop the tape from processing further tape orders for the duration of the parity 3 monitor program, which will be called in after all the tapes have been stopped.

The read and write crisis interrupts occur if the word transfer between the tape co-ordinator and the central computer is not met within the crisis time of approximately 13 microseconds. This causes a deck failure interrupt, which transfers control to this routine. It is treated in the same manner as a parity 3 interrupt.

Parity 6 interrupt occurs if an incorrect parity is detected in the tape co-ordinator when writing to tape. The buffer parity fault digit is set in the TCR of the effected channel. The routine examines the TCRs of all channels, and if any channel has the buffer parity fault digit set, its transfer is ended, and an indication is made in its respective ED that there has been a parity 6 fault. The other channels remain uneffected.

The parity 3 and parity 6 interrupt routines are combined into one routine, but they have different entry points as indicated below. The routine ends by resetting the interrupt in the V store and returning control to main. The BA. interrupt routine will then identify and deal with the fault accordingly.

Entry: 1. Parity 3 interrupts

2. Parity 6 interrupts

3. Deck failure interrupt

Exit: Main

Registers of fixed store	30
Instruction obeyed	76
B registers	111-116, 123

R411: Prepare Next Tape Order

The prepare next tape order routine is either entered as a long interrupt from the first BA interrupt after initiating a tape order, or it is entered as an extension of the basic order to tape queue routine, R421. Its purpose is to determine if the next tape order for the channel concerned can be updated and then initiated without stopping the tape on completion of the current tape order. This can be done if the tape order does not involve a change of direction, nor is it a search, nor a rewind, nor an Orion tape order, nor a supervisor order. If none of these conditions apply then the order is moved into the FO position and the organise store subroutine, R414, if required, prepares the corresponding page of core store to accept the transfer. If the order is a composite order; that is, more than one block is involved, then only one part of it is moved to the FO position. When the next BA interrupt occurs, if the organise store routine has prepared the corresponding page of core store and if the current tape order has been successfully completed, then the tape order in the FO location can be initiated by the BA interrupt routine without stopping the tape.

Parts of this routine are common with routines R412 and R413

Entry:	R402	BA interrupt routine
	R421	Basic order to tape queue
	R412	Clear last tape order
	R413	Start tape routine
Exit:	R202	Main
Subroutine Used:	R214	Free program
	R414	Organise store blocks

Registers of fixed store: 72

B registers used 100-110, Bt

R412: Clear last tape order

The clear last routine is entered as a long interrupt from the BA interrupt routine after the completion of a read or write order, if the next tape order has been initiated without stopping the tape. The purpose is to remove the lockout and lockdown digits from the page involved during the previous tape transfer.

Entry:	R402	BA interrupt routine
Exit:	R411	Prepare next tape order
	R202	Main

Subroutines Used: R205 unlock store block

Registers of fixed store	14
B registers used	100-110

R413: Start Tape Routine

The start tape routine, R413, is a preselected routine entered either as a long interrupt from the tape stopped interrupt routine, R403, or as an extension of the placing of a tape extracode into the tapes queue routine, R421. There are three parts to this routine: the clearing up of the previous tape order, the updating of the next tape order, and the initiation of a tape order by writing to the TCR.

The part concerned with the clearing up of the previous tape order begins by testing the ED and if there are any errors recorded in it, the routine exits to R419.

Next, the current order just completed is examined, and if it has not been marked as having been completed successfully, then it will be initiated again.

If it has been marked as complete, and it was a read or write order, then the lockout and lockdown digits from the PAR of the pages involved in the transfer will be removed before proceeding. Also, if the order was marked as a supervisor order, then subroutine R217 will be entered before proceeding to update the next tape order.

The updating part of this routine begins by examining if there is a tape order waiting to be processed in the FO location. If there is, then the order is either moved to the CU location and initiated, if the core store pages concerned are ready for the transfer, or the routine will exit to main control, and the order will be initiated when the store is ready.

If there is no tape order waiting to be initiated, the next order in the tape queue for the channel concerned is extracted and either moved to the FO location, if it is a read or write order, where it will wait until the corresponding pages of core store are prepared, or moved directly to the CU location, if it is a search or skip or rewind order, and initiated.

If there are no further entries in the tape queue for the channel concerned, the routine exits to main control.

If the tape queue should be marked as full, and the updated entry has left a space in the queue, then the tape queue full indication will be removed, and those programs which have been held up because of this will be freed.

The initiating part of the routine writes the necessary digits to the TCR, corresponding to the CU, and then reads them back again to ensure that they have been written correctly, before starting the tape. Also on starting the tape, the clock reading is written to location F3.

R413: continued

Except in the case of search and rewind orders, all tape orders are initiated by extracting the corresponding digits from the CU location and writing them directly to the TCR.

In the case of a search order, the direction in which the tape has to move is determined from a comparison of the WBA with the EBA, and its initial speed of motion is determined from the distance between the WBA and the EBA. If this distance is greater than 200 blocks, the tape is started at fast speed, if not, it is started at normal speed. The length of time, which a tape is to remain on fast speed during a search is calculated, and stored to the nearest second in counter CT, allowing one second for every 23 blocks beyond 200 blocks.

In the case of a rewind order, the following digits are written directly to the TCR: end transfer, end read, normal read bias, fast speed, start, reverse, disengage.

Entry: R403 Tape stopped interrupt

R421 Basic instruction to tape queue

Exit: Main control

Three times repeat monitor R419

TCR read back incorrect before start, machine monitor

Write permit not present on a write order, program monitor

Subroutine Used: R217 Enter SEC

R214 Free Program

R205 Unlock store block

R411 Prepare next tape order

Registers of fixed store 118

B register used: 100-110

R414: Organise store blocks for tape

Purpose: An SER to call blocks to core store for transfers to or from magnetic tapes and to set the page address registers for a tape transfer.

Register of fixed store: 62

Instructions obeyed: Supervisor block : 33 + entry to (1/318) + entry to R312
Program block : 45-56 + entry to (2/318) + entry to R312

Multiple program blocks : Around n times one program block for n blocks

Parameters used: (1) to (19)

Cross references:

(2)	=	(5/201)	SER re-entry
(3)	=	(7/201)	SER base
(4)	=	(52/400)	Following tape order (type spec)
(5)	=	0.4(52/400)	Following tape order (store spec)
(6)	=	(53/400)	Deck directory
(7)	=	(35/203)	Block location table
(8)	=	(2/203)	Block directory
(9)	=	(1/318)	Call to cores
(10)	=	(2/318)	Call to cores
(11)	=	(36/314)	Find directory entry
(12)	=	(32/314)	Lose sector
(13)	=	(1/312)	Set PAR
(14)	=	(1/218)	Step directory reference
(19)	=	(15/411)	Return address

Connections with other routines

Entered at (1) from tape routine with B100 = channel no. (digits 5-3) rest zero
Exit to (19/414) with B100 = Channel no.
Block (s) locked down in core store. Page address registers set for channel, forward or reverse, if tape stopped.
Otherwise, digit 14 inverted.
Page number of last block transferred in digits 23-3 of (52/400), digits 2-0 unaltered

Subroutines:

a) "Call to cores" (i) Entered at (1/318) if supervisor block with
B109 = block label (22-12) digits 23, 10, 9, 0 = 1
(lock down, operand, no timer)
B110 = return address.

(ii) Entered at (2/318) if block of object program with
B108 = Directory location relative to start of program, p 12-2. Program number p20-14
Digits 22,0 = 1 (lock down operand)
Rest zero
B110 = Return address
Block location table = BD position relative to start of BD

Exit to resume at re-entry address with B100 preserved if block on drum or drum queue full
Exit to return address when in core store with B109 = Page no., B100-104 preserved.

b) Find directory entry:
Entered at (36/314) with
B102 = Prog. no. (20-14); B.D. entry relative to program start (12-2)
B110 = Return address
Exit to return address with B107 = Program number (8-2)
B108 = Directory entry relative to start of BD

c) Lose sector:
Entered at (32/314) with
B100 = sector number, digits 11-1
B110 = return address
Exit to return address with B109, B101-104 unaltered

d) Set page address register:
Entered at (1/312) with
B108 = New contents of PAR, digits 23-12
B109 = Page no. digits 10-3
B110 = Return address
Exit to return address with B109 unaltered

e) Step directory reference:
Entered at (1/218) to step back at 1.1 (1/218) to step forwards with B107 = Prog. no. (20-14)
BD entry relative to start of program area (12-2)
Remainder irrelevant
B110 = Return address
Return via R203 with
B108 = Prog. no. (20-14)
BD entry of next block relative to start of program area (12-2)
p23 = 1
Rest zero
B105-109 altered.

Temporary working space: B101 - B110, B_t

R414: continued

Notes:

1. On entry to this routine, the "following" entry in the tape queue for the channel holds either.

   b (p22-12), rest zero if a supervisor block. Otherwise program no. (p21-15). Block directory location relative to start of area for program (p13-3), No. of blocks less 1 (p2-0) (always zero for Atlas) p23 = 1.

   On exit the contents are changed to page number (p23-3), p2-0 unaltered. No other tape directory is altered by this routine.

2. If the block is required for a read transfer from Atlas tape and is on the drum, it is "lost" before being called to core store, thus avoiding one drum transfer. This does not apply for an Orion transfer, which may not use the whole block.

3. If the block is a supervisor block, the "dont change timer" digit is set in the Page Directory, irrespective of the block label.

4. The P.A.R. are set as follows for channel n

Atlas	tape,	forward,	tape stopped	:	*77n0	(Orion	*77n1 etc)
"	"	"	tape moving	:	*77n4	("	*77n5 *77n6)
"	"	backward,	tape stopped	:	*77n7	("	*77n6 *77n5)
"	"	"	tape moving	:	*77n3	("	*77n2 *77n1)

   P.A.R. contents are also set to these values.
   Digit 1 of the deck directory reads 0 if the tape is stopped, 1 if moving.

5. The current SER base is set to 0.1 by this routine, and the re-entry address is altered.

R416: Metal Backing Routine

The metal backing routine is a long interrupt routine entered from the deck failure routine, R406, if the interrupt was caused by the tape stopping on the metal backing at the beginning or at the end of the tape.

If the tape, which has caused this interrupt is on channel 7 and is being addressed, then the routine exits to the addressing routine.

If the tape is in the process of a search order, and this interrupt occurs, then the deck timer is set for two seconds, and the routine exits to main. The tape will be restarted again moving in the opposite direction, from one to two seconds later, by the one second clock interrupt routine. The first expected BA interrupt will be preselected as a LBA.

If neither of the above cases has caused this routine to be entered, then it is assumed that there has been a mechanism failure, and hence the routine will disengage the tape and inform the operator accordingly.

Entry: Deck failure interrupt routine R406

Exits: 1. Addressing routine

2. Main control

3. Mechanism failure

Registers of store: 15

B registers used: 100-102

R419: Tape Error Repeat Routine

Whenever the start tape routine, R413, is entered, the ED of the channel concerned is examined. If there are any errors or faults recorded in it, control is transferred to the tape error repeat routine. This is a main store monitor routine. It examines the digits, which are set in the ED, one at a time, to identify the type of error or fault which has occured, and then either exits to a monitor routine, or initiates a procedure which will repeat the current order to see if the error can be cleared.

The deck faults which cause this routine to exit directly to the machine monitor R400, include the following:

1. Write digit not reset in the TCR after a write transfer (F7)

2. Read next BA not set at the first BA interrupt at the beginning of a read transfer, or not reset at the end of the transfer (F8)

3. Failure to clear a tape error after several repeats of the current order (F9).

The program faults which cause this routine to exit directly to the program monitor include the following:

1. Beginning of tape

   If the tape has been stopped because the program has referred to block 0, then this routine will space the tape forward to stop between the TBA of block 0, and the LBA of block 1, before entering the program monitor. However, if the supervisor wishes to refer to block 0, it can do so by setting EBA≯1 at this time.

2. End of tape

   If the tape has been stopped because the program has referred to block 5,000 then this routine will space the tape backwards to stop just before the LBA of block 5,000 before entering the program monitor. However, if the CU is a search order, the routine will not enter the monitor, but instead the WBA will be compared with the last EBA, to determine if the block is on the tape. If not it indicates that it is a short tape.

3. Thirteenth bit set in EBA

   This bit may have been set by the supervisor in order to stop the tape at the next LBA. In this case, the EBA will be made equal to the PBA, and the tape restarted.

R419: continued

The deck errors which cause the current order to be repeated include the following, which are tested in this sequence:

1. LBA error
2. TBA error
3. Checksum failure
4. Not 512 words transferred
5. Deck failure has occured and was reset immediately
6. Parity 6

The repeat process is accomplished by altering the EBA so that the tape can be realigned to stop just before the block where the error had occured, by the alignment of tape EBA routine, R404. The ED is cleared, and the tape restarted. When the tape has been realigned, the start tape routine, R413, will automatically re-initiate the order.

A three bit counter in the CU location determines how many times the CU has been repeated. Each time it is repeated, this counter is incremented. If it should reach a certain maximum, say 7 (it can be altered to any number up to 7), then the routine exits to the machine monitor R400, to indicate that the error has not cleared itself.

Deck errors which occur in block 0 are not repeated, but instead the routine exits directly to the monitor.

Other tape errors recorded in the ED, which cause this routine to exit to special routines, include the parity 3 error and the read and write crisis errors.

Entry: R413 Start Tape

Exit: Machine monitor

Program monitor

Main control

Registers of store: 96

B registers used: 100-105, 109

R421: Basic Order to Tape Queue

This routine finds the deck number, allocates and locks out the associated store blocks if necessary, of a basic tape order and then places the order in the tape queue linking it with the previous order for the same channel. If the tape queue should be full when the order is given, then the program affected will be halted.

The basic order to tape queue routine is entered directly from the extracode vector of basic tape orders. There are nine of these orders: search, read forward Atlas tape, read forward Orion tape, read reverse Atlas tape, read reverse Orion tape, write Atlas tape, skip forwards, skip reverse, and rewind. These orders can be simple or composite. A simple order involves only one block from tape. A composite order involves up to eight blocks from tape.

If the order should be a search, a test is made to check if the WBA is on the tape. If not, the program is monitored. Also, if the tape is in variable length mode, the variable length operations are ended before starting the search.

The order is then placed in the tape queue together with its block directory entry or WBA, according to the type of order involved, linked to the previous order for the same channel.

The routine exits either to prepare next long interrupt, or to start tape long interrupt, depending on whether the tape should be marked in the DD as in use or not, respectively.

Entry:	Transfer vector of basic tape extracodes	
Exit:	R411	Prepare next tape order
	R413	Start tape routine
	R213	Tape queue full
Subroutine Used	R221	Find deck number
	R203	Store location and lock out
Register of fixed store	100	
B register used	100-110, Bt, 91, 92, 96, 97	

Addressing and Readdressing Routine for Atlas I Magnetic Tape

Purpose

To test and address magnetic tape in accordance with the format laid out in the Atlas I General Description and to remove, from existing addressed tapes, any blocks found to be faulty.

Method of Use.

1. Addressing:-
Punch a steering tape as follows:-

```
JOB
VAS - ADDRESS TAPE

COMPILER TAD

A
n

***Z
```

Where n is the maximum number of blocks to be marked out (in decimal). If n is not punched the tape will be addressed from end to end.

2. Readdressing.

Punch a steering tape as follows:-

```
JOB
VAS -   ADDRESS TAPE

COMPILER TAD

R
a
b
c
d.
```
d_k
```
***Z
```

Where a = no. of blocks (in decimal))
b = identifier (1st, half)) attainable from the log
c = identifier (2nd Half)) of addressed tapes

$d_o \geqslant d_k$ = Faulty block addresses (inoctal)

N.B. Spaces may be used to terminate any of the elements in the data, but other deviation from this format will cause the programme to monitor.

3. For both addressing and readdressing:

Mount the magnetic tape on deck 7 but do not engage until instructed to do so by the operator's teleprinter. Feed in the steer tape on any tape reader.

When the routine has been entered, the message 'ENGAGE DECK 7' will be printed. Engaging the deck will cause the program to start and instructions and comments will be printed from time to time.

N.B. The commands 'MODIFY CHANNEL 7', 'UNMODIFY CHANNEL 7', and 'PERMIT WRITE ON CHANNEL 7' are accompanied by the deck being disengaged. Obey the command and then re-engage the deck.

TAPE ADDRESSING AND READDRESSING

DESCRIPTION

Summary

A Supervisor routine which will address tapes or remove from previously addressed tapes any blocks whose addresses are specified by a steering tape. Deck time is approximately 7 mins. per 1000 blocks.

Method

The process is broken up into passes as follows:-

Pass 1:-

A 20 ft. length of clear tape is run out to serve as a leader. This is followed by marked out blocks containing 8191 in the leading and trailing addresses and "all 1/s" in the region where information can be written. During this pass, a block is deemed faulty if:

1. A Reference Mark cannot be written.
2. A Block Mark can not be written.
3. Either address read back is not 8191
4. All 1's cannot be written in the information area.

Since addresses are not written sequentially in this pass, the fault list is compiled as a set of strings, consecutive entries in the list referring to good and faulty regions of tape, respectively. Each entry in the list is the count of the number of good block marks in the corresponding region.

Pass 2:-

The last 6 blocks are regarded as faulty to serve as a trailer and, by reference to the list prepared during pass 1, the Reference Marks associated with the faulty blocks are erased.

Pass 3:-

Using the Reference Marks which are left after pass 2, blocks are rewritten w ith sequential addresses from 0 onwards (except that 0,5000 and the last block are addressed as 8189, 8190, and 8191 respectively). The information area is erased and no clock pulses are written. Snigs ar e written in the interblock gap and a block is deemed faulty if:

1. A block mark cannot be written
2. The leading address is read back incorrectly.
3. The trailing address is not read back as zero.

The presence of faults reverts the programme to a previous pass.

Pass 4:-

Checks in reverse that the addresses are correct and that there is at least one snig in each inter-block gap.

Pass 5:-

Rewind connecting pass 3 to pass 1.

Pass 6:-

Search routine to enter pass 2 from pass 4.

After the successful completion of pass 4, Block 0 is written, the tape name being EX-SYSTEM. The deck is then unmodified, and a block of floating point zeros is written from Block 1 to Block 4999 or the penultimate block,which ever is the earlier. The tape is then rewound and named FREE. These two passes are not given a number within the tape addressing process.

TAPE ADDRESSING AND READDRESSING

MODES OF BEHAVIOUR

1. Monitoring

The Programme ends under the following fault conditions:-

OPERATOR'S OUTPUT	Reason
Error in data	Illegal character, wrong no. of entries, incorrect identifiers or otherwise inconsistent data on the steering tape.
Short blocks.	Average block length during pass 1 nominally less than 7.5 inches.
Too many faults.	Insufficient working space to accommodate entries from data tape or from fault checking routines. Working space is adequate for at least 190 faulty blocks at any stage in the process. (During passes 1 and 3, due to the Stringing System'a specific number of blocks cannot be quoted. Here the number of strings can be 277.)

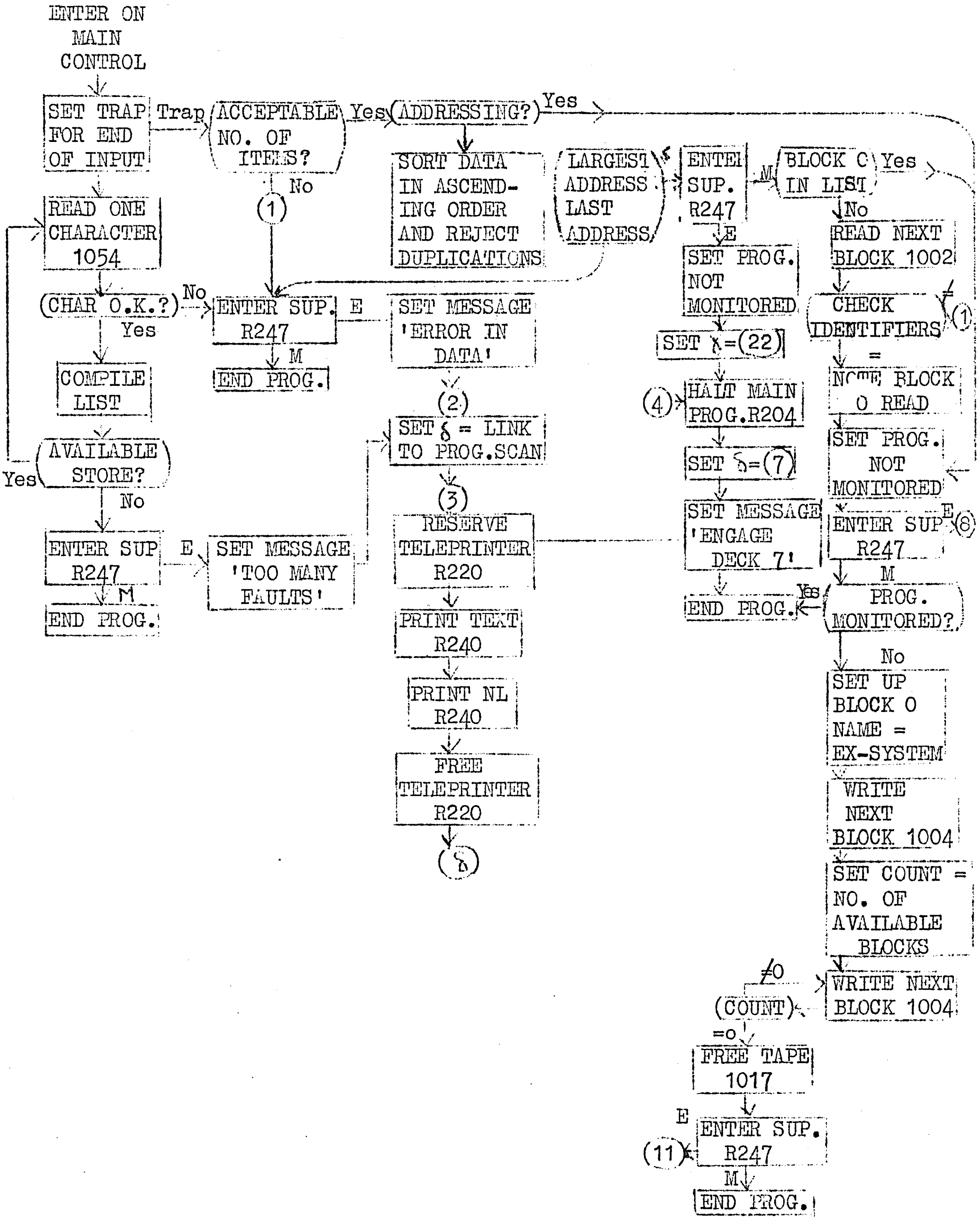
2. POSSIBLE PATHS THROUGH THE ROUTINE

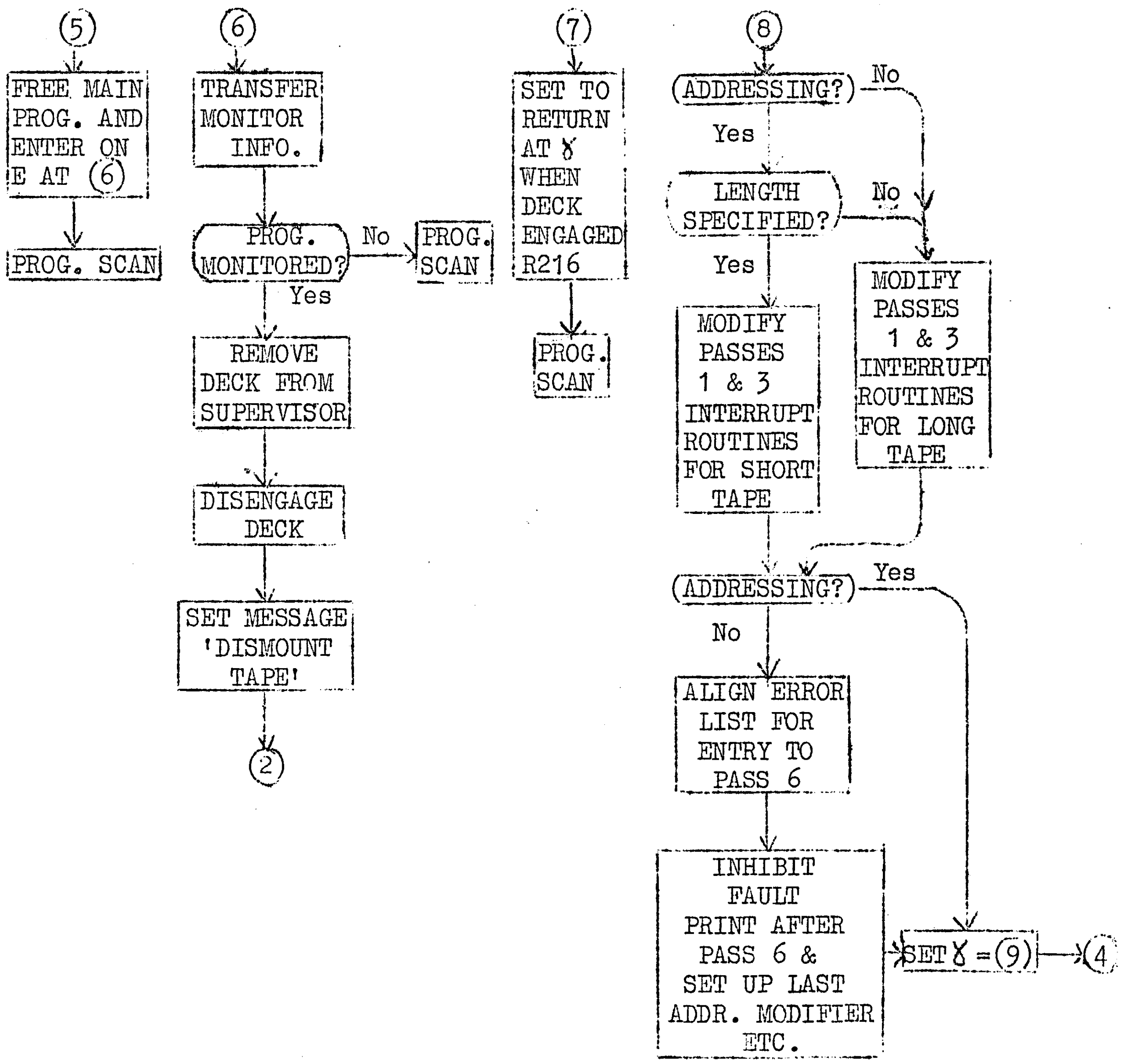
Normally, the process goes through passes 1,2,3, and 4 in that order. Reasons for reversion to previous passes are given below:-

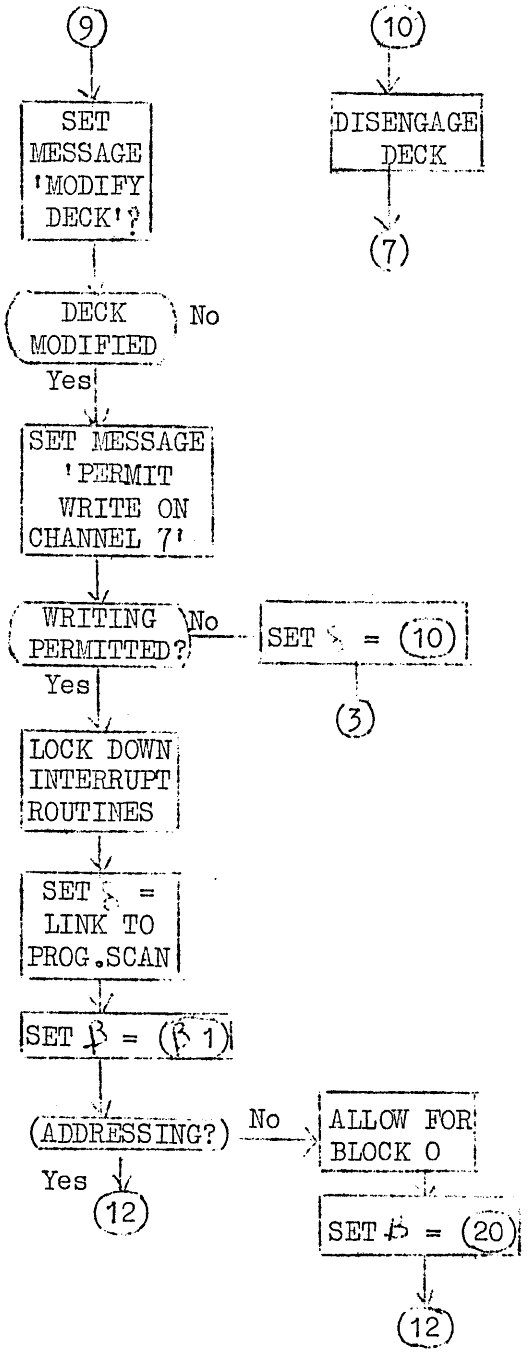
TRANSFER	REASON
2-> 1	No. of BM's read in pass 2 differs from the number classed as written during pass 1.
3-> 2	Faulty blocks written during pass 3.
3-> 5-> 1	No. of RM's read in pass 3 differs from the number expected to remain after erazing in pass 2.
4-> 3	No, of BM's read in pass 4 differs from the number classed as written in pass 3 or snig missing from interblock gap.
4-> 6-> 2	Address errors detected in pass 4.

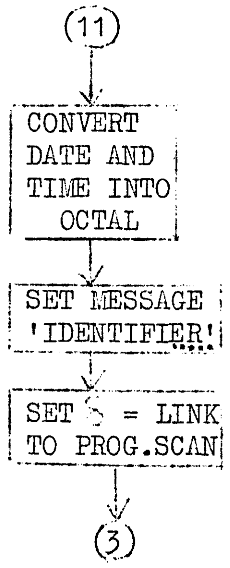
TAPE ADDRESSING AND RE-ADDRESSING

MAIN PROGRAMME AND MONITOR ROUTINES



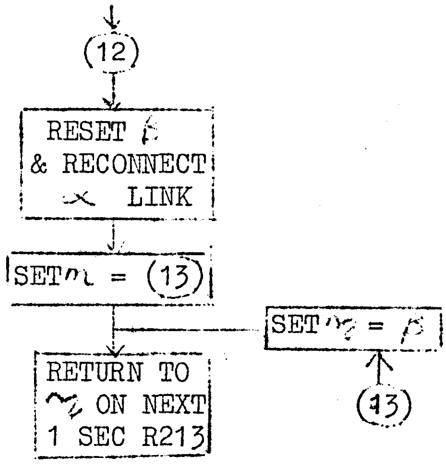

5
FREE MAIN PROG. AND ENTER ON E AT 6
PROG. SCAN
6
TRANSFER MONITOR INFO.
PROG. MONITORED?
No
PROG. SCAN
Yes
REMOVE DECK FROM SUPERVISOR
DISENGAGE DECK
SET MESSAGE 'DISMOUNT TAPE'
2
7
SET TO RETURN AT γ WHEN DECK ENGAGED R216
PROG. SCAN
8
ADDRESSING?
No
Yes
LENGTH SPECIFIED?
No
Yes
MODIFY PASSES 1 & 3 INTERRUPT ROUTINES FOR SHORT TAPE
MODIFY PASSES 1 & 3 INTERRUPT ROUTINES FOR LONG TAPE
ADDRESSING?
Yes
No
ALIGN ERROR LIST FOR ENTRY TO PASS 6
INHIBIT FAULT PRINT AFTER PASS 6 & SET UP LAST ADDR. MODIFIER ETC.
SET γ = 9
4



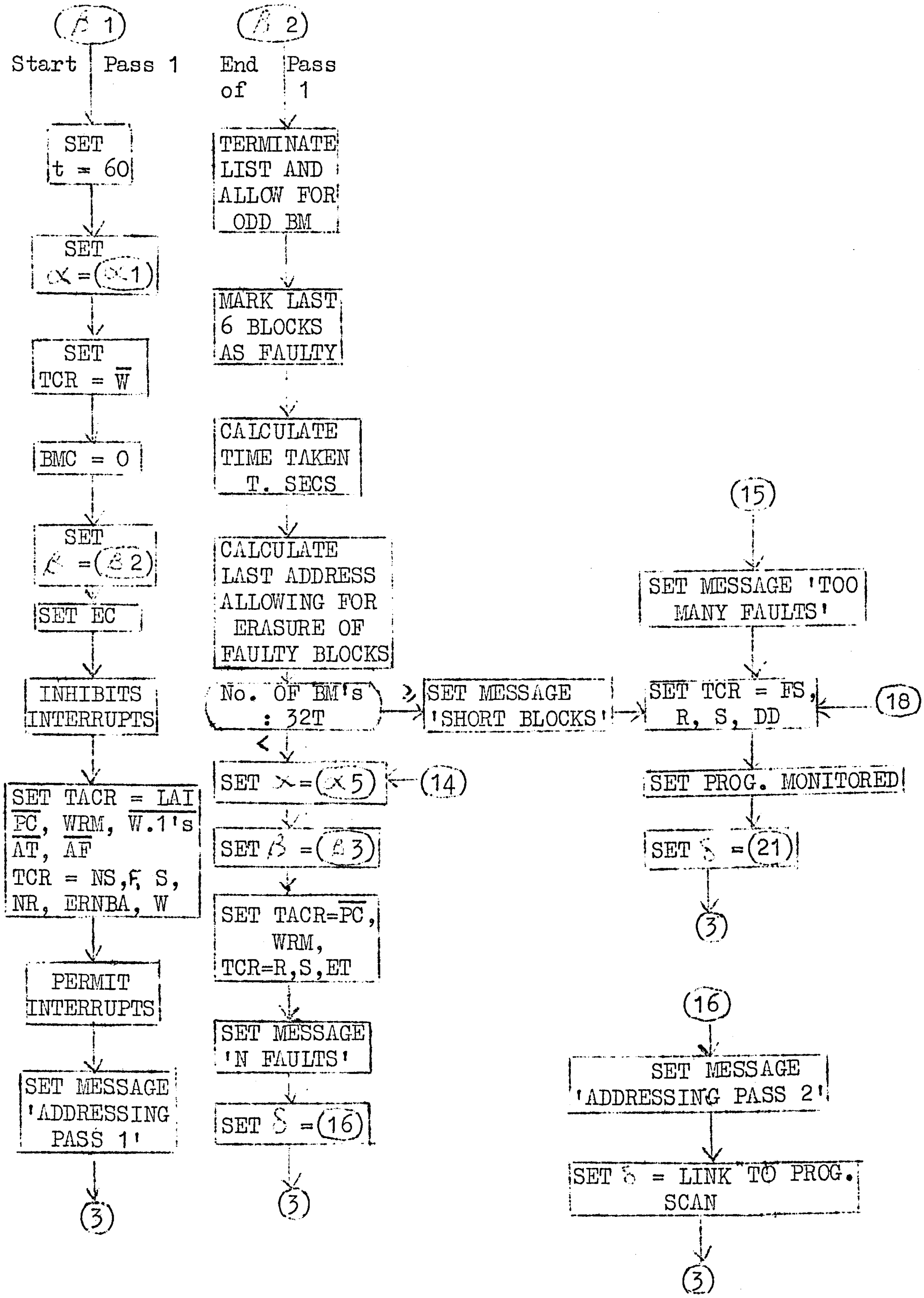


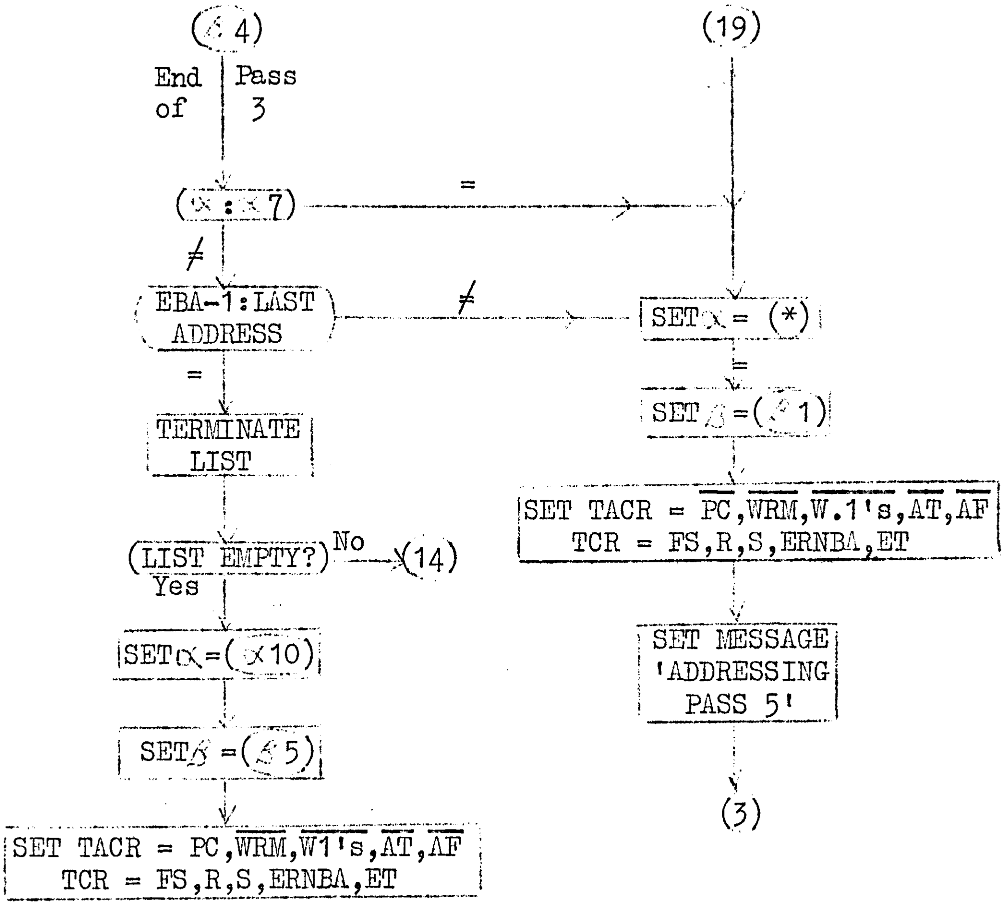
ENTRY FROM TAPE EXTRACODES ON END OF TAPE

ENTRY FROM TAPE EXTRACODE ROUTINE



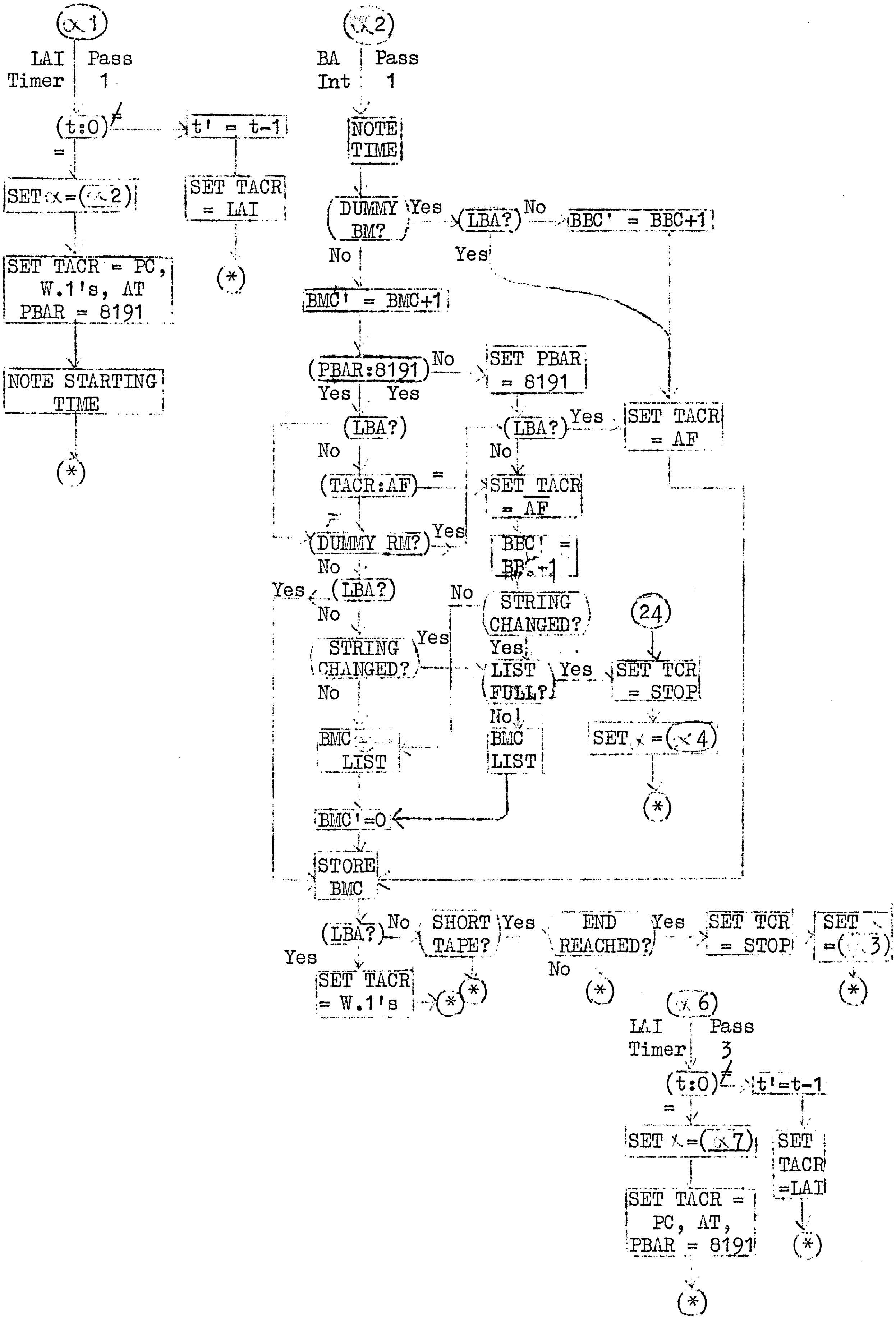
TAPE ADDRESSING AND RE-ADDRESSING 'END OF TAPE' ROUTINES




(β4)
End of
Pass 3
(α : α7)
=
≠
EBA-1 : LAST ADDRESS
≠
=
TERMINATE LIST
(LIST EMPTY?)
No
(14)
Yes
SET α =(α10)
SET β =(β5)
SET TACR = PC,WRM,W1's,AT,AF
TCR = FS,R,S,ERNBA,ET
(19)
SET α = (*)
=
SET β =(β1)
SET TACR = PC,WRM,W.1's,AT,AF
TCR = FS,R,S,ERNBA,ET
SET MESSAGE 'ADDRESSING PASS 5'
(3)

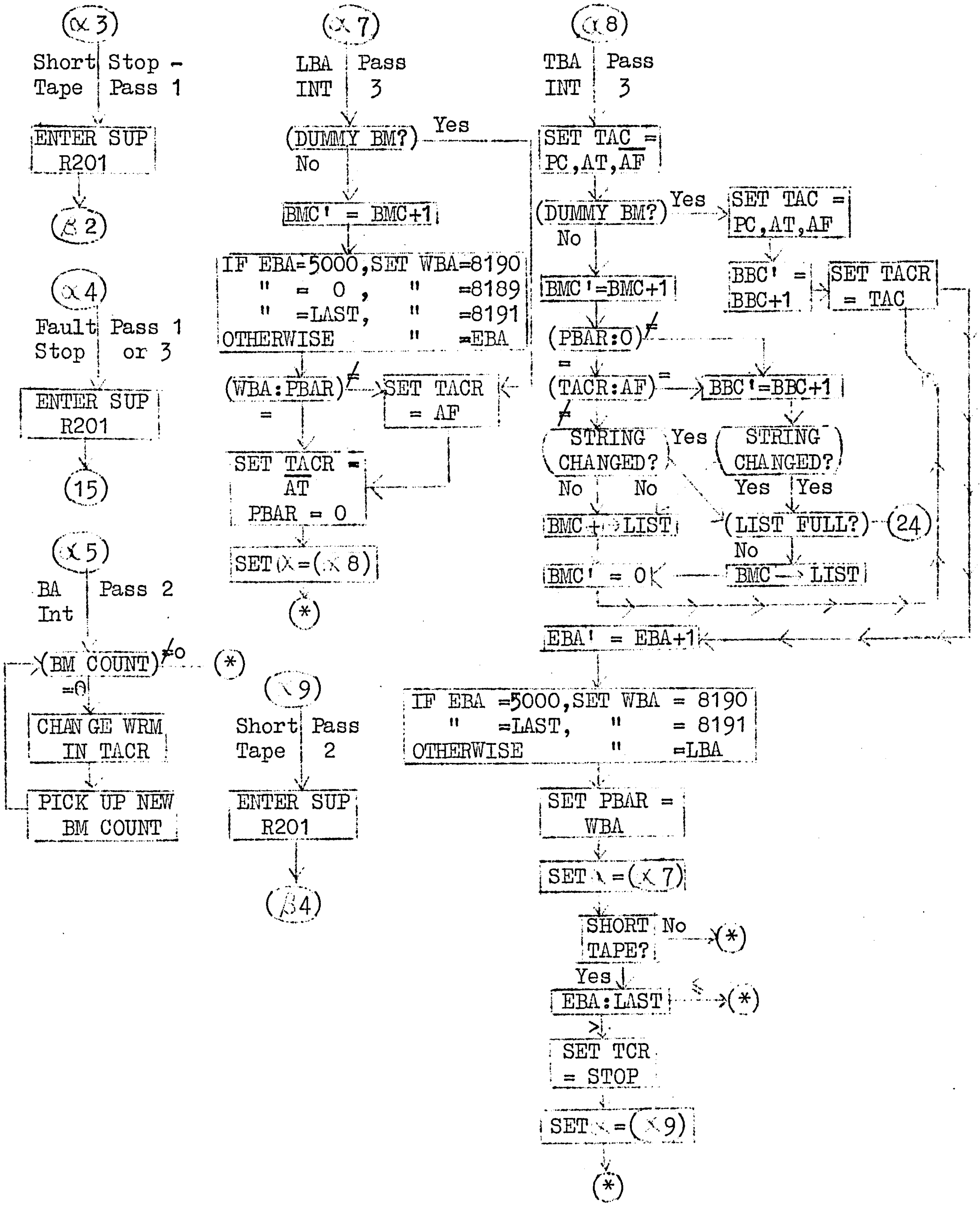
TAPE ADDRESSING AND RE-ADDRESSING

BA INTERRUPT ROUTINES

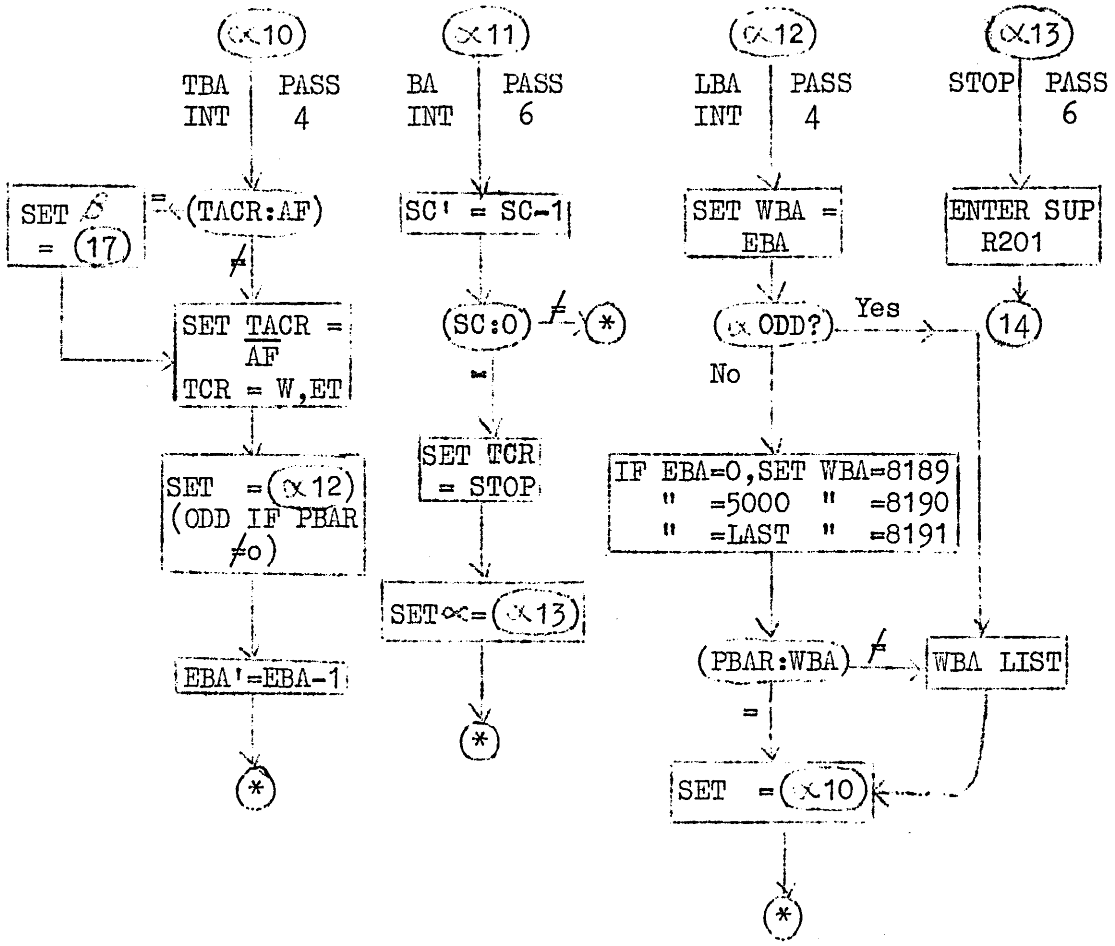


TAPE ADDRESSING AND RE-ADDRESSING

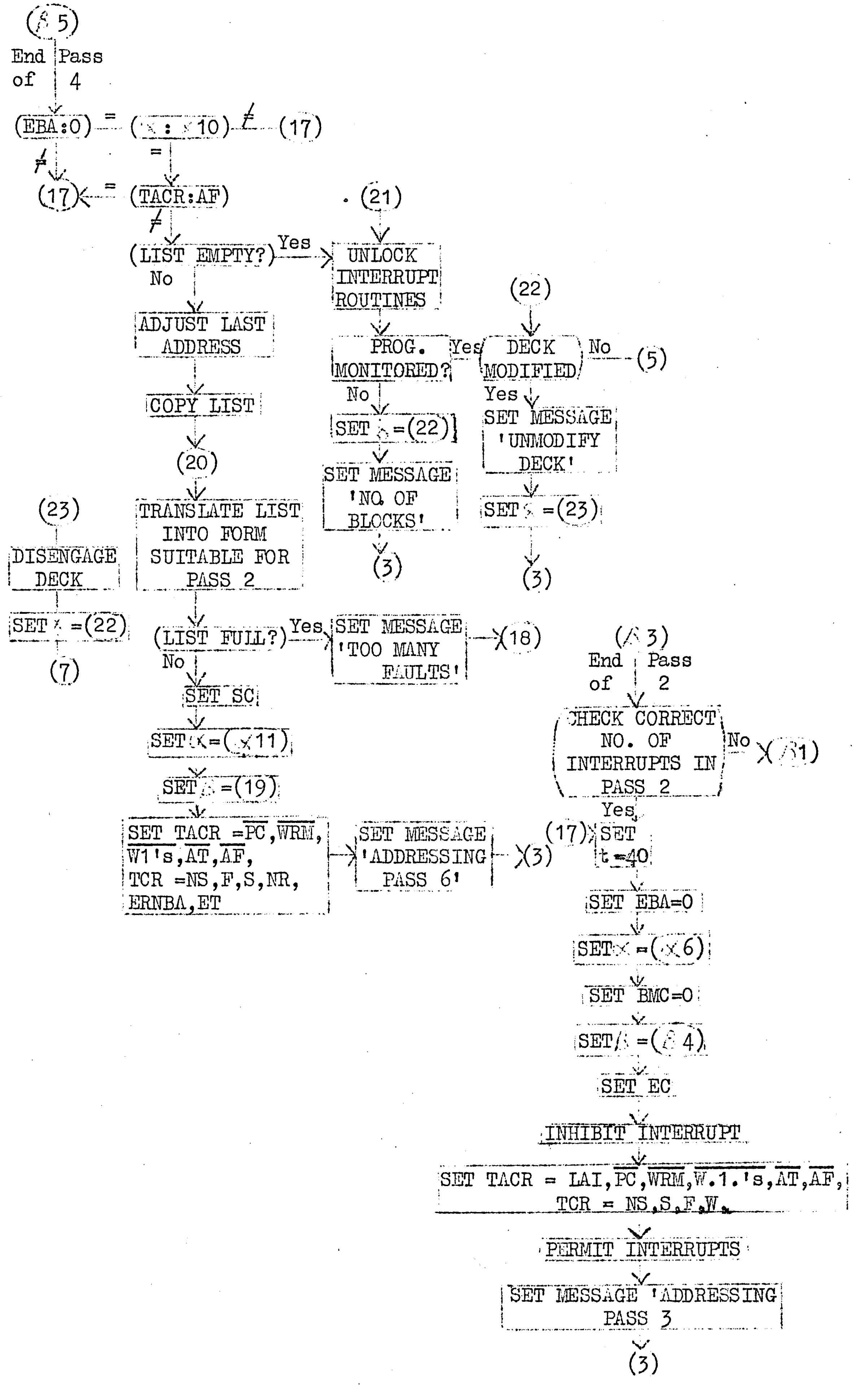
BA INTERRUPT ROUTINES



TAPE ADDRESSING AND RE-ADDRESSING BA INTERRUPT ROUTINES



TAPE ADDRESSING AND RE-ADDRESSING


(β5)
End of
Pass 4
(EBA:0)
=
≠
(α : α10)
(17)
(TACR:AF)
(LIST EMPTY?)
Yes
No
ADJUST LAST ADDRESS
COPY LIST
(20)
TRANSLATE LIST INTO FORM SUITABLE FOR PASS 2
(LIST FULL?)
SET MESSAGE 'TOO MANY FAULTS'
(18)
SET SC
SET α=(α11)
SET β =(19)
SET TACR =PC,WRM, W1's,AT,AF, TCR =NS,F,S,NR, ERNBA,ET
SET MESSAGE 'ADDRESSING PASS 6'
(3)
(21)
UNLOCK INTERRUPT ROUTINES
PROG. MONITORED?
SET β =(22)
SET MESSAGE 'NO. OF BLOCKS'
(22)
DECK MODIFIED
(5)
SET MESSAGE 'UNMODIFY DECK'
SET β =(23)
(23)
DISENGAGE DECK
SET β =(22)
(7)
(β3)
End of
Pass 2
CHECK CORRECT NO. OF INTERRUPTS IN PASS 2
(β1)
SET t =40
SET EBA=0
SET α =(α6)
SET BMC=0
SET β =(β4)
SET EC
INHIBIT INTERRUPT
SET TACR = LAI,PC,WRM,W.1.'s,AT,AF, TCR = NS,S,F,W.
PERMIT INTERRUPTS
SET MESSAGE 'ADDRESSING PASS 3

TAPE ADDRESSING AND READDRESSING

DESCRIPTION

Summary

A Supervisor routine which will address tapes or remove from previously addressed tapes any blocks whose addresses are specified by a steering tape. Deck time is approximately 7 mins. per 1000 blocks.

Method

The process is broken up into passes as follows:-

Pass 1:-

A 20 ft. length of clear tape is run out to serve as a leader. This is followed by marked out blocks containing 8191 in the leading and trailing addresses and "all 1/s" in the region where information can be written. During this pass, a block is deemed faulty if:

1. A Reference Mark cannot be written.
2. A Block Mark can not be written.
3. Either address read back is not 8191 .
4. All 1's cannot be written in the information area.

Since addresses are not written sequentially in this pass, the fault list is compiled as a set of strings, consecutive entries in the list referring to good and faulty regions of tape, respectively. Each entry in the list is the count of the number of good block marks in the corresponding region.

Pass 2:-

The last 6 blocks are regarded as faulty to serve as a trailer and, by reference to the list prepared during pass 1, the Reference Marks associated with the faulty blocks are erased.

Pass 3:-

Using the Reference Marks which are left after pass 2, blocks are rewritten with sequential addresses from 0 onwards (except that 0,5000 and the last block are addressed as 8189, 8190, and 8191 respectively). The information area is erased and no clock pulses are written. Snigs are written in the interblock gap and a block is deemed faulty if:

1; A block mark cannot be written
2. The leading address is read back incorrectly.
3. The trailing address is not read back as zero.

The presence of faults reverts the programme to a previous pass.

Pass 4:-

Checks in reverse that the addresses are correct and that there is at least one snig in each inter-block gap.

Pass 5:-

Rewind connecting pass 3 to pass 1.

Pass 6:-

Search routine to enter pass 2 from pass 4.

After the successful completion of pass 4, Block 0 is written, the tape name being EX-SYSTEM. The deck is then unmodified, and a block of floating point zeros is written from Block 1 to Block 4999 or the penultimate block,which ever is the earlier. The tape is then rewound and named FREE. These two passes are not given a number within the tape addressing process.

TAPE ADDRESSING AND READDRESSING

MODES OF BEHAVIOUR

1. **Monitoring**

The Programme ends under the following fault conditions:-

OPERATOR'S OUTPUT	Reason
Error in data	Illegal character, wrong no. of entries, incorrect identifiers or otherwise inconsistent data on the steering tape.
Short blocks.	Average block length during pass 1 nominally less than 7.5 inches.
Too many faults.	Insufficient working space to accommodate entries from data tape or from fault checking routines. Working space is adequate for at least 190 faulty blocks at any stage in the process. (During passes 1 and 3, due to the Stringing System a specific number of blocks cannot be quoted. Here the number of strings can be 277.)

2. **POSSIBLE PATHS THROUGH THE ROUTINE**

Normally, the process goes through passes 1,2,3, and 4 in that order. Reasons for reversion to previous passes are given below:-

TRANSFER	REASON
2-> 1	No. of BM's read in pass 2 differs from the number classed as written during pass 1.
3-> 2	Faulty blocks written during pass 3.
3-> 5-> 1	No. of RM's read in pass 3 differs from the number expected to remain after erazing in pass 2.
4-> 3	No. of BM's read in pass 4 differs from the number classed as written in pass 3 or snig missing from interblock gap.
4-> 6-> 2	Address errors detected in pass 4.

R450,451/6

TAPE ADDRESSING AND READDRESSING
GLOSSARY OF TERMS

*	2048*4 - Exit to Sort Interrupts.
α	Link obeyed on a Block Address Interrupt
β	Link obeyed on End of tape.
γ	Link to return from R216 when Deck 7 is engaged
δ	Link obeyed after Operator's Output
η	Link from 1 sec. SER
AF	Address Fault.
AT	Address Tape
BA	Block Address
BBC	Bad Block Count.
BM	Block Mark
BMC	Block Mark Count.
DD	Disengage Deck.
EBA	Expected Block Address
EC	Error Count
ERNBA	End Read at Next Block Address
ET	End Transfer
F	Forwards
FS	Fast Speed
LAI	If Deck modified and AT reset, causes BA interrupt after 40 insec.
LBA.	Leading Block Address
NR	Normal Read
NS	Normal Speed
PBAR	Present Block Address Register
PC	Permit Count
R	Reverse
RM	Reference Mark
S	Start
SC	Search Count
t	Timing count for leader in passes 1 and 3
T	Time in seconds between 1st and last BM's during pass 1.
TAC	Tape Addressing Command.
TACR	Tape Addressing Command Register
TBA	Trailing Block Address
TCR	Tap e Command Register
W	Write
WBA	Wanted Block Address
W1's	Write 1's
WRM	Write Reference Mark.
SNIG	A "return to zero" pulse in the clock track recorded after the end of the information stripes because there are an odd number of these in the complete block.

R480: Tape engage and disengage

Purpose: A main store SER to analyse decks detected by the One Second routine as having the engage status changed. A suitable SER is entered to the tape SER queue to deal with each deck calling for action.

Registers of main store: 26

Instructions obeyed: 3 + 6D+(8 to 23 per deck requiring attention) where D = no. of decks

Parameters used: (1) to (16)

Cross references:

(5)	=	(8/230)	Tape for action
(6)	=	(3/221)	Number of decks
(7)	=	(1/202)	Program scan
(8)	=	(68/400)	Record of engage tapes
(9)	=	(5/221)	Deck allocation directory
(10)	=	(1/217)	Tape exit to supervisor
(11)	=	(2/206)	Enter SER to queue
(12)	=	(5/214)	Base of SER queues
(13)	=	(3/213)	Halt positions in SER queues
(14)	=0.4	(53/421)	Expected Block Address
(15)	=	(1/482)	Entry to read title

Connections with other routines:

Entered at (1) via co-ordinator from entry in SER queue planted by the one second routine. B registers irrelevant.

Exit: to program scan (1/202) with action record (8/230) zero. The routine is never halted once entered.

Subroutines:

a) "Enter SER to queue"

Entered at (2/206) with
B107 = 1.0
B108 = Deck number(digits 7-3)
B109 = Entry to new SER (1/482)
B110 = Return address
B126 = odd

Return with B101, 108 unaltered

b) "Tape exit to Supervisor control"
Entered at (1/217) with
B109 = Deck number (digits 7-3)
B110 = Return address

Exit to return address, B registers irrelevant.

Temporary Working Space: B100-110 Bt

R480: continued

Notes:

1. For each deck requiring attention, an entry is made to the tape SER queue as follows: In all cases the information preserved is the deck number.

   a) Tape engage normally: SER 1/482 to read and check title.

   b) Tape engaged specially (e.g. for addressing testing or re-engaged after a fault): the SER recorded in the halted tape queue earlier via R216 is brough to the active part of the queue. This condition is indicated by 1 in digit 1 of the deck allocation directory. Digit 1 is reset to zero by R480.

   c) Tape disengaged following computer disengage. No action is taken. This condition is indicated by 1 in digit 11 of the deck allocation directory. Digit 11 is reset to zero by R480.

2. The above actions observed the limit of entries to the tape SER queue, one per deck and two per channel.

3. The record of tapes requiring action is reset to zero on exit.

4. The routine requires modification to deal with more than 16 decks, numbered 0 to 15.

R490: Fixed store tape organisation extracodes

Purpose: A fixed store extracode program entered from extracode jump table of extracodes 1007 to 1024 inclusive to enter relevant programs in main store, "in supervisor".

Registers of fixed store: 22

Instructions obeyed: 3 to 5 for each extracode before entry to R247.

Parameters used: (1) to (22)

Cross references:

(14)	=	(1/247)	Program load B
(15)	=	(3/247)	Prepare load store
(16)	=	(1/492)	Main store tape organisation
(17)	=	(1/498)	Exit for mount
(18)	=	(2/498)	Exit for mount free
(19)	=	(3/498)	Exit for mount next reel
(20)	=	(5/499)	Exit for accept - not used
(21)	=	(1/496)	Exit for release
(22)	=	(4/492)	Exit for rename

Connections with other routines:

All entries are direct from the extracode jump table. Exit is to 1/247 or 3/247 with B91, B92 as shown below. B121, 119 are unaltered. This causes exit to the address in B91, in supervisor with full recovery switch set

E Code	Entry to R491	Exit to (1) or (3) of 247	B91 on exit	B92 on exit
1007	(3)	3	(3/498)	-
1010	(1)	3	(1/498)	-
1011	(2)	3	(2/498)	-
1012	(1)	3	(3/498)	-
1013	(2)	3	(5/499)	-
1014	(5)	3	(1/492)	0.1
1015	(6)	3	(1/492)	0.4
1016	(7)	1	(1/492)	2.0
1017	(8)	1	(1/492)	2.5
1020	(9)	1	(1/492)	3.0
1021	(10)	1	(1/496)	-
1022	(11)	1	(4/492)	-
1023	(12)	3	(1/492)	1.1
1024	(13)	3	(1/492)	1.5

Temporary Working Space: Nil

Notes: The value of B92 where appropriate is carried over via R247 to the SER which starts in the address specified in B91.

R492: Main store tape organisational extracodes

Purpose: An SER in main store entered from fixed store R491 by extracodes referring to tape B. Finds the actual deck number involved and enters various routines to obey specific extracodes. Includes within itself the extracodes "Re-allocate" and "Tape length".

Registers of main store: 42

Instructions obeyed: Most extracodes: 6 + entry to R221 to find deck number.

Extracode "Length": 13 + entry to R221

Extracode "Re-allocate": 14 + entry to R221

Parameters used: (1) to (20)

Cross references:

(5)	=	(1/221)	Find deck number
(6)	=	(9/205)	Current program number in store control
(7)	=	(3/221)	Number of decks
(8)	=	(5/221)	Deck allocation directory
(9)	=	(9/230)	Deck timer directory
(10)	=	(5/201)	SER re-entry
(11)	=	(1/215)	Set full recovery switch
(12)	=	(99/900)	Extracode working space
(13)	=	(4/247)	Return to main program
(15)	=	(1/497)	"Where am I"? extracode
(16)	=	(2/494)	Write title
(17)	=	(1/494)	Read title
(18)	=	(2/495)	Unload
(19)	=	(1/495)	Free
(20)	=	(3/495)	Release tape

Connections with other routines:

Entered at 1) From R491 for extracodes 1014 - 1020, 1023 1024 with B92 as described for R491, full recovery switch set.

Exit: To monitor via15(1/221) if deck not defined
To halt program via 8(1/221) if deck not available
Otherwise to the relevant routine as listed above with full recovery switch set, extracode B lines and working space unaltered, and re-entry address set to the start of the relevant routine.
B100 = absolute deck number (digits 7 - 3)

Entered at 4) For re-allocate via R491
B119 = new label, digits 9 - 3 rest irrelevant
ba = old label, digits 9 - 3 rest irrelevant

Exit: To (4/247) to return to main program with B119 unaltered.
B121 = 0

R492: continued

Re-entered at (3) for extracodes "Length of tape"

Exit: (4/247) to return to main program with
B91 = B92 = 0
(99/900) = Length of tape, digits 15-3
rest zero.

Subroutines:

a) "Set full recovery switch":
Entered at (1/215) with B109 = Address of SER to deal with specific extracodes.
Exit: to address in B109. B110 unaltered.

b) "Find deck number":
Entered at (1/221) to find Atlas deck
B109 = Return address
B100 = Programmers label, digits 8-2
digits 0 = 0

Exit: To return address with B100 = deck number
digits 7 - 3
or to monitor
or to halt program: Re-enter at (1/492) if deck not available.

Entered at 15(1/221) if Atlas or Orion deck not found,
to monitor

Entered at 8(1/221) if Atlas or Orion deck found,
B107 = deck number, digits 7 - 3
B106 = contents of deck allocation directory
digit 0.
B109 = return address

Exit: To return address if deck available,
B100 = deck label
To halt program if deck not available
Note that the full recovery switch is reset
on exit to return address

Temporary working space: Entry 1) B100, 106-109
Entry 4) B100, 101, 121

Notes:

1. On entry (1) if B92 is odd, R221 is entered to locate an Atlas tape (monitor if Orion tape). If B92 is even, a search is made for an Atlas or Orion tape of the correct label.

2. The program is halted via R221 if the tape referred to is not available (e.g. being mounted, under supervisor control, etc.).

R493: Tape message printer

Purpose: An SER subroutine in main store to print messages to the tape operator. Alternative entry conditions allow for printing the title of a tape in addition.

Registers of main store: 44

Instructions obeyed: 25 if message only; maximum 39 +2D if title also, where D = number of decks.
Also entries to R240 to assemble output which will dominate the number of instructions obeyed.

Parameters used: (1) to (13)

Cross references:

(5)	=	(1/220)	Reserve output
(6)	=	(2/220)	Free output
(8)	=	(12/213)	SER dump address
(9)	=	(5/201)	SER re-entry address
(10)	=	(1/240)	Print message
(11)	=	(8/494)	Deck title directory
(12)	=	(3/240)	Print layout

Connections with other routines:

Entered at 1) with B100 = Deck number (digits 7-3)
B104 = Return address (digits 22-3)
Digit 0 = 1 (print title)
0 (no print title)
B103 = Location in store of message
Digits 22-0 (main store, starting at any character position).
Digit 23 = 0 (use title from title directory)
1 (use title from B105)
B105 = Location of title if other than title directory

Exit : a) To re-entry address if operators output is busy with B102, 101 altered

b) To return address when output is assembled in the buffer with

B100, 103, 104 unaltered
B126 digits 2 - 0 = 0
Re-entry address set to return address,
Digits 2-0 = 0

R493: continued

Subroutines:

a) "Reserve operators output": entered at (1/220) to reserve output channel

B101 = 0.4 (channel 1)
B100 = B102 = Deck number
B110 = Return address

Exit: To halt program (go back to re-entry) if busy
To return address, with B100 = working area of output, if channel not busy.

b) "Free operators output": Entered at (2/220) at conclusion with

B101 = 0.4
B100 = Deck number
B110 = Return address

Exit: To return address with B100, 103, 104 unaltered.

c) "Supervisor output": Entered at (1/240) to print message with

B100 = working store of output peripheral
B108 = 0.1 (message ends on character 00)
1.0 (message of two characters)
B109 = address of message
B110 = return address digit 0 = 1 to recover B100-104 on exit.
SER dump address = (7/493) - working space for R493

Exit after message written to buffer with
B100-104 preserved
Re-entry address set as B110 on entry

Entered at (3/240) to print "New line"
B100 = working area of peripheral
B109 = 2.1
B110 = Return address. Digit 0=1 as above.

Exit to return address with B100-104 preserved
re-entry address set as B110 on entry.

Temporary working space: B101, 102, B105-110, Bt
B100 used but reset to original value.

Notes:

1. Printing consists of message followed by deck number on one line, followed optionally by the tape title on a separate line. Message and title are in internal code, inner set, in store, and are terminated by character (octal) 00.

2. A title is only printed if B104 is odd on entry. The title is in the deck title directory for this deck, or in a separate location specified in B105 on entry. If the first half word of title is zero, the title "FREE" is printed.

R493: continued

3. After reserving the output channel, this routine uses a dump area for B100-104 in the event of halts. Only one such area is required, since only one message can be printed at once. If the output channel is busy, the routine calling in R493 is halted, with B100 103, 104 unaltered, and is resumed at the specified re-entry address; if this has digit 1=0, only B100 is preserved on restarting. Usually this is sufficient, as it contains the deck number both on entry to R493 and when the routine is halted by R220.

4. The message and tape title may be in any supervisor main store block bearing a reserved block label in the block directory, which can be called to core store by non-equivalence in supervisor. They must not occupy a supervisor main store block with a non-reserved block label.

R494: Extracodes Read/Write tape title

Purpose: A main store SER to implement the extracode Read title, Write title. An alternative entry provides a subroutine to copy a title from object program store to supervisor store, compressing where necessary.

Registers of main store: 80

Instructions obeyed: Read: Around 10 +2D + 7 per half word of title where D = number of decks.

Write: Around 20 + 2D + 10 per half word + 5 to 20 per character.

Parameters Used: (1) to (12)

(8) = "Deck title directory" 10 words per deck, holding title of tape on deck d in words 10d onwards.

Cross references:

(9)	=	(99/900)	Extracode working space
(10)	=	(4/247)	Exit to main program
(11)	=	(5/221)	Deck allocation directory
(12)	=0.4	(6/201)	Main program controls.

Connections with other Routines

Entry at (1) from R492 for "Read title to store S", in supervisor with full recovery switch set.
B119 = S
B100 = Deck number (digits 7-3)

Exit to (4/247) to reset full recovery switch and exit to main program with
B91 = 0
B92 = number of half words Extracode working space filled less 1 (digits 4-2)
Extracode working space = title or remainder of title.
B97 even
B119 unaltered

Entry at (2) from R492 for "Write title from store S", in supervisor with full recovery switch set
B119 = S
B100 = Deck number (digits 7-3)

Exit to (4/247) to reset full recovery switch and exit to main program with main program controls set to resume in main control.

Entry at 1(1) for subroutine "Find title"
B110 = Return address
B100 = Deck number (digits 7-3)

R494: continued

Exit to return address with
B101 = Location of title of deck in deck title directory (absolute address)
B109 = altered

Entry at (3) for subroutine "Copy title to supervisor store"
B110 = Return address
B101 = Location in supervisor store for title
B119 = Location of the title in the store belonging to the current main program in control of store.

Exit to return address with
B91-97, B100-110, Bt, B119 altered
Title copied and compressed (see notes)
(This subroutine is used by R498, which implements the extracodes "Mount" etc.).

Subroutine:

"Co-ordinate organisational extracode"

Entered with (4/247)

a) To copy to program store

B91 = Re-entry address to R494
B92 = Number of half words extracode working space loaded less 1, digits 8-2 = 3.4
B97 even
Return to address in B91 with B95 unaltered, B119 stepped to next transfer address

b) To read from program store

B91 = Entry address to R494
B92 = 0 (read one half word)
B97 odd
Return to address in B91 with one half word working space filled, B119 stepped by 0.4,B95 unaltered.

Temporary Working Space: B91-97, B100-110, Bt

Notes:

1. The title read from the deck title directory is in internal code characters; up to 79 significant characters are permitted. The last half word of title must contain zero in digits 5-0.

2. The title read from program store obeys the same rules. If 10 half words have been read and none has zero in digits 5-0, zero is forced to digits 5-0 of the last half word - the title is thus cut short.

R494: continued

3. After reading the title from program store and cutting short where necessary the title is analysed and condensed as follows:-

   a) Characters 03-07, 73-77 are omitted throughout.

   b) Character 02 (Tab) is replaced by 01 (space)

   c) At the start, characters 01, 12, 37 (space, comma, full stop) are ommitted.

   d) Throughout, multiple space characters are ignored (i.e. n spaces equal one space).

   e) The title is ended on character 00.

   (Octal internal code characters are used above).

   The last, partially filled, half word is filled by R494 with characters 00.

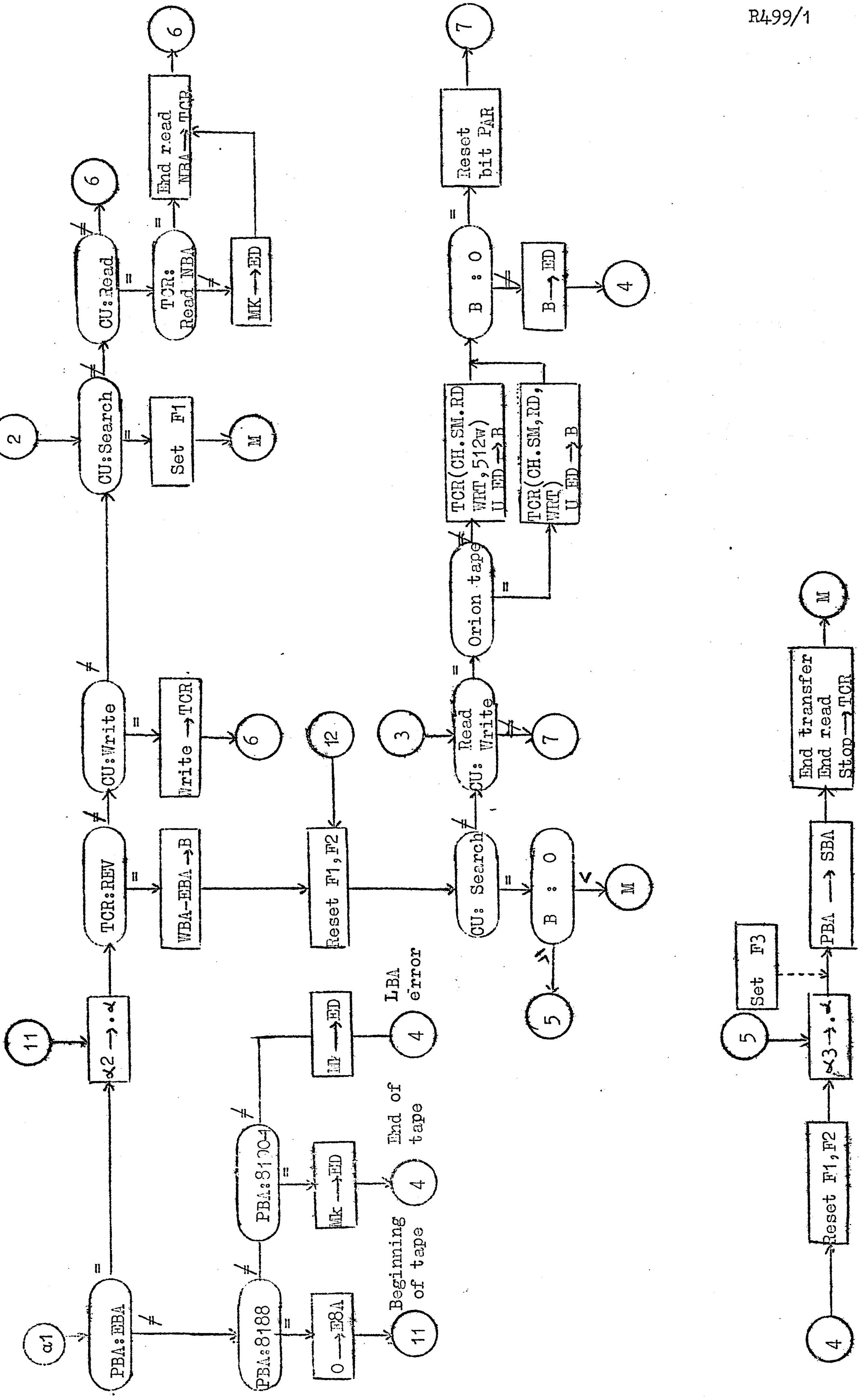
4. On entry at (2), digit 13 of the deck allocation directory is forced to 1 to indicate that the title has been changed.

5. Although for efficiency this routine would occupy the same block of store as the deck title directory, the routine still functions correctly if these are in separate blocks, and in fact it is convenient elsewhere to use separate blocks for these.

R499

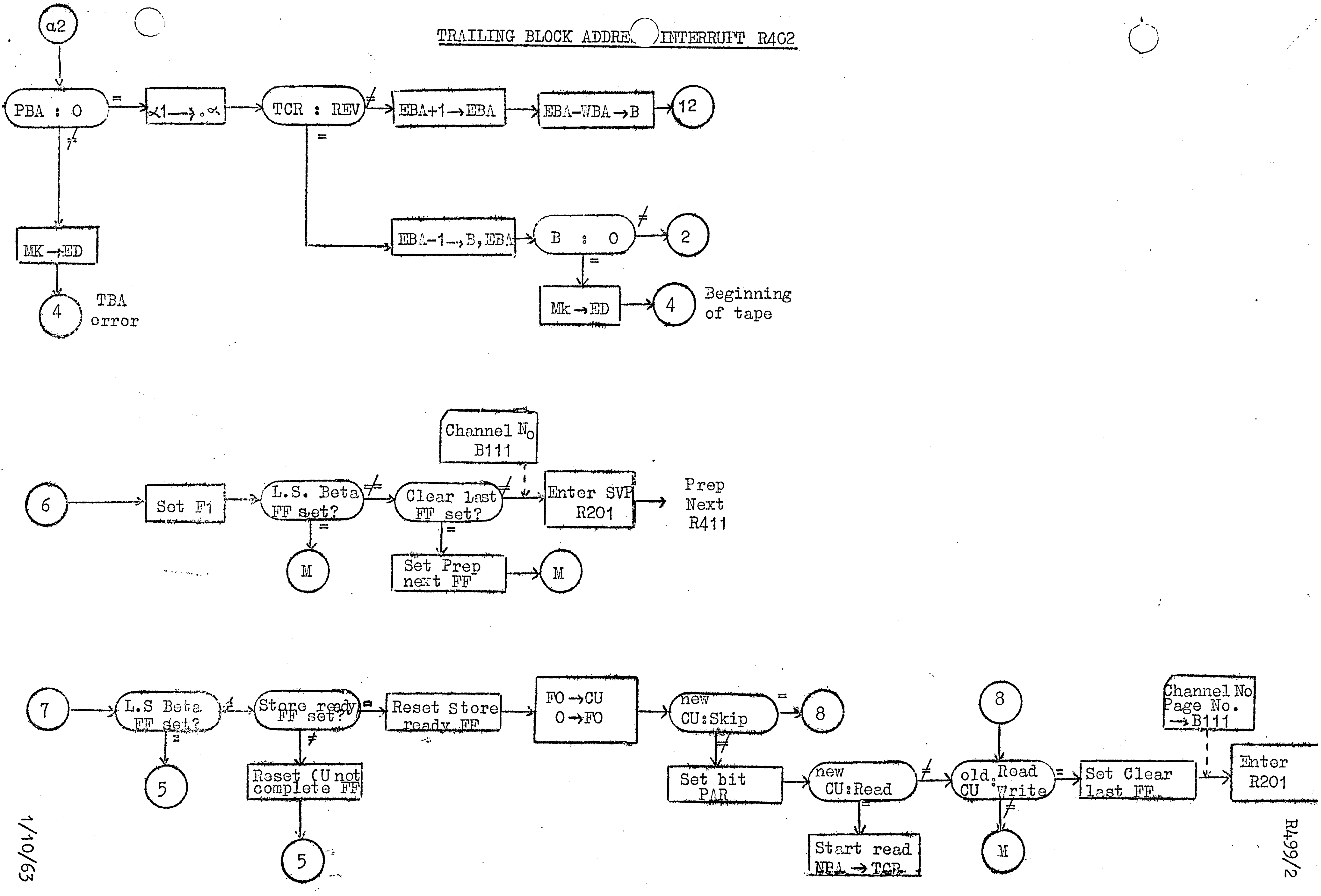
FLOW DIAGRAMS MAGNETIC TAPE ROUTINES

LEADING BLOCK ADDRESS INTERRUPT R402



1/10/63

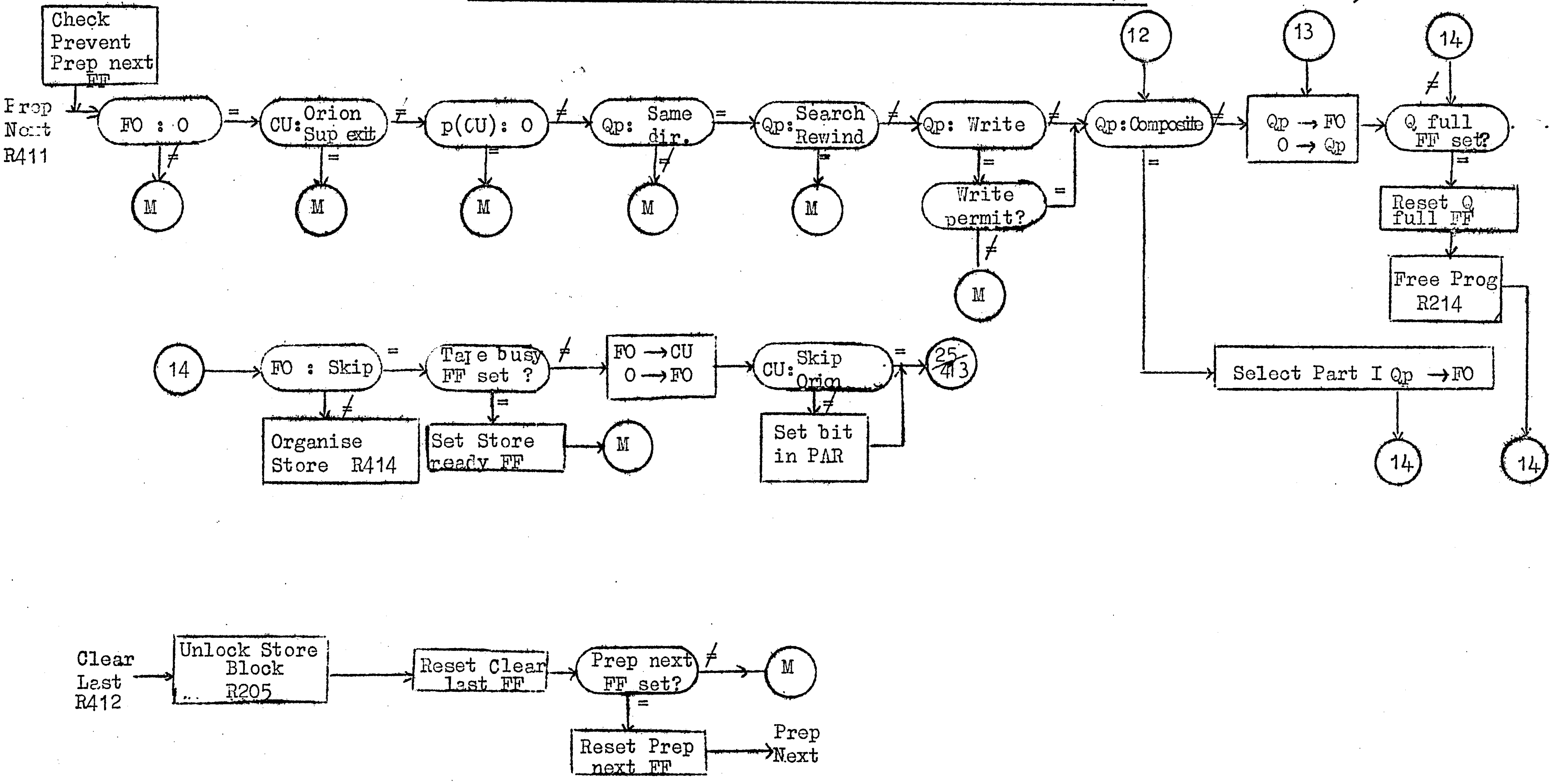
TRAILING BLOCK ADDRE INTERRUPT R402



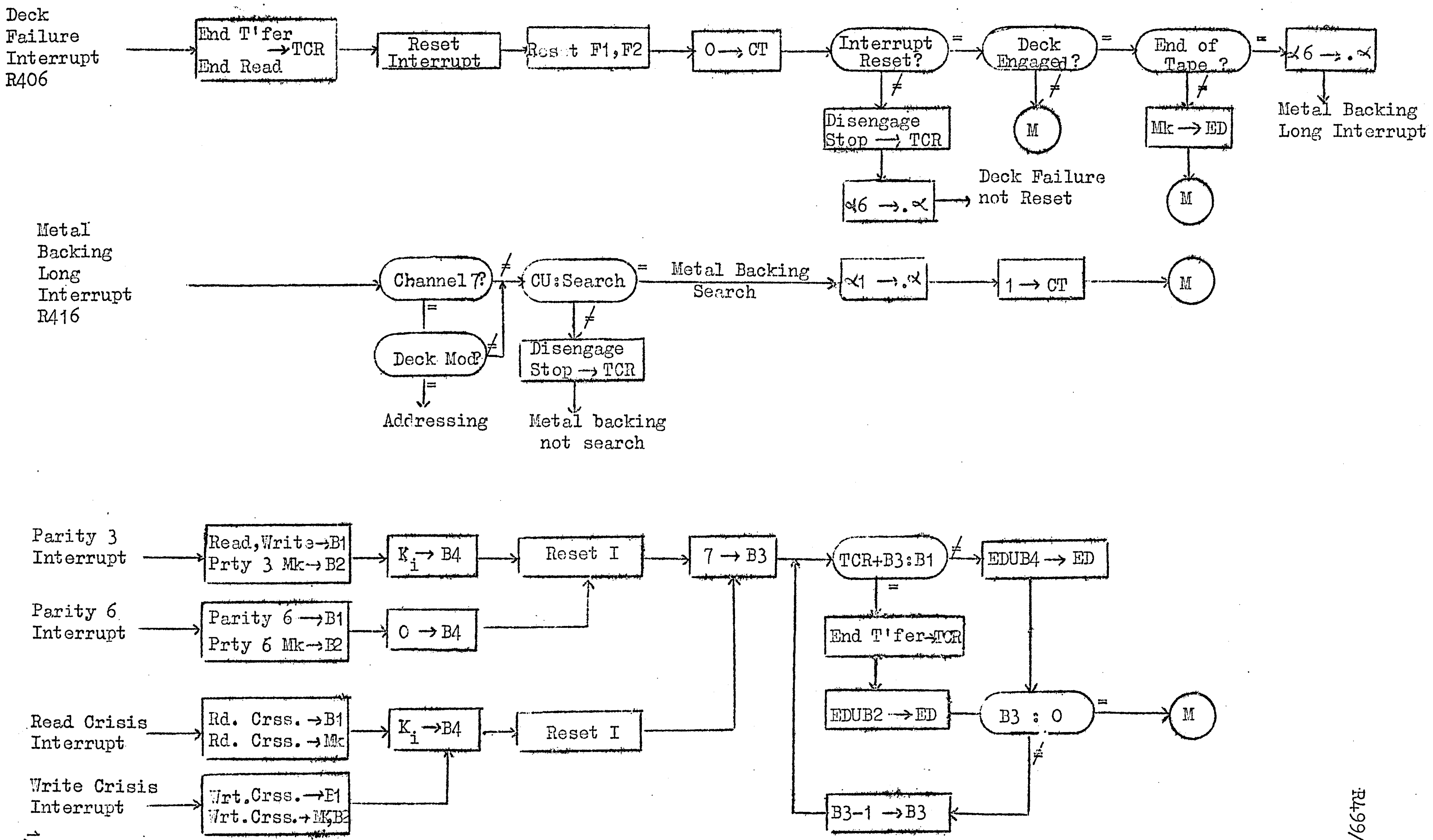
1/10/63

R499/2

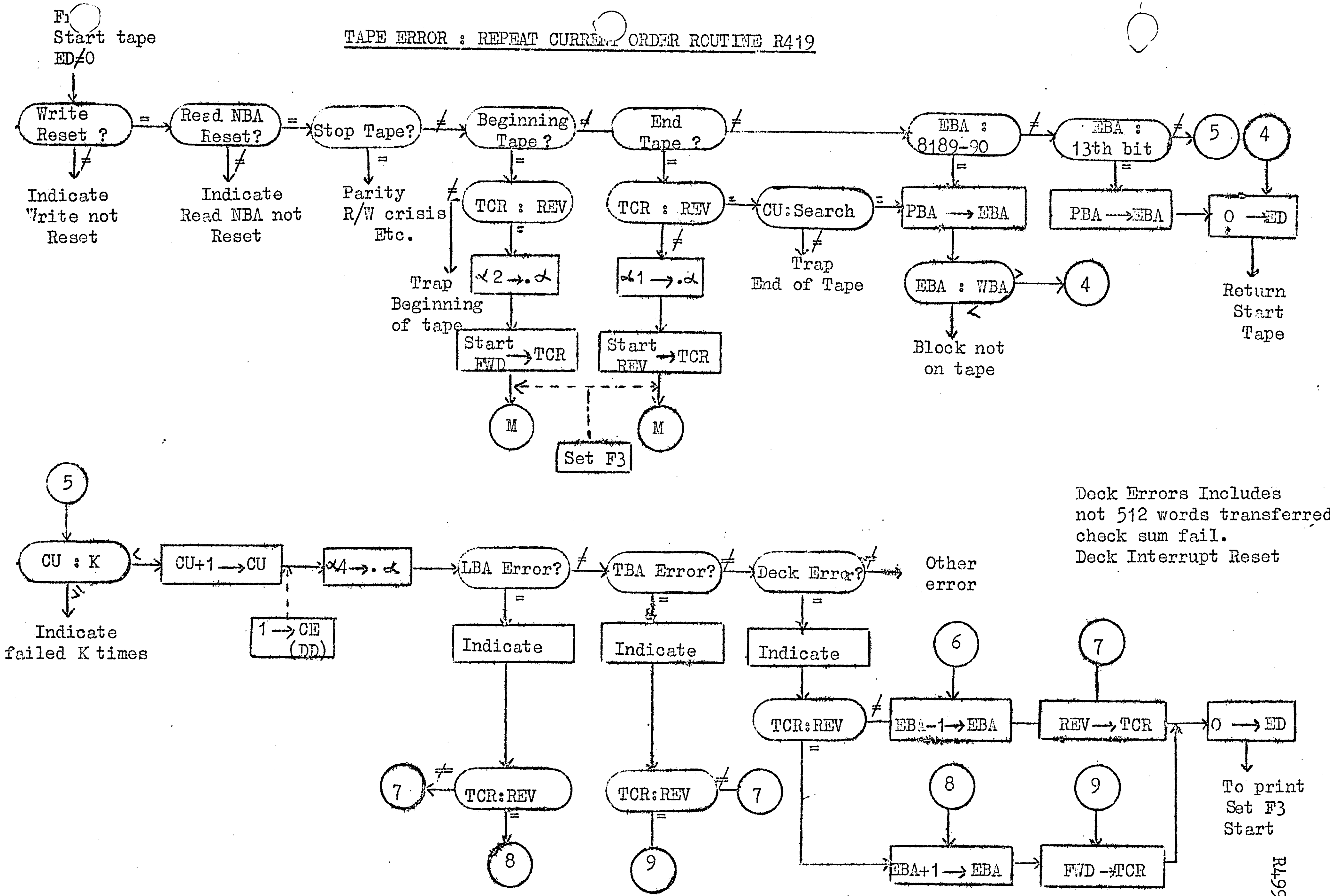
LONG INTERRUPTS PREPARED NEXT AND CLEAR LAST R411 AND R412



DECK FAILURE INTERRUPT, PARITY 3 6 and READ & WRITE CRISIS INTERRUPT R406, R407



TAPE ERROR : REPEAT CURRENT ORDER ROUTINE R419



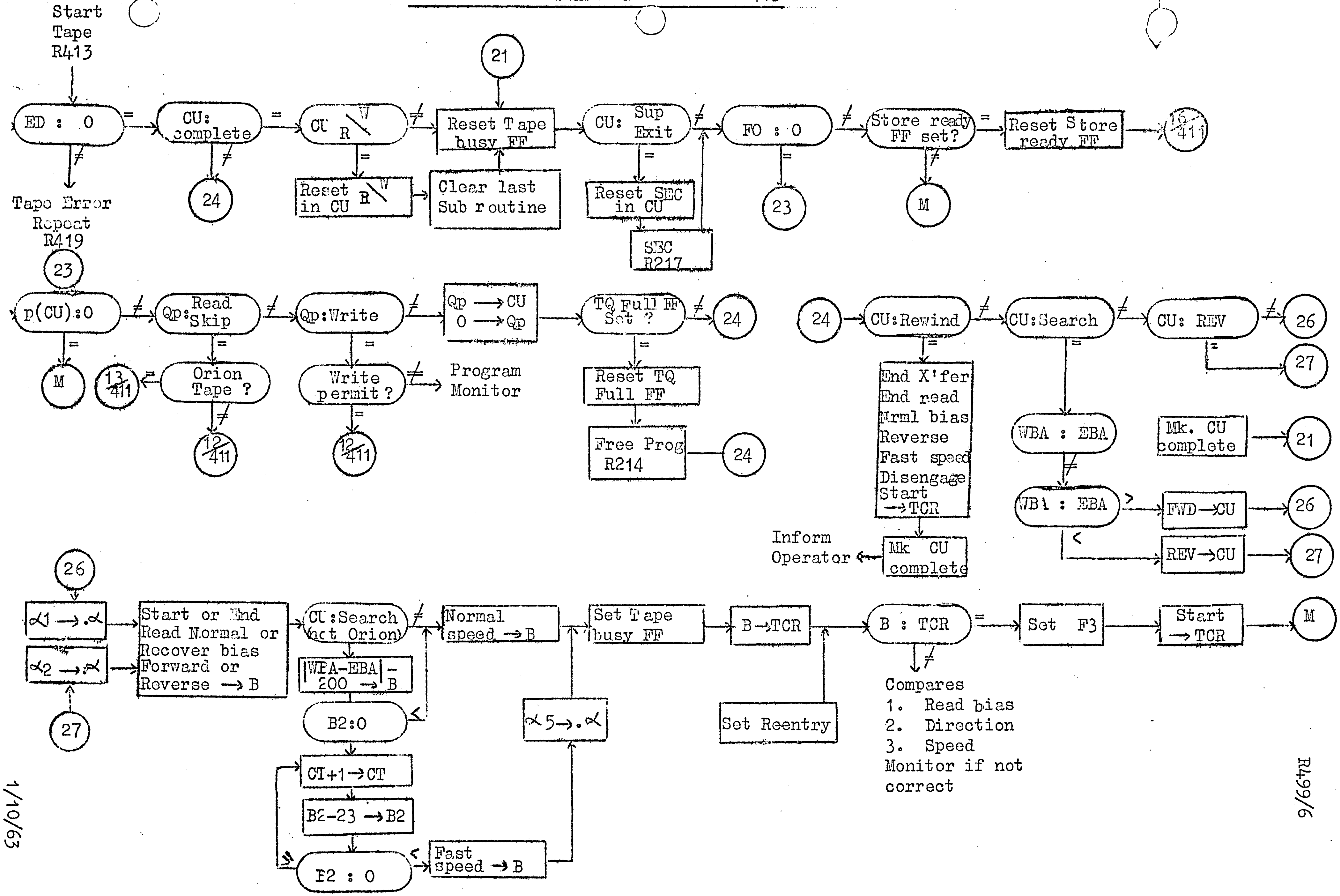
Deck Errors Includes
not 512 words transferred
check sum fail.
Deck Interrupt Reset

1/10/63

R499/5

R499/6

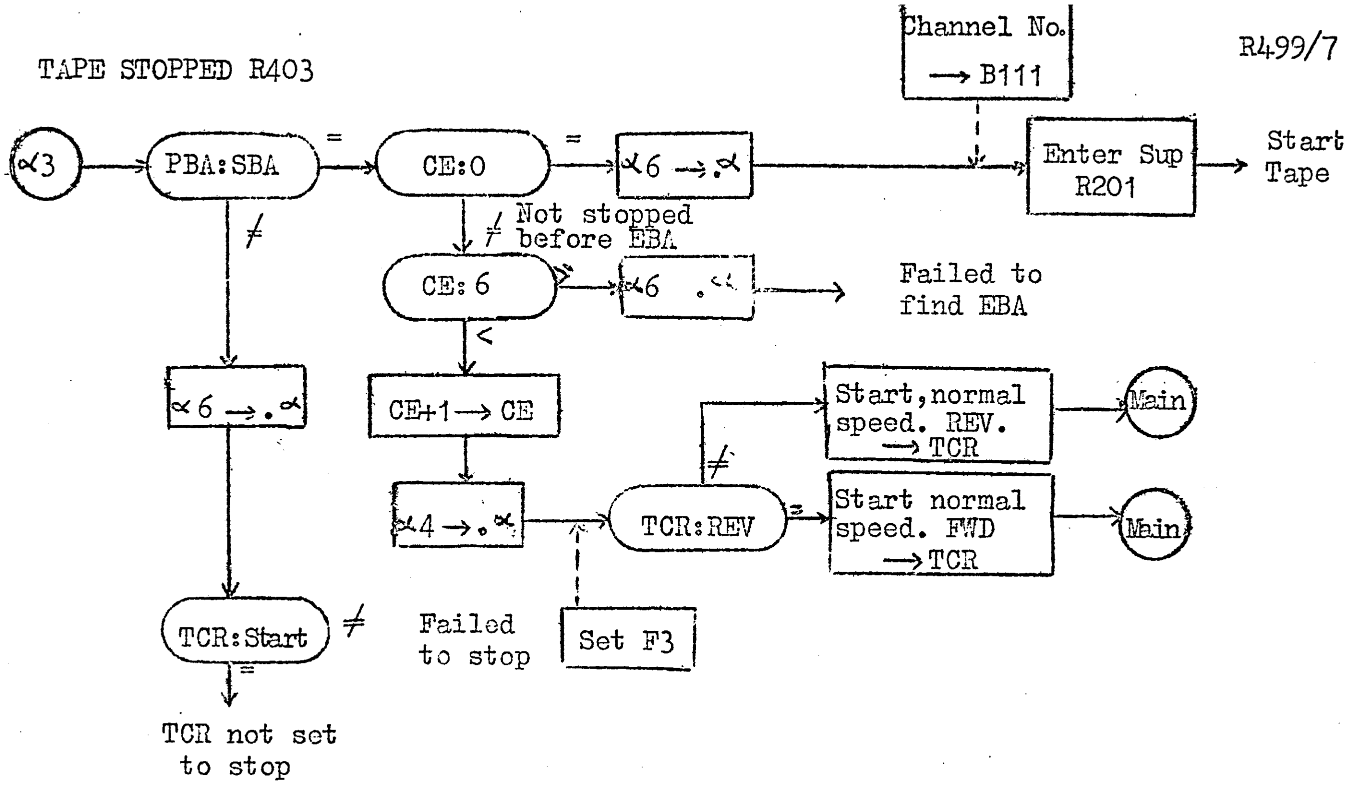
LONG INTERRUPT START TAPE ROUTINE R413



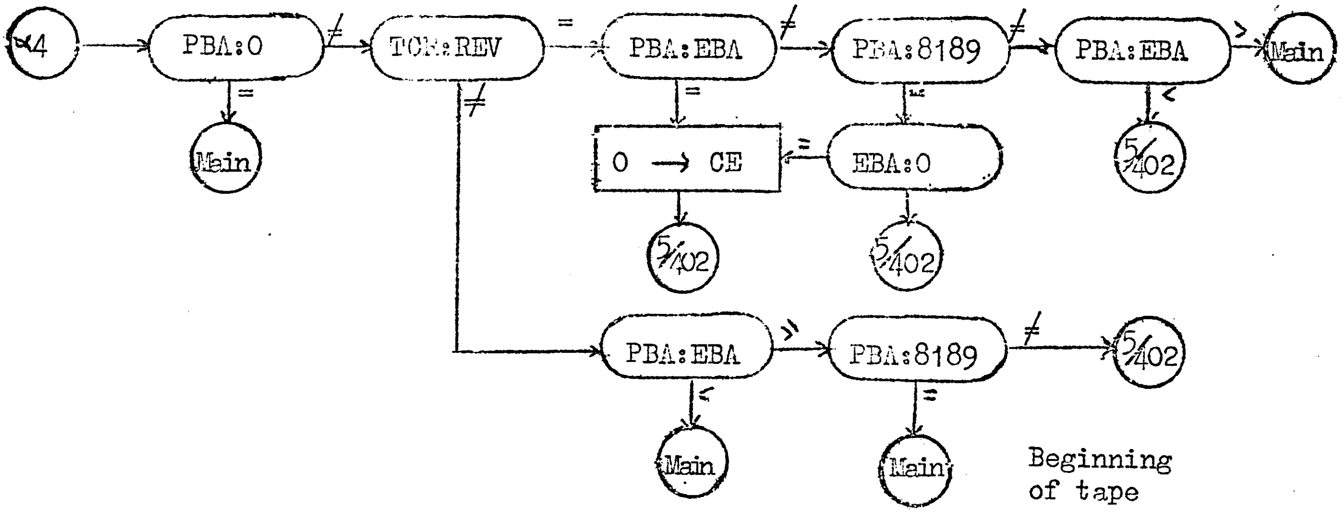
1/10/63

TAPE STOPPED INTERRUPT AND ALIGNMENT AND CALCULATION EBA INTERRUPTS R403, R405

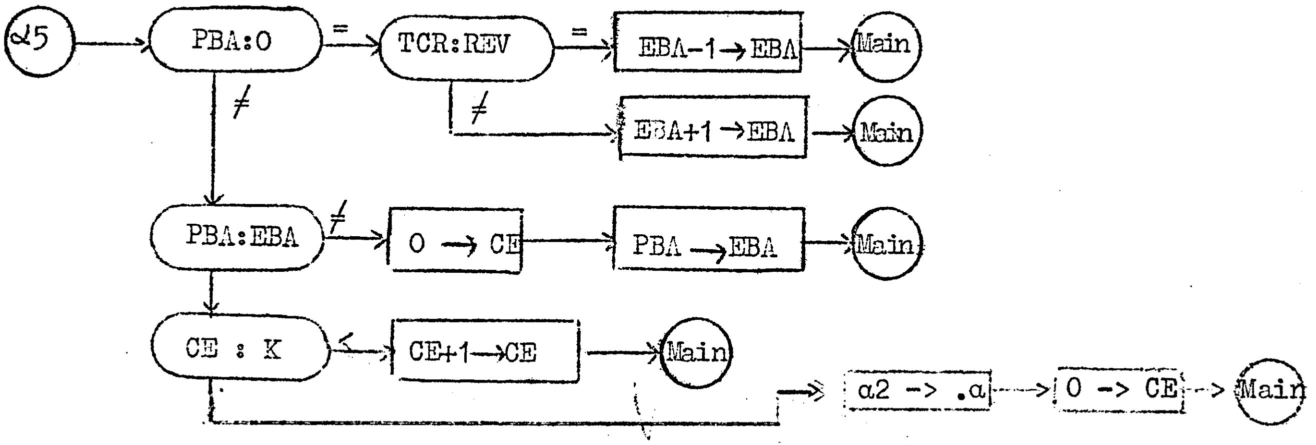
TAPE STOPPED R403



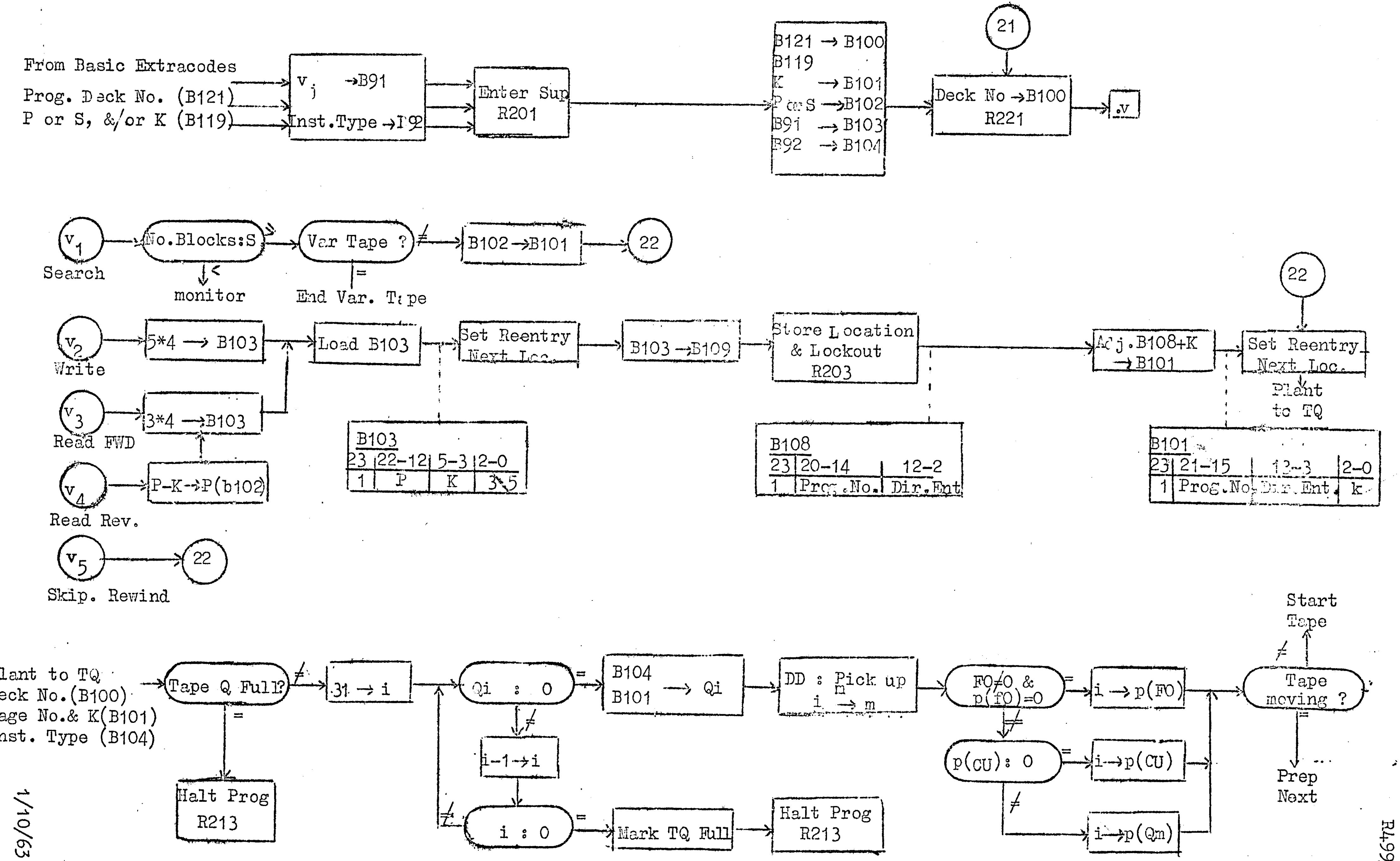
ALIGNMENT OF TAPE TO EBA AFTER AN R404



CALCULATION OF EBA AFTER AN OFF-CHANNEL SEARCH R405



BASIC TAPE ORDER TO TAPE QUEUE R421



SIMPLIFIED FLOW DIAGRAM

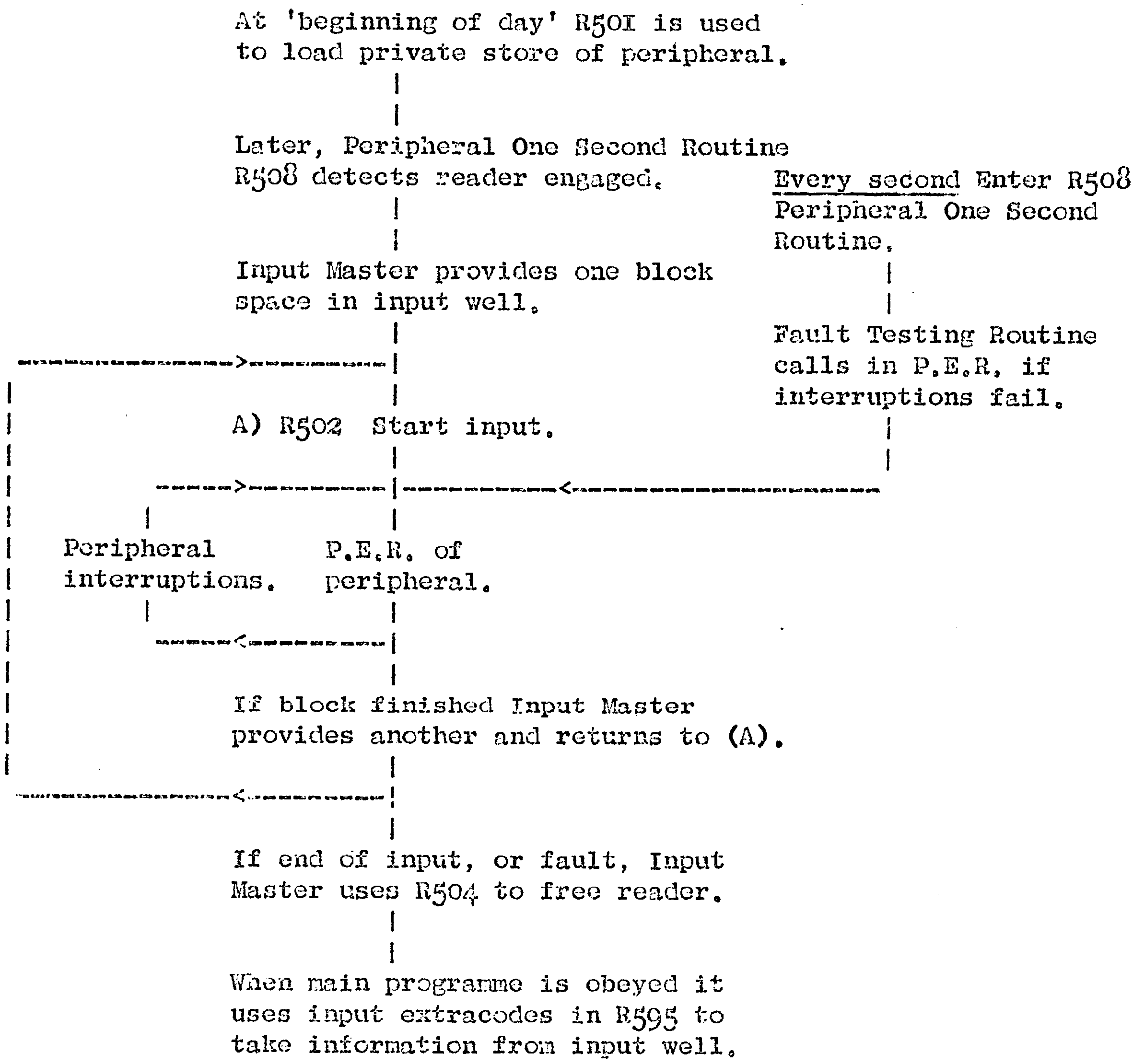
General scheme for output peripherals

```
               At 'beginning of day' R5OI is used
               to load private store of peripheral.
                         |
                         |
               Later, main programme uses output
               extracodes in R596 to fill output
               well.
                         |                         Every second Enter R508
                         |                         Peripheral One Second
               When main programme is finished,    Routine.
               Output Master takes first block             |
               from output well.                           |
                         |                         Fault Testing Routine
---------------->--------|                         calls in P.E.R. if
|                        |                         interruptions fail.
|              A) R503  Start output.                      |
|                        |                                 |
|         ----->---------|----------<-----------------------
|         |              |
|   Peripheral      P.E.R. of
|   interruptions.  peripheral.
|         |              |
|         -----<---------|
|                        |
|              Block finished.
|                        |
|                        |
|              Output Master: If there is another
|              block in output well, return to (A).
|                        |
---------------<---------|
                         |
               If no more in output well, use
               R504 to free peripheral.
                         |
                         |
               Peripheral is now available for output
               from next main programme.
```

2.9.63

SIMPLIFIED FLOW DIAGRAM

General scheme for input peripherals



At 'beginning of day' R50I is used to load private store of peripheral.

Later, Peripheral One Second Routine R508 detects reader engaged.

Input Master provides one block space in input well.

A) R502 Start input.

Every second Enter R508 Peripheral One Second Routine.

Fault Testing Routine calls in P.E.R. if interruptions fail.

Peripheral interruptions.

P.E.R. of peripheral.

If block finished Input Master provides another and returns to (A).

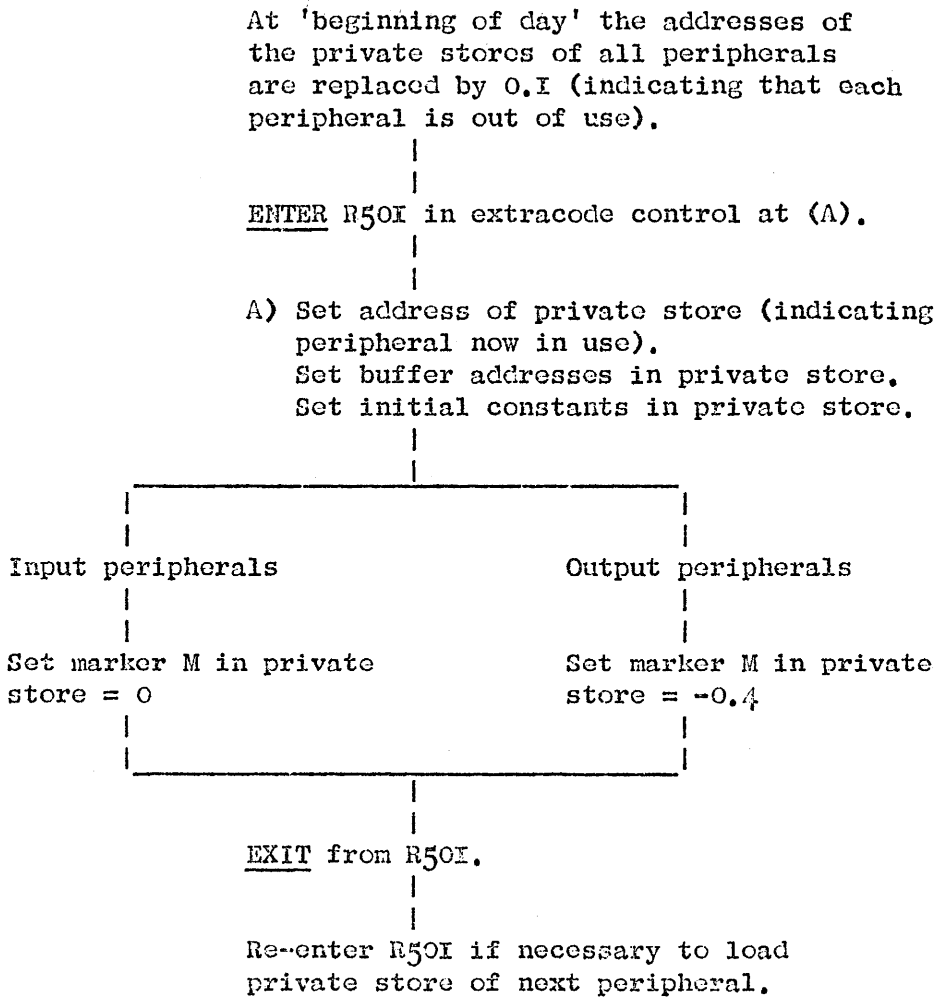
If end of input, or fault, Input Master uses R504 to free reader.

When main programme is obeyed it uses input extracodes in R595 to take information from input well.

2.9.63

SIMPLIFIED FLOW DIAGRAM

R50I Load private store of any peripheral



At 'beginning of day' the addresses of the private stores of all peripherals are replaced by 0.I (indicating that each peripheral is out of use).

ENTER R50I in extracode control at (A).

A) Set address of private store (indicating peripheral now in use).
Set buffer addresses in private store.
Set initial constants in private store.

Input peripherals

Set marker M in private store = 0

Output peripherals

Set marker M in private store = -0.4

EXIT from R50I.

Re-enter R50I if necessary to load private store of next peripheral.

2.9.63

SIMPLIFIED FLOW DIAGRAM

R502 Start any input peripheral

(If the Input Master routine requires to ignore any characters which may already be in the input buffer of this peripheral, it should have used R50I or R504 before entering here).

|

|

ENTER R502 in extracode control at (A) from Input Master.

|

|

A) Set limits of main store area to be used for reading into.

|

|

Set code conversion required.

|

|

EXIT to P.E.R. appropriate to peripheral type (i.e. go to R533).

2.9.63

SIMPLIFIED FLOW DIAGRAM

R503 Start any output peripheral

(If the Output Master routine requires to ignore any characters which may already be in the output buffer of this peripheral it should have used R50I or R504 before entering here).

ENTER R503 in extracode control at (A) from output master.

A) Set starting address of characters to be printed from main store.

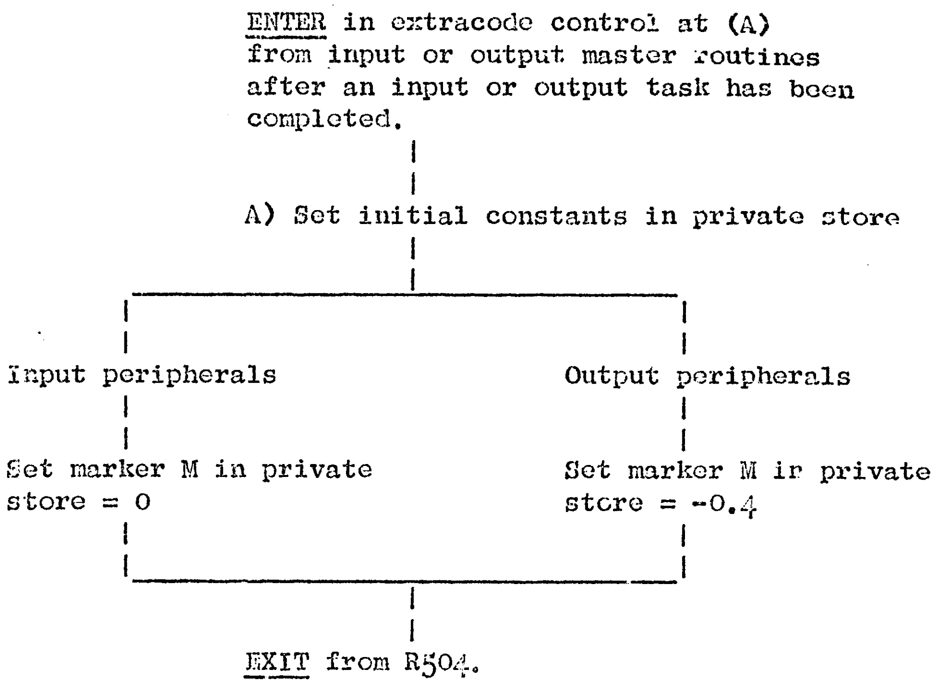
Set code conversion required

EXIT to P.E.R. appropriate to peripheral type (i.e. go to R553 or R573 or R579).

2.9.63

SIMPLIFIED FLOW DIAGRAM

R504 Free any peripheral



ENTER in extracode control at (A) from input or output master routines after an input or output task has been completed.

A) Set initial constants in private store

Input peripherals

Set marker M in private store = 0

Output peripherals

Set marker M in private store = -0.4

EXIT from R504.

2.9.63

SIMPLIFIED FLOW DIAGRAM

R508 Peripheral One Second Routine

ENTER in extracode control, from
One Second Interruption, at (A)

A) Test state of peripheral
Go to (B) or (C).

B) Not in use,
or Disengaged,
or $M > 0$
or $M = -0.4$

This peripheral needs
no attention,
Go to (H).

C) In use (i.e. private store has been set
up by R501).
and Engaged
and $M \leq 0$ (but $\neq -0.4$)

Use Fault Testing Routine appropriate to
peripheral type to examine state of peripheral
in greater detail (i.e. go to R530 or R540 or
R550 or R560 or R565 or R570 or R575 or R585).
Return to (D) or (E) or (F) or (G).

D) Input peripheral stopped, but with no fault;
Set $M = -0.4$
Insert Input Master in S.E.R. queue to
organise starting of reader.
Go to (H).

E) Output peripheral stopped, but with no faults:
Set $M = -0.4$.
Insert appropriate P.E.R. in S.E.R. queue to
continue output where it left off.
Go to (H).

F) Any peripheral already started (and therefore
with no faults):
Leave to continue undisturbed.
Go to (H).

G) Any peripheral stopped, with fault bit showing:
Set M = +ve mark to indicate specific fault.
Insert appropriate P.E.R. in S.E.R. queue to
clear up buffer and private store before
returning to Input/Output master.
Go to (H).

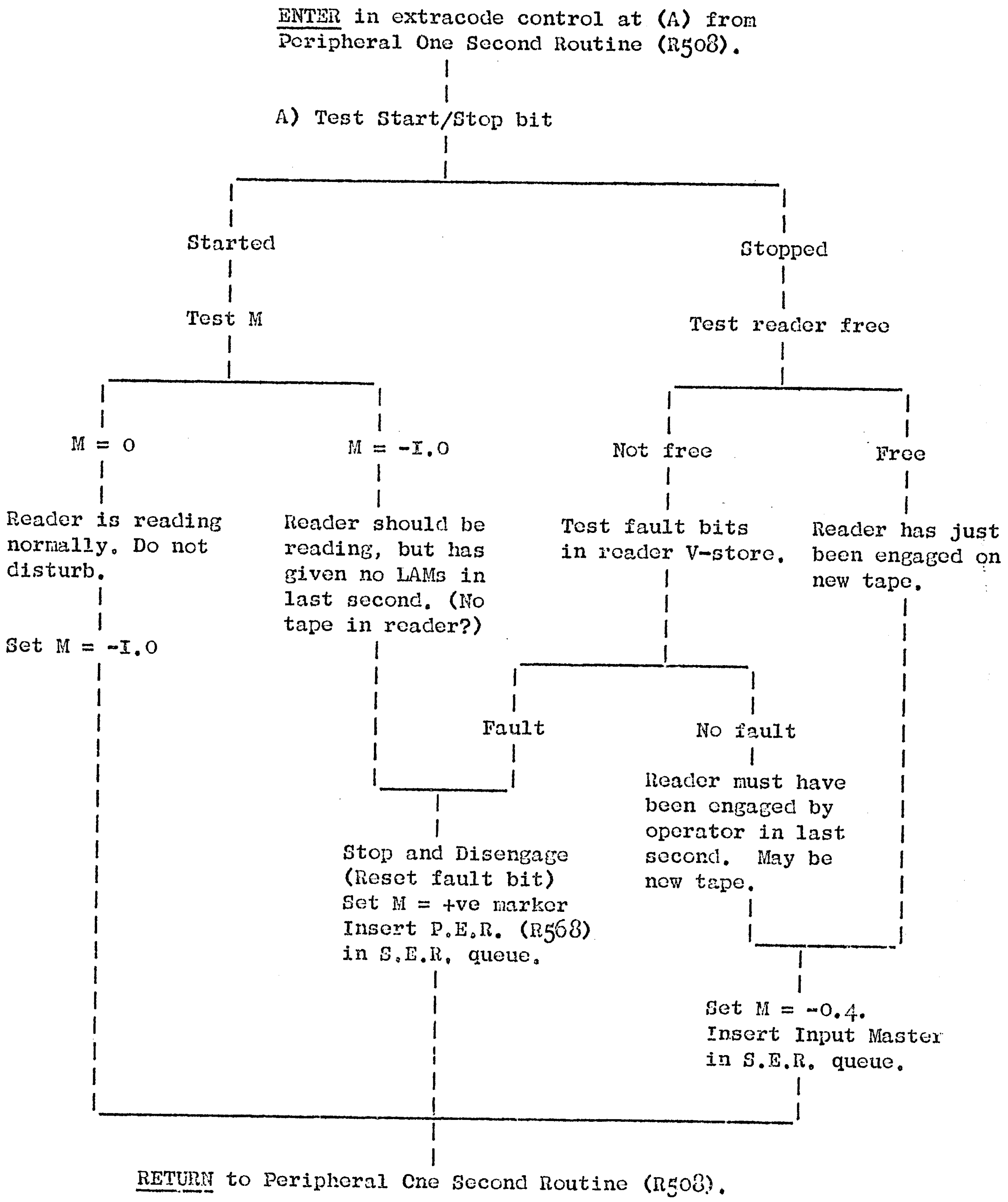
H) If there is another peripheral to
test, go back to (A).

If all peripherals have now been tested,
EXIT to next routine in S.E.R. queue
(i.e. go to R202).

2.9.63

SIMPLIFIED FLOW DIAGRAM

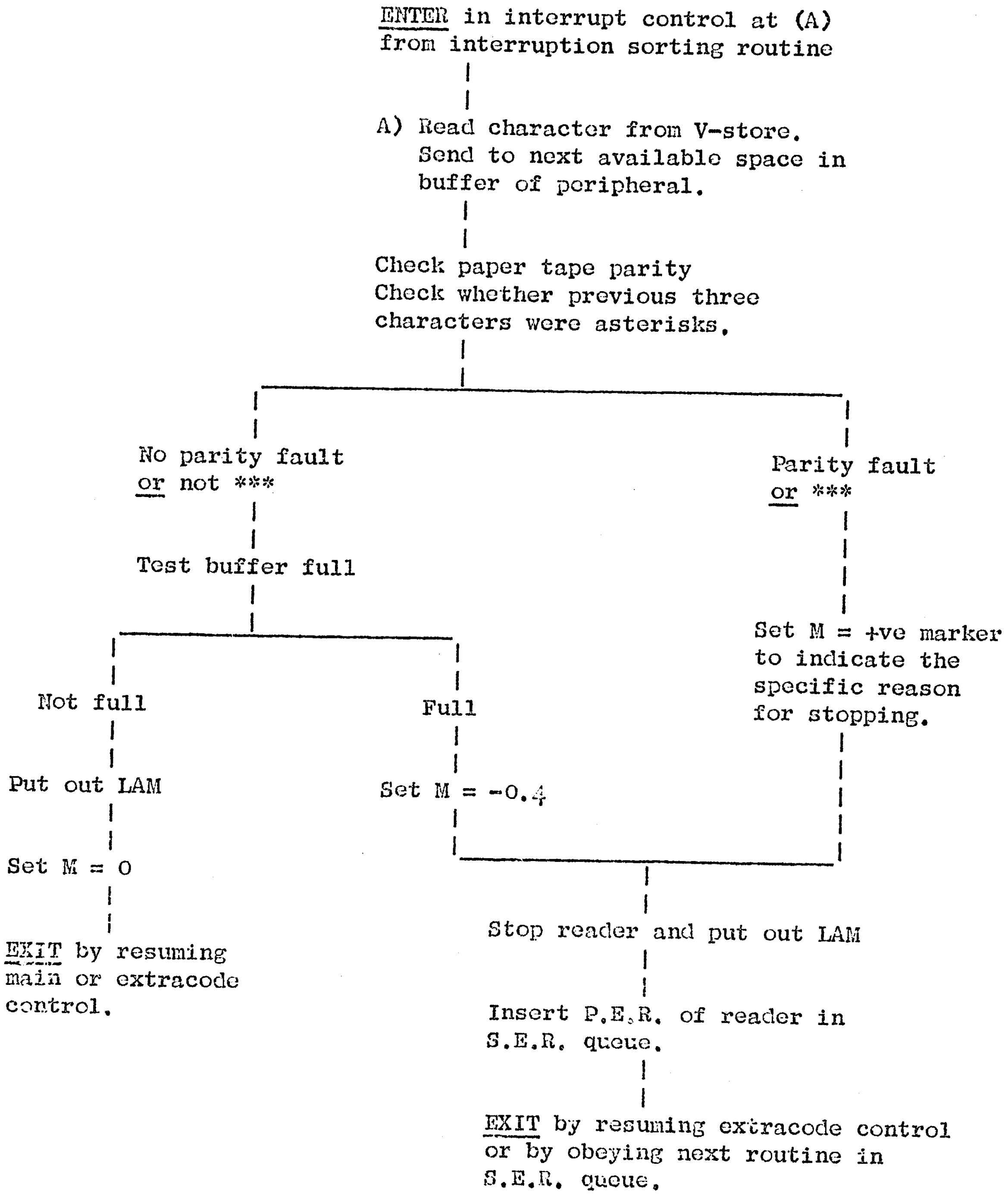
Typical reader fault testing routine (R565 for TR5 tape reader).


ENTER in extracode control at (A) from Peripheral One Second Routine (R508).
A) Test Start/Stop bit
Started
Stopped
Test M
Test reader free
M = 0
M = -I.0
Not free
Free
Reader is reading normally. Do not disturb.
Reader should be reading, but has given no LAMs in last second. (No tape in reader?)
Test fault bits in reader V-store.
Reader has just been engaged on new tape.
Set M = -I.0
Fault
No fault
Reader must have been engaged by operator in last second. May be new tape.
Stop and Disengage (Reset fault bit) Set M = +ve marker Insert P.E.R. (R568) in S.E.R. queue.
Set M = -0.4. Insert Input Master in S.E.R. queue.
RETURN to Peripheral One Second Routine (R508).

2.9.63

SIMPLIFIED FLOW DIAGRAM

Typical reader interruption routine (R566 for TR5 tape reader)



ENTER in interrupt control at (A) from interruption sorting routine

A) Read character from V-store. Send to next available space in buffer of peripheral.

Check paper tape parity
Check whether previous three characters were asterisks.

No parity fault or not ***

Test buffer full

Not full

Put out LAM

Set M = 0

EXIT by resuming main or extracode control.

Full

Set M = -0.4

Parity fault or ***

Set M = +ve marker to indicate the specific reason for stopping.

Stop reader and put out LAM

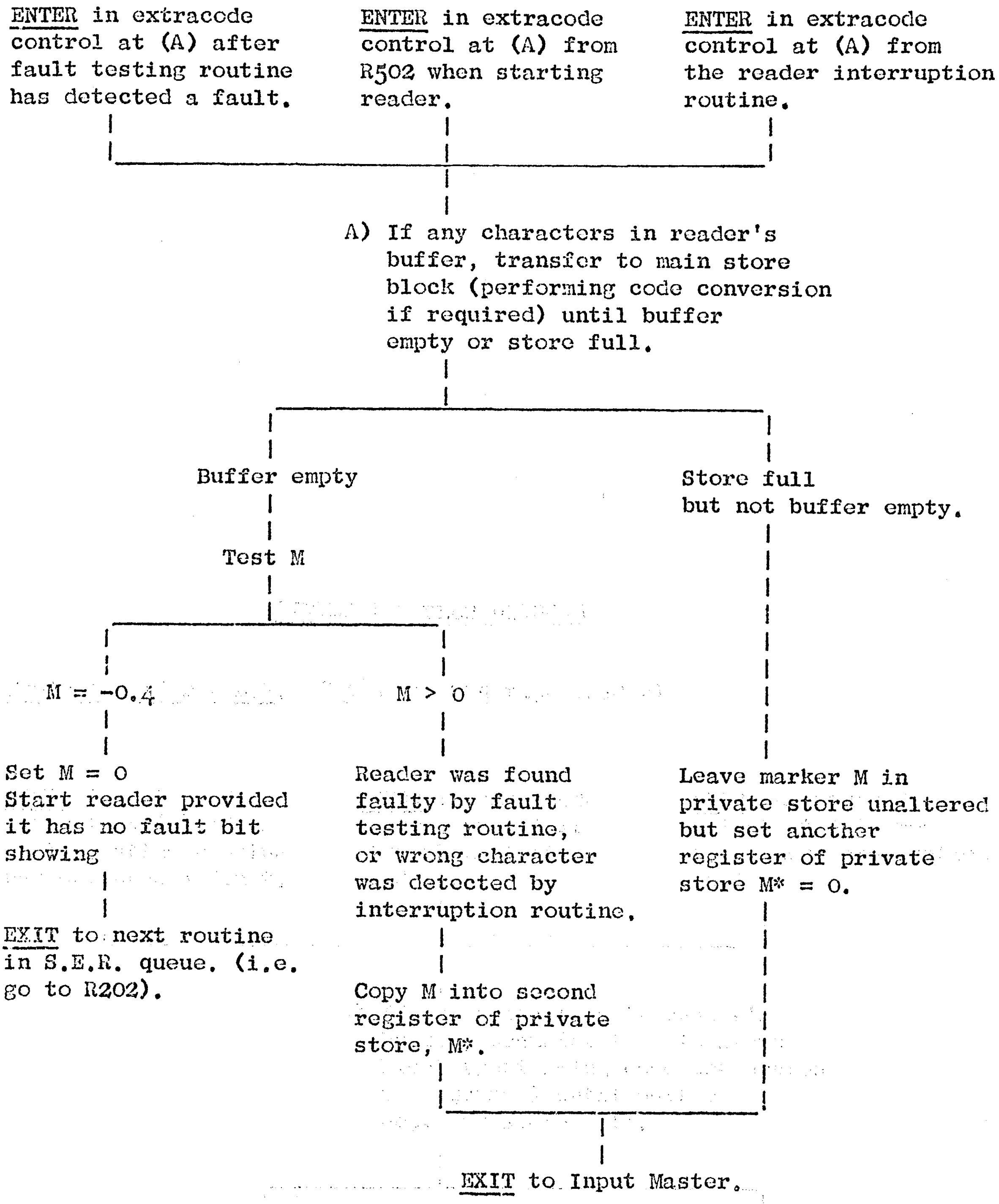
Insert P.E.R. of reader in S.E.R. queue.

EXIT by resuming extracode control or by obeying next routine in S.E.R. queue.

2.9.63

SIMPLIFIED FLOW DIAGRAM

Typical reader P.E.R. (R568 for TR5 tape reader)



ENTER in extracode control at (A) after fault testing routine has detected a fault.

ENTER in extracode control at (A) from R502 when starting reader.

ENTER in extracode control at (A) from the reader interruption routine.

A) If any characters in reader's buffer, transfer to main store block (performing code conversion if required) until buffer empty or store full.

Buffer empty

Test M

M = -0.4

Set M = 0
Start reader provided it has no fault bit showing

EXIT to next routine in S.E.R. queue. (i.e. go to R202).

M > 0

Reader was found faulty by fault testing routine, or wrong character was detected by interruption routine.

Copy M into second register of private store, M*.

Store full but not buffer empty.

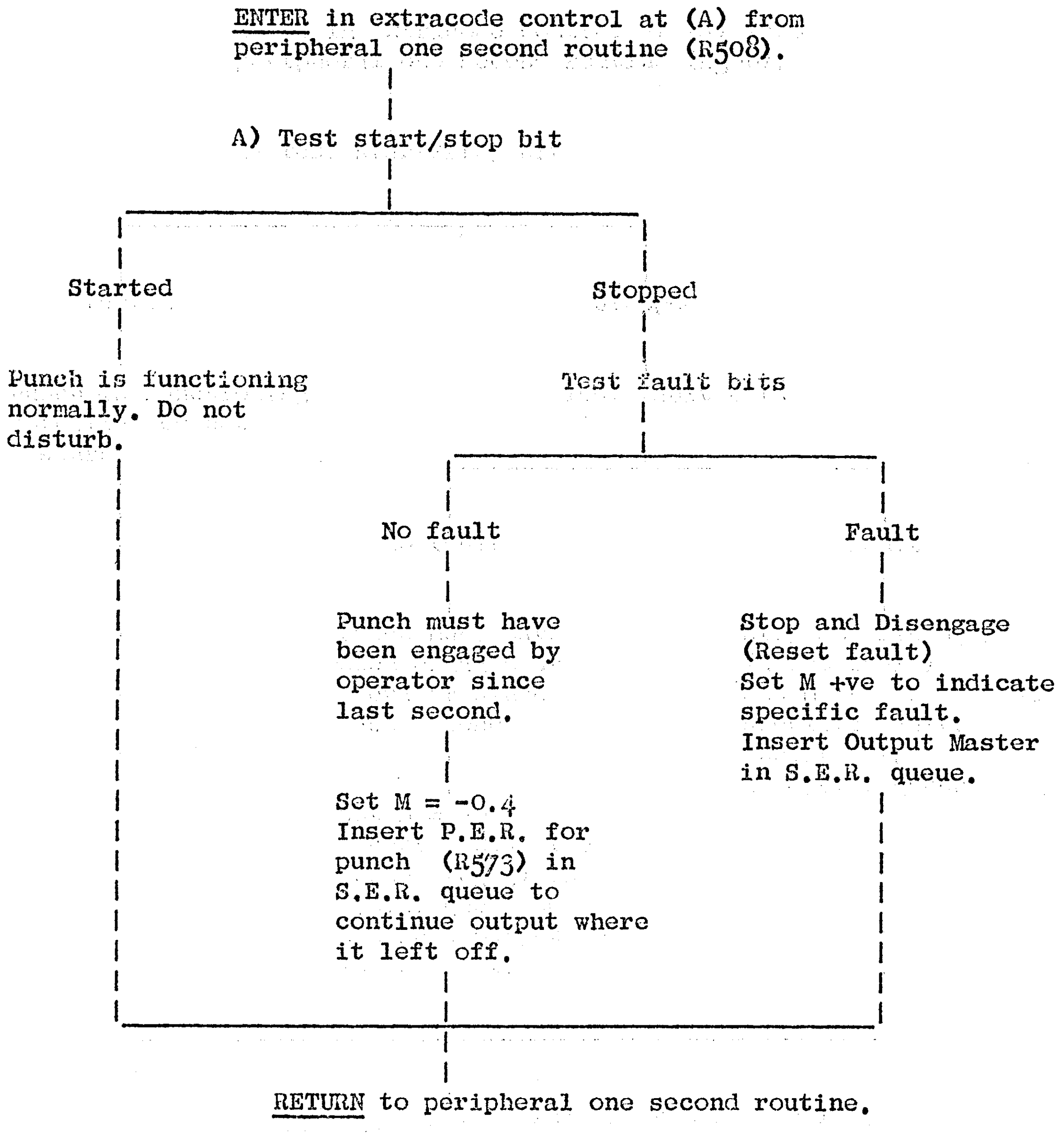
Leave marker M in private store unaltered but set another register of private store M* = 0.

EXIT to Input Master.

2.9.63

SIMPLIFIED FLOW DIAGRAM

Typical output peripheral fault testing routine (R570 for Teletype punch)



ENTER in extracode control at (A) from peripheral one second routine (R508).

A) Test start/stop bit

Started

Punch is functioning normally. Do not disturb.

Stopped

Test fault bits

No fault

Punch must have been engaged by operator since last second.

Set M = -0.4
Insert P.E.R. for punch (R573) in S.E.R. queue to continue output where it left off.

Fault

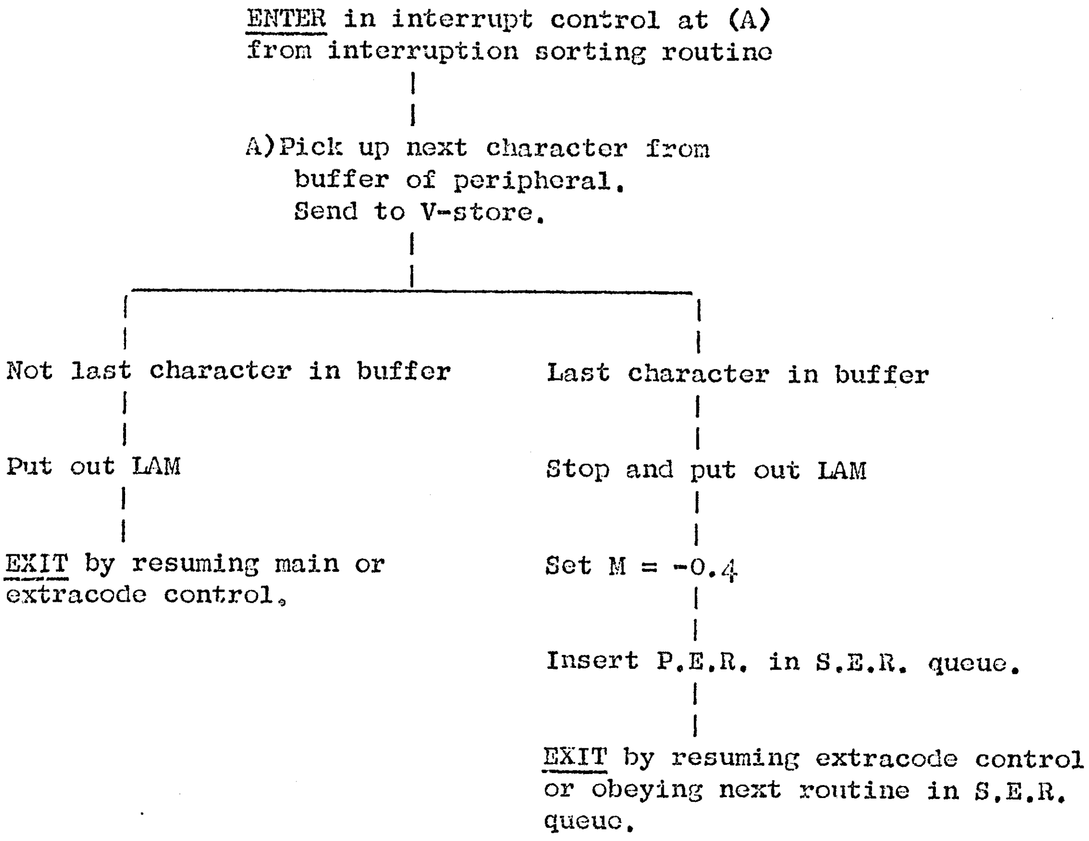
Stop and Disengage (Reset fault)
Set M +ve to indicate specific fault.
Insert Output Master in S.E.R. queue.

RETURN to peripheral one second routine.

2.9.63

SIMPLIFIED FLOW DIAGRAM

Typical output peripheral interruption (R571 for teletype punch)



ENTER in interrupt control at (A)
from interruption sorting routine

A) Pick up next character from
buffer of peripheral.
Send to V-store.

Not last character in buffer

Put out LAM

EXIT by resuming main or
extracode control.

Last character in buffer

Stop and put out LAM

Set M = -0.4

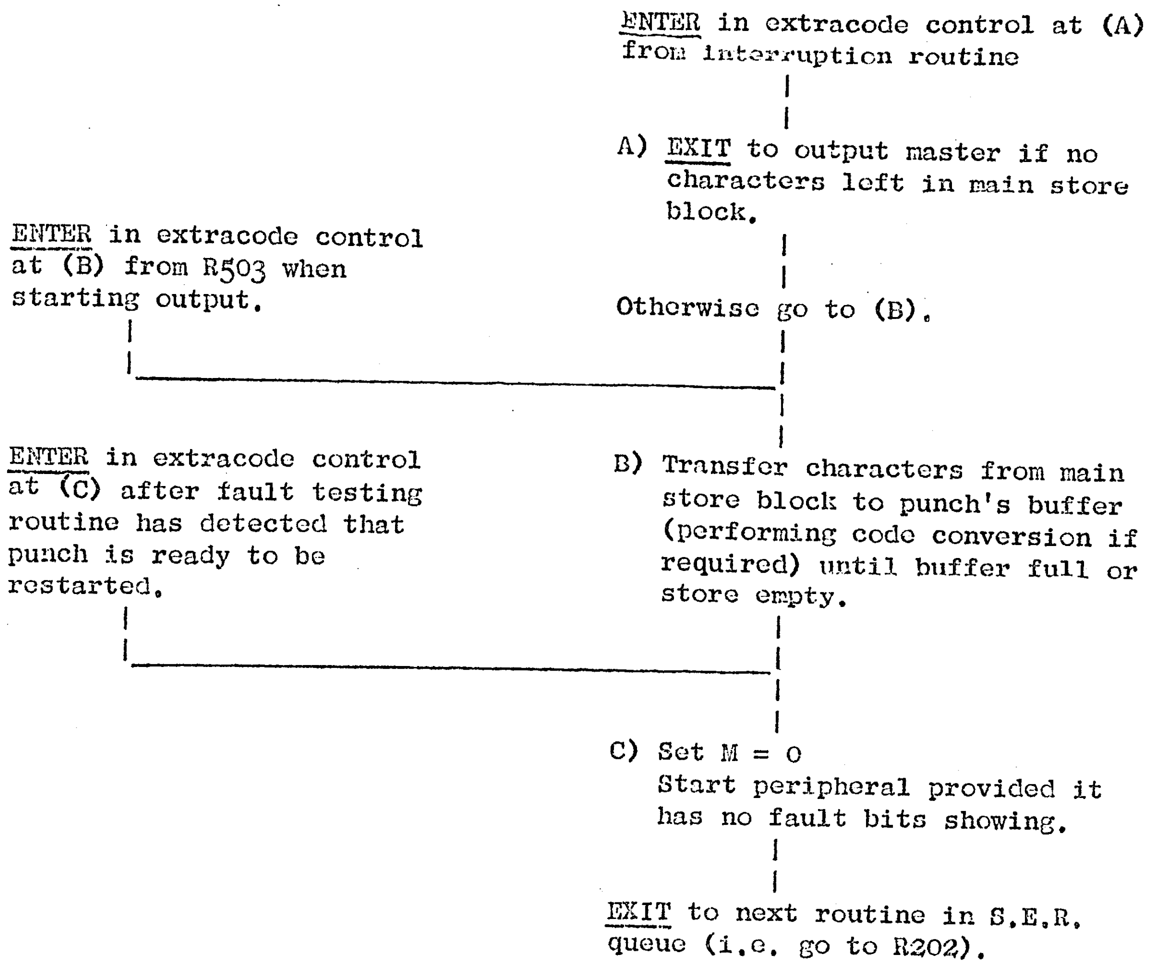
Insert P.E.R. in S.E.R. queue.

EXIT by resuming extracode control
or obeying next routine in S.E.R.
queue.

2.9.63

SIMPLIFIED FLOW DIAGRAM

Typical output P.E.R. (R573 for Teletype punch)



ENTER in extracode control at (A) from interruption routine

A) EXIT to output master if no characters left in main store block.

Otherwise go to (B).

ENTER in extracode control at (B) from R503 when starting output.

B) Transfer characters from main store block to punch's buffer (performing code conversion if required) until buffer full or store empty.

ENTER in extracode control at (C) after fault testing routine has detected that punch is ready to be restarted.

C) Set M = 0
Start peripheral provided it has no fault bits showing.

EXIT to next routine in S.E.R. queue (i.e. go to R202).

R50I Load private store of any peripheral.

Purpose A routine for use 'at the beginning of a day' to allocate private store to a peripheral and load it with the necessary initial constants.

Registers of Fixed Store: 77.

Enter at (I) with:

BI00 containing the address where the private store is to be (less *7).
BI0I containing the beginning address of the buffer to be used by this peripheral.
BI02 containing the end address of the buffer (i.e. next half word beyond the last).
BI09 containing the V-store address (less *6).
BII0 containing return address.

Exit: To address in BII0 with, in general:

(50) containing zero.
(5I) containing -0.4
(52) containing zero.
(53) containing zero.
(56) containing zero.
(60) containing starting address of buffer (less *7)
(6I) containing end address of buffer (less *7)
(62) containing starting address of buffer (less *7)
(66) containing zero.
(67) containing starting address of buffer (less *7)
(68) containing the V-store address (less *6)

Alternative exit conditions depending on peripheral type.

TR5 and TR7 Private store, as above except
(50) containing 0.I
(5I) containing zero.

Card Reader Private store, as above except
(5I) containing zero.

Teletype punch, Creed 3000 and teleprinter. Private store, as above.

Card Punch Private store, as above except
(67) containing zero

Anelex printer. Private store, as above except:
(50) containing *5
(60) containing 0.
(6I) containing 60.
(67) as (60)

Working Space: BIOO to BIO2, BIO6 to BIIO.

Subroutines: R509, R599.

Notes:

I. If an input peripheral; the peripheral is now ready for use when the operator engages it. When the operator does engage the reader the Peripheral One Second Routine, R508, causes an entry to the Input Master Routine. The Input Master Routine uses R502 to initiate input. When input is finished, the reader remains reserved. The Input Master Rout may then use R504 to return the peripheral to its present state.

2. If an output peripheral; the peripheral is now ready for use and will be started as soon as

(a) it is engaged by the operator
and
(b) R503 has been entered to initiate output.

The peripheral will return to the present state when the output has been finished, unless there have been any faults on the peripheral. It is better if the Output Master Routine uses R504 to ensure returning the peripheral to its present state.

3. The buffer will generally be in blocks 3663 and 3664, but any other blocks can be specified if required.

4. If the buffer is in main store, then before the peripheral is started, it is essential to lock the chosen block down on the cores.
R5OI does not do this.

R502. Start reading from any input peripheral.

Purpose Provides entries to routines concerned with the chosen input peripheral. These routines, themselves, start and stop the reader as necessary and transfer information to the store area specified.

Registers of fixed store: 34

Entries: Under extracode control at (2),(3) or (4)

Enter at (2) to start the reader with

BI00 containing address of private store (less *7)
BI0I containing address of first half word space in store (2 l.s. bits ignored)
BI02 containing end address of store space, i.e. next after last character
BI03 containing information on the code conversion as subsequently required by the P.E.R. of the peripheral.
BI04 containing information on action required if a punching fault:
(20) if require to stop.
(2I) if require to mark faulty character but continue.
BII0 containing return address.

Enter at (3) to specify a new area of store and to read into it with the same code conversion as used at last entry, with

BI00, BI0I, BI02, BII0, as above.

Enter at (4) to change the end address of the store being used previously, and to continue reading into it with the same code conversion as before, with

BI00, BI02, BII0, as above.

Exit from R502 is to the P.E.R. of the peripheral concerned. Before exit to the P.E.R., the S.E.R. current entry address is updated to an address in this routine.

Return to Master Routine is from R5I4 under extracode control to the address given in BII0.

BI00 contains address of private store (less *7).
Other information is in private store (see specification of R5I4)

Working Space All supervisor B lines (see Note I)

Subroutines. Many of the peripheral routines R500 to R599 are used, following an entry to this routine. Also the co-ordinator routines R20I, R202, R206 and drum routines, R3I2, R3I8.

2.9.63

<u>Private store:</u> Before entry to this routine; the private store should contain some or all of the following information:

(50) containing information on the last four characters to have been read by the reader.

(5I) containing -0.4 or > 0 indicating why the reader is at present standing stopped.

(52) containing information on the code conversion (if any) currently being used for input by this peripheral.

(53) bits 23-2 containing count of punching errors detected since last freeing the reader (R504)
bit 0 containing I if current requirement is to stop when a punching error is detected on the input.

(56) containing address reserved for separator (less *7) or zero if a new record has not been started yet.

(60) containing starting address of input buffer (less *7)

(6I) containing end address of input buffer (less *7)

(62) containing end address (less *7) of the information at present lying in the input buffer.

(64) containing the character address in the (main) store immediately after the address of the character which was the last to be transferred there from the input buffer.

(65) bits 23-I2 containing *4667 (= *3667 less *7) if (64) refers to main store block: alternatively containing same as bits 23-I2 of (64) (less *7) if (64) refers to subsidiary store.
bits II-0 containing end address of space in the current (main store) block.

(66) bits 23-2 containing information on why a return was made via R5I4 the last time that there was an entry to this routine.

(67) containing address (less *7) of the next character in the input buffer which is due to be transferred to to main store.

(68) containing the V-store address of the peripheral (less *6)

For entry at (2) the private store should contain,
(50),(5I),(60),(6I),(62)(66),(67),(68), as above.

For entry at (3) the private store should contain,
(50),(5I),(52),(53),(60),(6I),(62),(66),(67),(68)
as above.

For entry at (4) the private store should contain,
(50),(5I),(52),(53),(56),(60),(6I),(62),(64),(66),
(67),(68), as above.

For entry at (5) the private store should contain all information as above.

On exit from R502 to the P.E.R. of the peripheral, the private store will have been updated, as above, except:

(66) containing the address for return when input is complete.

*/

Notes:

1. In the course of reading, control may pass to 'Program Scan' (R202) and thence to any other routine. In addition to using all supervisor B lines, the main store blocks which were on core store pages at entry to the routine may therefore have gone back to the drum by the time return is made.

2. The private store of the peripheral which is being started must be on cores (main or subsidiary) at the time of entry to R502. If main store, it must be locked down. It must also be unlocked, unless lockout is ignored. The buffer must be on cores (main or subsidiary). If in main core store it must be locked down but need not be unlocked.

3. Input can only be to one main store block at a time. The block address digits in BI02 at entry are therefore ignored, if main store, and assumed equal to those in BIOI.

4. When entering, the first half word of input, whether internal code or binary, is necessarily a separator. Thus, in particular, if only one character is to be read, five character spaces must be provided.

5. Whichever entry is made to R502 the first characters to be read will be those which are waiting in the buffer and have not yet been transferred to the (main) store. If these characters are to be ignored, then the Input Master Routine should first free the reader (R504) and then either allow the 'One Second' routine to detect that the reader is waiting engaged, or alternately, enter R502 directly at (2)

6. When the return is made (via R5I4) the next half word beyond the end of the last record is cleared. This is to indicate to the input extracodes that the end of the information has been reached. The address of this half word, contained in register (64) on return, and should be used as the starting address if further input to the same block is to be continued.

R503 Start writing to any output peripheral.

Purpose Provides entry to routines concerned with the chosen output peripheral. These routines themselves start and stop the peripheral as necessary and transfer information to it from the store.

Registers of fixed store 9

Entries at (7) under extracode control, with:

BI00	containing address of private store (less *7)
BI0I	containing address of first separator in output.
BI03	containing a number depending on the output peripheral and the particular code required.
BII0	containing return address.

Exit is to the P.E.R of the peripheral concerned.

Return to Output Master Routine is from R5I4, under extracode control to the address which was given in BII0

BI00 contains address of private store (less *7)
Other information is in private store, (see specification of R5I4).

Working Space: All Supervisor B lines (see Note I).

Subroutines: Many of the Peripheral Routines R500 - R599 are used following an entry to this routine R503. Also the Co-ordinator routines R20I, R202, R206 and drum routines R3I2, R3I8.

Private Store Before entry to this routine the private store should contain the following information:

(5I)	containing -0.4 or >0
(52)	containing information on the code conversion (if any) currently being used for output by this peripheral.
(60)	containing starting address (less *7) of the output buffer.
(6I)	containing end address (less *7) of the output buffer.
(62)	containing the address (less *7) of the character in the output buffer which will be sent out at the next interruption.
(67)	containing the address in the buffer (less *7) of the space available for the next character to be transferred from the (main) store.
(68)	containing the V-store address (less *6) of the peripheral.

On exit from R503 to the P.E.R. of the peripheral, the private store will have been updated, as above, and in addition:

(50)	containing zero
(56)	containing zero
(64)	containing the address of first separator in the output.
(66)	contains the address for return when output is complete.

Notes

I. During output, control may pass to 'Program Scan' (R202) and thence to any other routine.
In addition to using all supervisor B lines, the main store blocks which were on core store pages at entry to the routine may, therefore, have gone back to the drum by the time the return is made.

2. Output must be of at most one main store block at a time.

3. Output continues until a separator is found containing all bits zero.

4. Before an exit is made to the P.E.R. of the peripheral, the S.E.R. 'current entry address' is updated to an address in this routine.

R504 Free any peripheral.

Purpose Resets the private store and buffer of any peripheral to the value appropriate 'at the beginning of the day', and in the case of an input peripheral, indicates to R508 'Peripheral one second' that the input master routine should be called in when the operator next engages it.

Entry at (I) with

BIOO	containing address of private store (less *7)
BIIO	containing return address.
(60)	containing address of beginning of buffer (less *7)
(6I)	containing end address of buffer (less *7)
(68)	containing V-store address of peripheral (less *6)

Exit to address in BIIO with:

BIOO, BIIO unaltered.
Private store (50) to (58) as at exit from R50I.

Working Space BIOO, BIO8 to BIIO.

Subroutines R50I, R599.

Notes: Normally, when a peripheral has been finished with, most of the private store is already in the appropriate condition for it to be started again. In that case, the only purpose of entering this routine is to reset the code conversion parameter in (52) so that the next start will be made with the peripheral in its normal code shift and the internal code beginning a new record. In addition for input peripherals (5I) is set to zero as an indication to R508 'peripheral one second' that the reader is no longer being used.
After a peripheral fault, eg. 'disabled' or 'overdue', this routine should be entered to clear any existing information in the private store or buffer, ready for a fresh start.

R508. Peripheral One Second.

Purpose Test all peripherals for Disengaged, Calls Input Master Routine to S.E.R. queue for each input peripheral which is free and has just been engaged. Arranges, in the S.E.R. queue, for a return to be made via R5I4 for any peripheral which has been stopped by a fault condition.

Registers of fixed stores: 34

Entry at (I)
At entry, the table of peripheral private store addresses must contain, for each peripheral, either
a) bit 0 = I : in which case the peripheral is ignored
or
b) bit 23 -2 containing the address of the private store (less *7), and
Bit 0 = 0 : in which case the peripheral is tested and the private store must contain:

(5I) containing a marker indicating the state of use of the peripheral.
(66) containing zero if the peripheral is free; non zero otherwise.
(68) containing V-store address (less *6)
Other information in private store as required by the 'fault testing' routine of the peripheral type concerned.

Exit To R202 ' Program Scan'

Working Space BIOI to BIIO.

Subroutines R202, R206, R509, R599.
All peripheral 'Fault Testing' routines.

Notes: When a peripheral is to be tested (i.e. bit 0 if its private store address is zero) then the following test is performed:

Ignore peripheral if disengaged.
Ignore peripheral if the marker in (5I) is either
a) >0 : indicating that some fault condition which was recognised previously has not yet been attended to by the input or output master routine.
or
b) =-0.4 : indicating that the peripheral has been stopped intentionally by its own interruption routine, and that its P.E.R. is now standing in the S.E.R. queue.

(continued)

R508 continued...

If the peripheral is still being considered after this test, then its 'Fault Testing' routine is entered with interruptions inhibited and:

BI00 containing address of private store (less *7)
BI02 containing V-store address of peripheral (less *6)
BI03 containing the marker found in (5I) (negated)
BI04 containing zero if the peripheral is free; non zero otherwise.

Return to the Peripheral One Second from the Fault Testing routine of the peripheral is to (7),(8) or (9).

Return to (7) with:
BI00 unaltered.
BI08 containing the new value to be set in (5I)
BI09 containing the entry address of a peripheral routine which is required to be inserted in the peripheral S.E.R. queue.

Return to (8) with:
BI00 unaltered.
BI08 as above, when only (5I) is to be reset and no address is to be inserted in the S.E.R. queue.

Return to (9) when (5I) is not to be altered and no address is to be inserted in the S.E.R. queue.

Interruptions are permitted again soon after returning to any of these return addresses.
The fault testing routine must not alter BI0I.

R509 Find peripheral type.

Purpose Given the private store address; to find the type number of a peripheral.

Registers of fixed store 7

Entry At (1) with:
B100 containing address of private store (less *7)
B110 containing return address.
(68) containing V-store address (less *6)

Exit To address in B110 with:
B109 bits 23-6 all zero.
bits 5-2 containing the type number.
bit 1 containing 0 for peripherals 0-7 of that type.
1 for peripherals 8-15 of that type.

Working Space B100, B109, B110

Subroutines R599

R5II Find store length available.

Purpose To find how much space remains in the block currently being filled by an input peripheral and to reserve space for the separator if a new record is to be started.

Registers of fixed store I2

Entry at (I) with:

BIOO containing address of peripheral private store(less*7)
BIOI containing address of next available character space in the current block (less *7)
BIIO containing return address.
(56) containing address reserved for separator(less*7) or zero if a new record is being started.
(65) containing end address of space in current block (less *7)

Exit to address in BIIO with:

BIOO unaltered.
BIOI updated if space has been reserved for a separator.
BIO5 containing number of half word spaces left in current input block,
BIIO unaltered.
(56) containing address reserved for separator (less *7) or zero if there was not enough space left to start a new record in the current block.
(65) unaltered.

Working Space BIOO, BIOI, BIO5, BIO6, BIIO

Subroutine R599

Note Unless the record has already been started before entering this routine, then the half word whose address appears in BIOI on entry is set to zero before exit.

R5I2 Shift up character in half word.

Purpose To shift any specified 6 bit character to the most significant end of a half word.

Registers of fixed store I4

Entries at (I) to shift the specified character to the most significant end of a half word.

BI07 containing the character to be shifted.
BII0 bits 23-3 containing return address.
bits I, 0 containing character position in half word.

or

at(2) when the 2 least significant bits of each character are to be given an additional 6 bit shift towards the most significant end.

BI07 as above.
BII0 as above.

Exit to address in BII0 with:

BI06 bits 3, 2 containing the number of upward 6 bit/shifts that would be required to restore the original character position(excluding the additional shift imposed by entry (2))
BI07 containing characters in the new positions.
BII0 unaltered.

Working Space BI06, BI07, BII0.

Subroutines None.

2.9.63

*

R5I3. Restore character positions.

Purpose: To shift the m.s. 6 bit character in a half word to any specified character position within the half word.

Registers of fixed store: 8

Entries: at (I) or (2) depending on exit required.
BI07 containing 6 bit characters.
BI08 bits I, 0 containing position which the m.s. character is required to occupy.
BII0 containing return address.

Exits a) Normally to address in BII0 with:
BI07 containing characters in required position.
BII0 unaltered.

b) In special case when entry is at (I) and no shifts are required, exit is to address in BII0 + I.

Working Space: BI07, BI08, BII0

Subroutines: None.

Private store: Other information in the private store at this stage is:

a) Input peripherals.

(50) containing information on the last four characters to have been read by the reader.

(5I) containing an indicator of the reason why the reader is now stopped.

(52) containing information on the code conversion currently being used for input by this peripheral.

(53) bits 23-2 containing count of punching errors detected by the reader since last freeing the reader (R504)

bits 0=I if current requirement has been to stop when a punching error is detected on the input.

(56) containing zero.

(60) containing starting address of input buffer (less *7)

(6I) containing end address of input buffer (less *7)

(62) containing end address (less *7) of the information at present lying in the input buffer.

(64) containing the character address in the (main) store immediately after the address of the character which was the last to be transferred there from the input buffer.

(65) bits 23-I2 containing *4667(=*3667 less *7) if (64) refers to a main store block: Alternatively containing same as bits 23-I2 of (64) (less *7) if (64) refers to the subsidiary store.

bits II-0 containing end address of space in the current (main store) block.

(66) see above under exit conditions.

(67) containing the address (less *7) of the next character in the input buffer which is due to be transferred to the (main) store.

(68) containing the V-store address of the peripheral (less *6)

b) Output peripherals.

(50) bit 23 = 0 if there is no carriage control character at the end of the record currently being transferred to the buffer of the peripheral.
= I if there is a carriage control character at the end of the current record which has not yet been transferred to the buffer.
bits 5 - 0 containing the carriage control character or zero.
(In particular, this half word contains zero if there is no record currently being transferred).

(5I) containing -0.4 or >0.

(52) containing information on the code conversion that has currently been used for output by this peripheral.

(56) containing the number of characters in the current record which have not yet been transferred to the output buffer. (In particular, this half word contains zero if there is no record currently being transferred).

(60) containing starting address (less *7) of the output buffer.

(6I) containing end address (less *7) of the output buffer.

(62) containing the next address in the output buffer (less *7) beyond that of the character which was sent out at the last interruption.

(64) containing the address in the (main) store immediately after the address of the character which was the last to be transferred to the output buffer.

(66) see above under exit conditions.

(67) containing the next address in the output buffer (less *7) beyond that of the character which was last transferred there from the (main) store.

(68) containing V-store address (less *6) of the peripheral.

R515 Start any peripheral.

Purpose To check that the peripheral is in a safe condition to be started: to send the start bit to it; and to set a marker in the private store to indicate that the peripheral is now supposed to be running.

Registers of fixed store 13

Entries at (1) or (2) with some or all of the following information:

B100 containing address of private store (less *7)
B109 containing a mask appropriate to the particular peripheral being started: if any of the corresponding bits in the peripheral control line are read as one, the peripheral will not be started.
(60) containing starting address of input/output buffer of peripheral (less *7)
(68) containing V-store address of peripheral (less *6)

Entry at (1) with B100, B109, (60), (68) as above.
Exit is to co-ordinator at (1/202) with:
B100 unaltered
(51) containing zero
(60) unaltered
(62) as (60)
(67) as (60)
(68) unaltered

Entry at (2) with B100, B109, (68) as above.
Exit is to co-ordinator at (1/202) with:
B100 unaltered.
(51) containing zero.
(68) unaltered

Working Space B100, B107 to B109

Subroutines R599, R202

R516 Set code conversion parameters.

Purpose To set up the parameters in B lines and subsidiary store required by the code conversion routines R517, R527.

Registers of fixed store 11

Entry at (1) with:

B100 containing address of private store (less *7)
B110 containing return address.
(52) bits 23-3 containing the address of a table containing the parameters of the particular code conversion required. (See specification R517)

bit 2 containing 0 or 1 depending on whether the last character which was converted previously was in the first or second shifts of the 'primary' code.

bit 1 containing 0 or 1 depending on whether the last character which was converted previously was placed in the first or second shifts of the 'secondary' code.

bit 0 containing 1 if the information is in binary, i.e. no code conversion is required.

Exit to address in B110 with

B100 unaltered
B108 Bits 23-3 as (52)
Bits 2 -0 zero
B109 as (52)
(90) containing the address of the table in which the first character is to be looked up.
(91) bits 23-3 containing one of four numbers depending on whether the first character is expected to be in the first or second shifts of the 'primary' code and in the first or second shifts of the 'secondary' code.
bits 2, 1 as (52)

Working space B100, B108 to B110

Subroutines R517, R599.

Note (90) and (91) are the addresses of two half words within the supervisor extracode working space.

R517 Character code conversion.

Purpose To convert characters from any two shift code to any other two shift code. The routine is entered every time a character is to be converted. The exit is to one of five specified addresses, depending on whether there has been a one-to-one conversion of the character; or if a shift character is required to precede it in the 'secondary' code; or if the character in the 'primary' code was a shift character which does not occur in the 'secondary' code; etc.

Registers of fixed store 72

Parameters and tables All entries require:

B108 Containing the address of a list of parameters for the particular code conversion required. The parameters, in successive half words are:

0.0 & 0.4: the address of the two tables in which to look up each character. One table for each of the two shifts of the 'primary' code.

1.0 & 1.4: the two shift characters in the 'secondary' code.

2.0 : the exit address when there has been a one-to-one conversion.

2.4 : the exit address when one of the shift characters in the 'secondary' code has to be inserted.

3.0 : the exit address when the character to be converted is found to be a significant shift character in the 'primary' code.

3.4 : the exit address when the character to be converted is found to be a redundant shift character in the 'primary' code.

4.0 : the exit address when the character to be converted is found to belong to a chosen subset in the 'primary' code. e.g. the end of line characters.

(90) containing a copy of the address of one of the two tables in which to look up the first character.

(91) containing one of four addresses depending on which 'primary' code shift and which 'secondary' code shift the first character is expected to be in.

The character tables (refered to at the address in B108 +0.0 & +0.4)must contain, in each half word:

bits 23-21 : 0 if the corresponding character is to be included in the subset for which a special exit is made.
1 if the corresponding character appears in both shifts of the 'secondary' code.
2 if the character in the 'primary' code is a shift character.
3 if the character in the 'primary' code is the other shift character.
4 if the corresponding character belongs to the first shift in the 'secondary' code.
5 if the corresponding character belongs to the second shift in the 'secondary' code.
6 if the character is the first shift character in the 'secondary' code.
7 if the character is the second shift character in the 'secondary' code.

bits 20-0 : The character in the 'secondary' code.

<u>Entries</u> at (1) or (2) or (3)

Enter at (1) with B108, (90), (91) as above
B103 containing the address (less *7) of the half word containing the 'primary' code character. The character is to occupy bits 23-2 of the specified half word.

Enter at (2) with B108, (90), (91) as above
B109 bits 23-2 containing the 'primary' code character.

Enter at (3) with B108, (91) as above.
B109 bits 20-0 containing the 'secondary' code character
bits 23-21 '' information as in any half word of the character tables.

<u>Exit</u> to one of the addresses specified in the parameter list in B108 with:
B108 unaltered.
B109 containing the 'secondary' code character in bits 20-0
(90), (91) updated ready for next character.

<u>Working Space</u> B100, B103, B108 to B110

<u>Subroutine</u> R599.

<u>Note</u> (90) and (91) are the addresses of two half words within the supervisor extracode working space.

R519 Insert separator.

Purpose To be entered after R517 as an alternative to R518 when conversion has been into the internal code and a separator is required to be inserted.

Registers of fixed store 22

Entries at (1) or (2)

Enter at (1) if this is the true end of the record, with:
B100 containing the address of the private store (less *7)
B101 containing the address (less *7) of the next vacant character space in the block currently being filled by the peripheral.

B110 containing return address.
(52) bits 23-3, 0 as at entry to R516
bits 2-1 immaterial.
(91) bits 23-3, 1, 0 immaterial.
bit 2 as at exit from R517
(56) containing address (less *7) of that half word in the block currently being filled by the peripheral which has been reserved for the separator.

Enter at (2) if the record is to be broken off, but continued elsewhere, with information as above.

Exit to address in B110 with:
B100 unaltered.
B101 rounded up to the next half word address (and that half word is cleared)
(52) bit 23-3, 0 unaltered.
bit 2 as (91)
bit 1 zero (i.e. effective change to inner set)
(56) contains zero if entry was at (1)
0.1 if entry was at (2)
B108 bits 23-3 as (52)
bits 2-0 zero.
B109 as (52)
(90) containing the address of the (inner set) table in which the next character should be looked up.
(91) bits 23-3 containing one of two numbers depending on whether the next character is expected to be in the first or second shifts of the 'secondary' code.
bits 2, 1, as (52)

Working space B100, B101, B107 to B110

Subroutines R516, R517, R518, R599

Notes

1. (90)(91) are the addresses of two half words in subsidiary store in the supervisor extracode working space.
2. Exit is via R516, so that parameters required at any subsequent entry to R517 or R527 are already set up.

R520 Set reserved block label.

Purpose To bring the block currently being filled by an input peripheral or being printed from by an output peripheral, into core store, and to change its block number to *3667.

Registers of fixed store 15

Entries This routine is effectively two routines and must be entered twice, first at (1), and again at (2). For each entry, the following information is required.

B100 containing address of private store (less *7)
B110 containing return address.
(64) containing any address in the block which is to be brought to cores.

Exit from each part is to address in B110. At the second exit:
B100 is unaltered.
B101 bits 23-12 contain *4667(=*3667 less *7)(see Note 1)
bits 11-0 as bits 11-0 of (64)
B102 bits 23-12 zero.
bits 11-3 page number occupied by block. (see Note 1)
(64) unaltered.

Notes

1. The block refered to is expected to be in main store. The routine may be entered, however, even if the block is in subsidiary or fixed store. In this case the block is left where it is, and on exit.

B100 is unaltered.
B101 as (64) except (less *7).
B102 contains *4
(64) unaltered.

2. If a drum transfer is required to obtain the specified block a return will be made not to the address in B110 but to the 'current S.E.R. entry address' held in (5/201). In this case, the exit conditions are:
B100 unaltered.
(64) unaltered.
Before entering this routine the 'current S.E.R. entry address' should therefore be set to some suitable restart point where this routine R520 will be entered at the beginning again.

Working Space All supervisor B lines.

Subroutines R312, R318, R599.

R521 Pick up record separator.

Purpose

To pick up the next half word of information being output:
to interpret it as a separator:
to preserve in the private store the character count and the carriage control character: and
to change the current internal code shift to 'inner set'.

Registers of fixed store 36

Entry

at (1) with:

B100	containing address of private store (less *7)
B101	containing the next character address (less *7) in the (main) store beyond the address of the character which was last transferred to the output buffer.
B110	containing return address.

Exits

a)

Usually to address in B110 with:

B100	unaltered.		
B101	advanced to the address (less *7) of the first character in the new record.		
B108	containing the separator.		
(50)	bit 23	= 1	
	bits 22-6	zero	
	bits 5-0	containing carriage control character.	
(56)	containing count of character in the record excluding the carriage control character. (bits 0 upwards).		
(52)	bits 23-3, 1	unaltered.	
	bit 2	= 0	
	bit 0	= 0	if internal code record.
		= 1	if binary record.

b)

If the record is not complete and there is no carriage control character, the exit is to address in B110 with:

B100	unaltered.		
B101	advanced as above.		
B108	containing the separator.		
(50)	containing zero.		
(56)	containing count of characters in part record.		
(52)	bits 23-1		unaltered.
	bit 0	= 0	if internal code.
		= 1	if binary.

c) If the separator is found to be zero, the exit is to address in B110 with:

B100 unaltered.
B101 rounded up to the address of the separator (less *7)
B108 containing zero.
(50) containing zero.
(56) containing zero.
(52) unaltered.

d) A check is made by this routine that the next separator is in the same 512 word block as the present separator. If this check fails, an exit is made to monitor (here labelled (99/521) with:

B100 unaltered.
B101 advanced to first character address in present record (less *7)
B108 containing the faulty separator.
(50) unaltered.
(56) containing the erroneous character count.
(52) unaltered.

Working Space B100, B101, B106 to B110.

Subroutines R512, R599.

Note If the separator is found to contain bit 14 = 1 this is interpreted as if the separator contained bits 22, 21 = 1, all other bits zero, indicating a record with zero character count whatever the other bits actually are.

R522 Find peripheral buffer in part page.

Purpose To pick up the address of the next available space in the peripheral buffer and to find the length remaining.

Registers of fixed store 6

Entries at (1) or (2)

Enter at (1) when the length remaining is to be reckoned from the next available space in the buffer to the end of the buffer, with:

B100 containing address of private store (less *7)
B110 containing return address.
(61) containing end address of buffer (less *7)
(67) containing address in the buffer (less *7) of the space available to receive the next character to be transferred from the (main) store.

Enter at (2) when the length remaining is to be reckoned from the next character in the buffer to the last character at present lying in the buffer, with:

B100 containing address of private store (less *7)
B10 containing return address.
(62) containing end address (less *7) of the information at present lying in the input buffer.
(67) containing the address (less *7) of the next character in the input buffer which is due to be transferred to the (main) store.

Exit to address in B110 with:

B100 unaltered.
B103 as (67)
B104 containing number of half words left in buffer.
(61), (62), (67) unaltered.

Working space B100, B103, B104, B110.

Subroutines R599.

2.9.63

R523 Remove reserved block label.

Purpose To alter the P.A.R. of the page holding the block currently being filled by an input peripheral or being printed from by an output peripheral: and
to preserve in the private store the address of the next character space or character in that block and the address of the next character or character space in the buffer of the peripheral.

Registers of fixed store 9

Enter at (1) with:

B100 containing address of private store (less *7)
B101 containing the next character address in the reserved block (bits 23-12 immaterial)
B102 containing:
either bits 23-12 zero,
bits 11-3 page number occupied by block.
or if the block is not in the main store at all, but in subsidary or fixed store:
bit 23 = 1
bits 22 - 0 zero
B103 containing the next character address in the buffer (less *7)
B110 containing return address
(64) bits 23-12 containing original number with which the block was labelled.
bits 11-0 immaterial.

Exit to address in B110 with:
B100 unaltered.
(64) bits 23-12 unaltered.
bits 11-1 as B101 at entry
(67) as B103 at entry
The P.A.R. is set to *4

Working Space B100 to B103, B107 to B110.

Subroutines R312, R599.

R527 Carriage control code conversion.

Purpose To convert an internal code carriage control character into corresponding characters in any two shift output code. The routine is entered each time a carriage control character is to be converted. Exit is to one of four specified addresses depending on whether there has been a simple conversion of the character; or if a shift character is required to precede it in the output code etc.

Registers of fixed store 28

Parameters and tables Each entry requires:

B108 containing the address of a list of parameters of the particular code conversion required. The parameters in successive half words are:

0.0 & 0.4 : immaterial.
1.0 & 1.4 : the two shift characters in the output code.
2.0 to 4.0: immaterial.
5.0 : the address of a table in which to look up each carriage control character.
5.4 : the exit address when this has been a simple conversion to the output code.
6.0 : the exit address when a shift character is required in the output code.
6.4 : the exit address when the carriage control character to be converted is found to belong to a chosen subset of characters.
7.0 : the exit address when there is found to be no carriage control character.

The table referred to at the address in B108 + 5.0 must contain in each half word:

Bits 23 & 22: 00 if the corresponding character is to be included in the subset for which a special exit is made.
01 if the corresponding character belongs to the 'first' shift in the output code.
10 if the corresponding character belongs to the 'second' shift in the output code.
11 if the corresponding character can appear in either shift of the output code.

bits 17-12 containing the internal code carriage control character which represents the balance of carriage control information after the present output character has been sent off.(e.g. where the single internal code character has to be represented by a whole sequence of output characters)

bits 11-0 containing the character in the output code.

R527 continued...

Enter at (1) with:

B100 containing address of private store (less *7)
B108 as above.
(50) bit 23=0 if there is no carriage control character to be transferred to the output buffer.
=1 if there is a carriage control character.
bits 5-0 containing the carriage control character (if any)
(52) bit 1=0 if output is currently in 'first' shift of output code.
=1 if output is currently in 'second' shift of output code.
bits 23-2, 0 immaterial.

Exit to one of the addresses specified in the parameter list in B108 with:

B100, B108 unaltered.
B109 containing the output code character in bits 11-0 (and including other bits 23-12 as table)
(50)&(52) updated. If exit is at address where there has been no carriage control character, (50) contains zero.

Working space B100, B107 to B110

Subroutine R599.

R530 Card reader fault test.

Purpose This is a subroutine of the Peripheral One Second R508. It examines the control bits of the card reader to find if the equipment needs any attention and returns to the Peripheral One Second with the appropriate information.

Registers of fixed Store 23

Enter at (1) with:

B102 containing V-store address of card reader (less *6)
B104 containing zero if and only if the card reader has been freed (by use of R504)

Exits a) if the card reader is already started:
to (9/508)

b) if the reader is stopped and free:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine R724.

c) if the reader is not free, but is stopped and disabled:
to (7/508) with:
B108 containing 1.0
B109 containing entry address to Card Reader P.E.R. R533.
The card reader disengaged.

d) if the reader is not free, but is stopped and overdue:
to (7/508) with:
B108 containing 2.0
B109 containing entry address to Card Reader P.E.R. R533.
The card reader disengaged and overdue reset.

e) if the reader is not free, but is stopped with card levels bit set:
to (7/508) with:
B108 containing 4.4
B109 containing entry address to card reader P.E.R. R533.
The card reader disengaged.

f) if the reader is not free, but is stopped with no fault indications:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine, R724.

Working space B102, B104, B106 to B109

Subroutines R508, R533, R724.

2.9.63

R531 Card reader column interruption.

Purpose To read a card column and store it in the card reader's buffer: to read from the checking station, and compare with the previous column read: to record the number of check failures: and to look for the punching 7 and 8 in the first column, indicating the end of a pack of cards.

Registers of fixed store 40

Enter in interrupt control at (1) when a one appears in bit P0 of the central computer V-store with:

(50) bits 17-12 containing the count of check failures detected previously on this card.
bits 7-2 containing the count of columns previously read on this card.
bit 0 containing 1 if the first column of this card had the punching 7 and 8.
(62) containing the address (less *7) of the next available space in the card reader's buffer.

Exits in interrupt control.

a) normally to R500 with:
(50) updated
(62) updated
The L.A.M. extinguished.

b) to R201 (via R532) if more than 81 column interruptions have occured since the last End of Card interruption, with:
B111 containing address (less *7) of card reader's private store.
B112 containing entry address to card reader P.E.R. (R533)
(50) updated
(51) containing 5.0 indicating reader check failure.
(62) updated
Card reader stopped and disengaged.
The L.A.M. extinguished.

c) to monitor (here labelled (99)) if the most significant L.A.M. in the peripheral V-store type 15 line 8, does not refer to a card reader, with:
B123 containing the number of the most significant L.A.M.

d) to R500 if the peripheral V-store type 15 line 8 is found to contain no L.A.M.

Working Space B111 to B118, B123

Subroutines R201, R500, R532, R533, R599.

R532 Card reader end-of-card interruption.

Purpose To check that the number of columns read on the card was correct and that there have been no check failures: to find whether there is sufficient buffer to accept another card: and if not, to call the Card Reader P.E.R. to the S.E.R. queue.

Registers of fixed store 29

Enter in interrupt control at (1) when a L.A.M. appears in bit 31 of peripheral V-store type 15 line 15, with:

(50) bits 17-12 containing the count of check failures detected on the card.
bits 7-2 containing the count of columns read on this card.
bit 0 containing 1 if the first column of this card had the punching 7 and 8.
(61) containing the end address (less *7) of the card reader's buffer.
(62) containing the address in the buffer (less *7) of the space available for the next interruption.

Exits in interrupt control.

a) to R500 if there have been no faults on this card and there is buffer space for the next one, with:
(50) containing zero.
The L.A.M. extinguished.
The card directed to the main stacker.

b) to R201 if there have been no faults on this card, but there is insufficient buffer space for another one, with:
B111 containing address of private store (less *7)
B112 containing entry address to Card Reader P.E.R., R533.
(50) containing zero.
(51) containing -0.4
The L.A.M. Extinguished.
The card reader stopped.
The card directed to the main stacker.

c) to R201 if there have been no faults on this card, but bit 0 of (50) was set =1 (first column punched 7 and 8) with:
B111 containing address of private store (less *7)
B112 containing entry address to Card Reader P.E.R., R533.
(50) containing zero.
(51) containing 5.4
The L.A.M. Extinguished.
The card reader stopped and disengaged.
The card directed to the main stacker.

d) to R201 if there has been a check fail on the card or if the count of columns is not 81, with:
B111 containing address of private store (less *7)
B112 containing entry address to Card Reader P.E.R., R533.
(50) unaltered.
(51) containing 7.0
The L.A.M. Extinguished.
The card reader stopped and disengaged.

Note When an exit is made without fault (exits (a), (b) or (c)), the half word of buffer corresponding to the 81st. card column is set to contain:
bit 11 = 1
all other bits = 0

Working space B111 to B115, R123

Subroutines R201, R500, R533, R599.

R533 Card reader P.E.R.

Purpose: To pick up information deposited in the card reader buffer by the card reader interruption routines; to perform code conversion if requested: to pack the information in the main store block: and to re-start the card reader if necessary.

Registers of fixed store: 181

Enter at (1) with:

B100 containing address of private store (less *7)

(50) bits 7-2 containing either the count of columns read on the last card if the last card was misread in any way; or zero if the last card was read without mishap.

(51) containing either -0.4 if the card reader has been stopped only because the card reader buffer is full; or >0 if it has been stopped for any other reason.

(52) bits 23-3 containing the address of a table required by the code conversion routine (see below)

bit 2 containing one if this particular card is being translated as binary (because its first column was a non-standard punching) although current requirement is to convert to internal code.

bit 1 containing either 0 if the last character at the previous entry to this routine was translated into the inner set; or 1 if it was translated into the outer set.

bit 0 containing either 0 if code conversion is required, or 1 if no code conversion is required (in which case bits 23-1 are ignored).

(53) bits 23-2 containing the count of punching errors detected previously.

bit 0 containing either 0 if translation is required to continue past any card column which is found to contain a mispunching; 1 if the current requirement is to exit when a mispunching is detected.

(56) containing either address (less*7) reserved for separator; or zero if a new record has not been started yet.

(60) containing starting address of card reader buffer (less*7)

(61) containing end address of card reader buffer (less *7)

(62) containing end address (less *7) of information at present lying in the card reader buffer.

(64) containing the next available character address in the (main)store.

(65) bits 23-12 containing either *4667 (=*3667 less*7) if (64) refers to main store block: or same as bits 23-12 of (64) (less*7) if (64) refers to subsidary store.

bits 11-0 containing end address of space in the current (main store) block.

(66) containing address for return to the Input Master Routine when reading is complete.
(67) containing address (less*7) of the next character in the card reader buffer which is due to be transferred to the (main) store.
(68) containing the V-store address of the card reader (less*6)

Note — A restart address in case this routine is interrupted by non-equivalence, must be set in (5/201) before entering this routine.

Code conversion

On entry, if code conversion is required, (52) bits 23-3 contains the address of a table required by the code conversion routine. The table and conversion routine incorporated here translates from Harwell card code to internal code, with the rules that if the first column of a card is not an allowed punching in the card code, the whole card is stored as binary, and if the first column has punchings 7 and 8 this is taken to be the end of the pack of cards. If any other code conversion is required an appropriate table and routine should be provided, and (52) bits 23-3 should contain address of the table.

Exits a) if all the information in the buffer has been transferred to the (main) store and if (51) contains >0 : to the address given in (66) with:

B100	containing address of private store (less *7)	
(50)	containing zero.	
(51)	unaltered.	
(52)	bits 23-3 0	unaltered.
	bits 2, 1	updated.
(53)	bits 23-2	updated.
	bit 0	unaltered.
(56)	containing zero.	
(60)	unaltered.	
(61)	unaltered.	
(62)	unaltered.	
(64)	updated.	
(65)	unaltered.	
(66)	as (51)	
(67)	as (62)	
(68)	unaltered.	

b) If all the information in the buffer has not been transferred to the (main) store because the store area allocated has been filled: to the address given in (66) with B100 and private store as at exit (a) except:
(66) containing zero.
(67) containing the address (less *7) of the next character in the card reader buffer which is due to be transferred to the (main) store.

c) if all the information in the buffer has not been transferred to the (main) store because a punching error has been detected and (53) is odd: to the address given in (66) with B100 and private store as at exit (a) except:

(51) containing 4.0
(62) containing the address in the card reader buffer (less *7) after the character which was mispunched.
(66) as (51)
(67) as (62)

d) If all the information in the buffer has been transferred to the main store and (51) contains -0.4: to Program Scan, R202, with:

B100 containing address of private store (less *7)
(50) containing zero.
(51) containing zero.
(52) bits 23-3, 0 unaltered.
bit 2, 1 updated.
(53) bits 23-2 updated.
bit 0 unaltered.
(56) updated.
(60) unaltered.
(61) unaltered.
(62) as (60)
(64) updated.
(65) unaltered.
(66) unaltered.
(67) as (60)
(68) unaltered.

The card reader started, provided it is not overdue, nor with card levels wrong, nor disabled, nor disengaged.

Working space B100 to B110
Three half words of supervisor extracode working space, here labelled (90), (91) and (92)

Subroutines R201, R202, R312, R318, R511, R512, R513, R514, R515, R516, R517, R518, R519, R520, R522, R523, R599.

R540 TR7 fault test.

Purpose: This is a subroutine of the Peripheral One Second R508. It examines the control bits of the tape reader to find if the equipment needs any attention, and returns to the Peripheral One Second with the appropriate information.

Registers of fixed store 28

Enter at (1) with:

B102 containing V-store address of tape reader (less *6)

B103 containing a marker showing the state of use of the reader.

B104 containing zero if and only if the tape reader has been freed (by use of R504).

Exits a) If the reader is already started and B103 contains zero:
to (8/508) with:
B108 containing -1.0

b) if the reader is already started and B103 contains -1.0 (indicating that the reader has given no L.A.M. during previous second):
to (7/508) with:
B108 containing 1.4
B109 containing entry address to TR5 P.E.R., R568.
The reader stopped and disengaged.

c) if the reader is stopped and free:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine R724.

d) if the reader is not free, but is stopped and disabled:
to (7/508) with:
B108 containing 1.0
B109 containing entry address to TR5 P.E.R., R568.
The tape reader disengaged.

e) if the reader is not free, but is stopped and overdue:
to (7/508) with:
B108 containing 2.0
B109 containing entry address to TR5 P.E.R., R568.
The tape reader disengaged and overdue reset.

f) if the reader is not free but is stopped with tape warning bit set:
to (7/508) with:
B108 containing 4.4
B109 containing entry address to TR5 P.E.R, R568.
The tape reader disengaged.

g) if the reader is not free, but is stopped with no fault indications:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine R724.

Working space B102 to B104, B108, B109.

Subroutines R508, R568, R724.

2.9.63

R541 TR7 interruption.

Purpose To read a character from the tape reader and store it in the tape reader's buffer: to check the parity of the tape character: to record the number of failures: and look for three consecutive asterisks indicating the end of the tape.

Registers of fixed store 35

Enter in interrupt control at (1) when a one appears in bit 28 of the peripheral V-store type 15 line 15, with:

(50) bits 17-0 containing zero if the three previous characters were all asterisks.

(53) bits 23-2 containing the count of parity errors detected previously on the tape.

bit 0 containing either 0 if reading is required to continue past a character with even parity, or 1 if the current requirement is to stop the reader when such a character is detected.

(61) containing end address (less *7) of the tape reader's buffer.

(62) containing the address (less *7) of the next available space in the tape reader's buffer.

Exits in interrupt control.

a) normally to R500 with:
(50) updated.
(51) containing zero.
(61) unaltered.
(62) updated.
The L.A.M. extinguished.

b) to R201 if the end of the tape reader's buffer has been reached with:
B111 containing address of private store (less *7)
B112 containing entry address to TR5 P.E.R., R568.
(50) updated.
(51) containing -0.4
(53) updated.
(61) unaltered.
(62) updated (now as (61)),
The L.A.M. extinguished.
The reader stopped.

c) to R201 if a parity fault has been detected on 7-hole tape and if (53) bit 0 contains 1, with:
B111 containing address of private store (less *7)
B112 containing entry address to TR5 P.E.R., R568.
(50) updated.
(51) containing 4.0
(53) updated.
(61) unaltered.
(62) updated.
The L.A.M. extinguished.
The reader stopped and disengaged.

d) to R201 if the three previous characters were asterisks, i.e. if (50) bits 17-0 contain zero with:

B111 containing address (less *7) of private store.
B112 containing entry address TR5, P.E.R., R568.
(50) updated.
(51) containing 0.4
(53) updated.
(61) unaltered.
(62) updated.
The tape reader stopped.
The L.A.M. extinguished.

e) to R500 if the peripheral V-store type 15 line 4 is found to contain no L.A.M.

<u>Notes</u> a) (50) is never updated and an exit is not made at (d) if the character now being read is one of the shift characters UC LC FS or LS

b) While testing for these shift characters or for asterisks the tape code is assumed to be either 7 hole Flexowriter code or 5 hole Mercury/Pegasus code.

Working space B11 to B118, R123

Subroutines R201, R500, R568, R599.

R550 Anelex Fault Testing Routine.

Purpose This is a subroutine of the Peripheral One Second, R508 It examines the control bits of the Anelex printer to find out if the equipment needs any attention and returns to the Peripheral One Second with the appropriate information.

Registers of Fixed Store: 17

Enter at (1) with:

B102 containing V-store address of Anelex (less *6)

Exits

a) if the Anelex is started already:
to (9/508)

b) if the Anelex is stopped without a fault:
to (7/508) with:
B108 containing -0.4
B109 containing entry to Anelex P.E.R. R553.

c) if the Anelex is stopped and disabled:
to (7/508) with:
B108 containing 1.0
B109 containing return to Output Master Routine R514.
Printer disengaged.

d) if the Anelex is stopped and paper low:
to (7/508) with:
B108 containing 3.0
B109 containing return to Output Master Routine. R514
Printer disengaged.

Working space B102, B106 to B109.

Subroutines used R508, R514 & R553.

R551 Anelex Interruption.

Purpose To check whether it is a print or overflow interrupt; to enable the contents of the Anelex buffer to be printed; to call the P.E.R. to the S.E.R. queue if a print interrupt; to register a fault if an overflow interrupt.

Registers of fixed store 25

Entry at (1) in interrupt control when a one appears in bit 27 of peripheral V-store type 15 line 13.

Exits in interrupt control.

a) to R201 if a print L.A.M. has occured, with:
B111 containing address of private store (less *7)
B112 containing entry address to P.E.R. for Anelex, R553
(51) containing -0.4
L.A.M. extinguished.

b) to R201 if an overflow L.A.M has occured with:
B111 containing address of private store (less *7)
B112 containing return to Output Master Routine R514.
(51) containing 5.0.
The L.A.M. extinguished and the printer stopped and disengaged.

c) to monitor (here labelled (99/599)) if the most significant L.A.M. in the peripheral V-store type 14 line 3 does not refer to the Anelex, with:
B123 containing the number of the most significant L.A.M.

d) to R500 if the peripheral V-store type 14 line 3 is found to contain no L.A.M.

Working space B111 to B114, and B123.

Subroutines R599, R514, R553, R201, and R500.

R553 Anelex P.E.R.

Purpose To take characters stored in the (main store) block and transfer them to the Anelex buffer; to perform code conversion if required; to start the Anelex if an end of record character is recognised or if the Anelex buffer is full.

Registers of fixed store 200

Enter at (1) after an interrupt has been completed with:

B100 containing address of private store of Anelex (less *7)

(50) bits 23 = 0 if there is no carriage control character at the end of the record currently being transfered to the Anelex buffer.
= 1 if there is a carriage control character at the end of the record.
bits 5 - 0 contain the carriage control character or zero.

(51) containing -0.4 or >0 if the Anelex has been stopped with a fault.

(52) bits 23 - 3 containing address of code conversion table.
bit 2 = 0 if the last character at the previous entry to this routine was in inner set.
= 1 if the last character at the previous entry to this routine was in outer set.
bit 1 = 0
bit 0 = 0 if the information is in internal code.
= 1 if the information is binary and no code conversion is required.

(56) containing number of characters in record still to be printed.

(60) containing zero.

(61) containing 60.0

(62) containing zero.

(64) containing address of next character to be printed.

(66) containing return address to the Output Master Routine when printing is finished.

(67) containing zero.

(68) containing peripheral V-store address of Anelex (less*6).

at (2) after a new (main store) block of information has been received, with:

B100 containing address of private store of Anelex (less*7).

(50) containing zero.

(51) as at entry (1)

(52) as at entry (1)

(56) containing zero.

(60) containing zero.

(61) containing 60.0

(62) containing *5 at beginning of new output, zero if output is continuing.

(64) containing address of first separator.

(66) as at entry (1)

(67) containing position in Anelex buffer for next character.

(68) as at entry (1)

R561 Creed 3000 interruption.

Purpose To send the next character from the Creed 3000 punch buffer to the punch: and to stop the punch when the last character is reached.

Registers of fixed store 13

Enter in interrupt control at (1) when a one appears in bit 30 of peripheral V-store type 15 line 14, with:

(62) containing the address (less *7) of the next character in the buffer of the Creed 3000.

Exits in interrupt control.

a) normally to R500 with:
(62) updated.
The L.A.M. extinguished.

b) to R201 after a character has been found in the buffer with bit 1 containing 1, with:
B111 containing address (less *7) of private store of Creed 3000 punch.
B112 containing entry address to teletype punch P.E.R. (R573)
(51) containing -0.4
(62) updated.
Creed 3000 stopped.
The L.A.M. extinguished.

c) to R500 if the peripheral V-store type 14 line 14 is found to contain no L.A.M.

Working space B111 to B114, B123.

Subroutines R201, R500, R573, R599

R565. TR5 fault test.

Purpose This is a subroutine of the Peripheral One Second, R508. It examines the control bits of the tape reader to find if the equipment needs any attention, and returns to the Peripheral One Second with the appropriate information.

Registers of fixed store 20

Enter at (1) with:

B102 containing V-store address of tape reader (less *6)
B103 containing a marker showing the state of use of the reader.
B104 containing zero if and only if the tape reader has been freed (by use of R504).

Exits a) If the reader is already started and B103 contains zero: to (8/508) with:
B108 containing -1.0

b) if the reader is already started and B103 contains -1.0 (indicating that the reader has given no L.A.M's for the previous second):
to (7/508) with:
B108 containing 1.4
B109 containing entry address to TR5 P.E.R., R568.
The reader stopped and disengaged.

c) if the reader is stopped and free:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine R724.

d) if the reader is not free, but is stopped and overdue:
to (7/508) with:
B108 containing 2.0
B109 containing entry address to TR5 P,E.R, R568.
The tape reader disengaged and overdue reset.

e) if the reader is not free, but is stopped with no fault indications:
to (7/508) with:
B108 containing -0.4
B109 containing entry address to Input Master Routine R724.

Working space B102 to B104, B108, B109.

Subroutines R508, R568, R724.

R566 TR5 Interruption. (Tape readers 0-7).

Purpose to read a character from the tape reader and store it in the tape reader's buffer: to check the parity of the tape character: to record the number of parity failures: and look for three consecutive asterisks indicating the end of the tape.

Registers of fixed store 35

Enter in interrupt control at (1) when a one appears in bit 27 of the peripheral V-store type 15 line 14 with:

(50) bits 17-0 containing zero if the three previous characters were all asterisks.
(53) bits 23-2 containing the count of parity errors detected previously on the tape.
bit 0 = 0 if reading is required to continue past a character with even parity.
= 1 if current requirement is to stop the reader when such a character is detected.
(61) containing end address (less *7) of the tape reader's buffer.
(62) containing the address (less *7) of the next available space in the tape reader's buffer.

Exits in interrupt control.

a) normally to R500 with:
(50) updated.
(51) containing zero.
(61) unaltered.
(62) updated.
The L.AM. extinguished.

b) to R201 if the end of the tape reader buffer has been reached with:
B111 containing address of private store (less *7)
B112 containing entry address to TR5 P.E.R., R568.
(50) updated.
(51) containing -0.4
(53) updated.
(61) unaltered.
(62) updated (now as (61))
The L.A.M. extinguished.
The reader stopped.

c) to R201 if parity has been detected and if (53) bit 0 contains 1 with:
B111 containing address of private store (less *7)
B112 containing entry address to TR5 P.E.R., R568.
(50) updated.
(51) containing 4.0
(53) updated.
(61) unaltered.
(62) updated.
The L.A.M. extinguished.
The reader stopped and disengaged.

d) to R201 if the three previous characters were asterisks, i.e. (50) bits 17-0 contain zero, with:

B111 containing address (less *7) of private store.
B112 containing entry address to TR5 P.E.R., R568.
(50) updated.
(51) containing 0.4
(53) updated.
(61) unaltered.
(62) updated.
Tape reader stopped.
The L.A.M. extinguished.

e) to R500 if the peripheral V-store type 14 line 11 is found to contain no L.A.M.

Notes a) (50) is never updated and an exit is not made at (4) if the character now being read is one of the shift characters. UC, LC, FS, or LS.

b) While testing for these shift characters or for asterisks the tape code is assumed to be either 7-hole Flexowriter code of 5-hole Mercury/Pegasus code.

Working space B111 to B118, B123.

Subroutines R201 , R500, R568, R599.

R568 TR5 P.E.R.

Purpose to pick up the information deposited in the tape reader buffer by the tape reader interruption routine:to perform code conversion if requested: to pack the information in a main store block and to restart the tape reader if necessary.

Registers of fixed store: 256

Enter at (1) with:

B100 containing address of private store (less *7)

(50) containing a number depending on the four characters last read by the TR5 Interruption R566.

(51) containing either -0.4 if the tape reader has been stopped only because the tape reader buffer is full or >0 if it has been stopped for any other reason.

(52) bits 23-3 containing address of table for required code conversion (see below)

bit 2 containing either 0 if the last character at the previous entry to this routine was in upper case or letter shift; or 1 if it was in lower case or figure shift.

bit 1 containing either 0 if the last character at the previous entry to this routine was translated into the inner set; or 1 if it was translated into the outer set.

bit 0 containing either 0 if code conversion is required, or 1 if no code conversion is required (in which case bits 23-1 are ignored).

(53) bits 23-2 containing the count of tape parity errors detected previously.

bit 0 containing either 0 if translation is required to continue past any parity error; or 1 if current requirement is to exit when a parity error is detected.

(56) containing either address (less *7) reserved for separator; or zero if a new record has not been started yet.

(60) containing starting address of tape reader buffer (less *7)

(61) containing end address of the tape reader buffer (less *7)

(62) containing end address (less *7) of information at present lying in the tape reader buffer.

(64) containing the next available character address in the (main) store.

(65) bits 23-12 containing either *4667 (=*3667 less *7) if (64) refers to main store block: or same as bits 23-12 of (64) (Less *7) if (64) refers to subsidiary store.

bits 11-0 containing end address of space in the current (main store) block.

(66) containing address for return to the Input Master Routine when reading is complete.
(67) containing address (less *7) of the next character in the tape reader buffer which is due to be transferred to the (main) store.
(68) containing the V-store address of the tape reader (less *7)

Note   A restart address in case this routine is interrupted by non-equivalence, must be set in (5/201) before entering this routine.

Code Conversion   On entry, if code conversion is required, (52) bits 23-3 contains address of a table required for the code conversion routine. The table incorporated here translates form either 7 hole Flexowriter code or 5 hole Mercury/Pegasus code to internal code. If any other code conversion is required an appropriate table should be provided, and (52) bits 23-3 should contain its address.

Exits   a)   if all the information in the buffer has been transferred to the (main) store and if (51) contains >0 : to the address given in (66) with:

B100 containing address of private store (less *7)
(50) unaltered.
(51) unaltered.
(52) bits 23-3, 0 unaltered.
     bits 2, 1, updated.
(53) bits 23-2 updated.
     bit 0 unaltered.
(56) containing zero.
(60) unaltered.
(61) unaltered.
(62) unaltered.
(64) updated.
(65) unaltered.
(66) as (51)
(67) as (62)
(68) unaltered.

b)   if all the information in the buffer has not been transferred to the (main) store because the store area allocated has been filled: to the address given in (66) with B100 and private store as at exit (a) except:

(66) containing zero.
(67) containing the address (less *7) of the next character in the tape reader buffer which is due to be transferred to the (main) store.

c) if all the information in the buffer has been transferred to the main store and (51) contains -0.4 :
to Program Scan, R202, with:

B100	containing address of private storo (less *7)	
(50)	unaltered.	
(51)	containing zero.	
(52)	bits 23-3, 0 unaltered.	
	bits 2-1	updated.
(53)	bits 23-2	updated.
	bit 0	unaltered.
(56)	updated.	
(60)	unaltered.	
(61)	unaltered.	
(62)	as (60)	
(64)	updated.	
(65)	unaltered.	
(66)	unaltered.	
(67)	as (60)	
(68)	unaltered.	

The tape reader started, provided it is not overdue or disengaged.

Working space B100 to B110
Two half words of supervisor extracode working space, here labelled (90) and (91)

Subroutines R201 R202 R312 R318 R511 R512 R513 R514 R515
R516 R517 R518 R519 R520 R522 R523 R599.

R570 Teletype fault test.

Purpose This is a subroutine of the Peripheral One Second R508. It examines the control bits of the teletype punch to find if the equipment needs any attention, and returns to the Peripheral One Second with the appropriate information.

Registers of fixed store: 17

Enter at (1) with:

B102 containing V-store address of teletype punch (less *6)

Exits a) if the teletype punch is already started:
to (9/508)

b) if the teletype punch is stopped and disabled:
to (7/508) with:
B108 containing 1.0
B109 containing entry address to R514
The teletype punch disengaged.

c) if the punch is stopped with the paper warning bit set:
to (7/508) with:
B108 containing 3.0
B109 containing entry adddress to R514
The teletype punch disengaged.

d) if the punch is stopped with no fault indication:
to (7/508) with:
B108 containing -0.4
B109 containing entry to Teletype Punch P.E.R. R573 at label (43)

Working space B102, B106 to B109.

Subroutines R5088, R514, R515.

R571 Teletype punch interruption, (Teletypes 0-7)

Purpose To send the next character from the teletype punch buffer to the punch: and to stop the punch when the last character is reached.

Registers of fixed store 13

Enter in interrupt control at (1) when a one appears in bit 24 of peripheral V-store, type 15, line 14, with:

(62) containing the address (less *7) of the next character in the teletype punch buffer.

Exits in interrupt control.

a) normally to R500 with:
(62) updated.
The L.A.M. extinguished.

b) to R201 after a character has been found in the buffer with bit 1 containing 1 with:

B111 containing address (less *7) of teletype punch's private store.
B112 containing entry address to teletype punch P.E.R. (R573)
(51) containing -0.4
(62) updated.
Teletype punch stopped.
The L.A.M. extinguished.

c) to R500 if the peripheral V-store type 14 line 8 is found to contain no L.A.M.

Working space B111 to B114, B123.

Subroutines R201, R500, R573, R599.

R573 Teletype punch P.E.R.

Purpose To pick up information from (main) store: to perform code conversion if requested: to store the information in the output buffer of the teletype punch: and to start the punch.

Registers of fixed store 304

Enter at (1) or (2) with:

B100 containing address of private store (less *7)

(50) bit 23 = 0 if there is no carriage control character at the end of the record currently being transferred to the output buffer of the teletype punch.
= 1 if there is a carriage control character at the end of the current record which has not yet been transferred to the output buffer of the punch.
bits 5-0 containing the carriage control character or zero.
(In particular, this half word contains zero if there is no record currently being transferred).

(51) containing either -0.4 if the teletype punch is stopped waiting for more information; or >0 if it has been stopped with any fault.

(52) bits 23-3 containing address of table required for code conversion routine (see below)
bit 2 containing either 0 if the last character at the previous entry to this routine was in the inner set; or 1 if it was in the outer set.
bit 1 containing either 0 if the last character at the previous entry to this routine was translated into the upper case; or 1 if it was translated into the lower case.
bit 0 containing either 0 if code conversion is required or 1 if no code conversion is required (in which case bits 23-1 are ignored).

(56) containing the number of character in the current record which have not yet been transferred to the output buffer of the teletype punch.
(In particular, this half word contains zero if there is no record currently being transferred).

(60) containing starting address of teletype punch buffer (less *7)

(61) containing end address of teletype punch buffer (less *7)

(62) containing the address (less *7) of the character in the output buffer which will be sent out at the next teletype punch interruption.

(64) containing address in the (main) store of the next character to be transferred to the buffer of the teletype punch.

(66) containing address for return to the output master routine when punching is complete.

(67) containing the address in the buffer of the teletype punch (less *7) of the space available for the next character to be transferred from the (main) store.

(68) containing the V-store address of the teletype punch (less *6)

Note

A restart address in case this routine is interrupted by non-equivalence, must be set in (5/201) before entering this routine.

Code Conversion

On entry, if code conversion is required, (52) bits 23-3 contains address of a table required by the code conversion routine. The tables incorporated here translate:

a) from internal code to Flexowriter code, or

b) from internal code to Mercury/Pegasus code.

If any other code conversion if required an appropriate table should be provided, and (52) bits 23-3 should contain its address.

Exits

a) if the buffer is empty (i.e. (67) as (60)) and the next separator in the (main) store is zero : to the address given in (66) with:

B100 containing address of private store (less *7)
(50) containing zero.
(51) unaltered.
(52) bits 23-2, 0 unaltered.
bit 1 updated.
(56) containing zero.
(60) unaltered.
(61) unaltered.
(62) unaltered, as (60)
(64) updated.
(66) containing zero.
(67) unaltered, as (60)
(68) unaltered.

b) If all the information in the (main) store up to zero separator, has been transferred to buffer: to program scan, R202, with:

B100 containing address of private store (less *7)
(50) updated.
(51) containing zero.
(52) bits 23-2, 0 unaltered.
bit 1 updated.
(56) updated.
(60) unaltered.
(61) unaltered.
(62) as (60)
(64) updated.
(66) unaltered.
(67) as (60)
(68) unaltered.

The teletype punch started, provided it is not with paper warning bit set, nor disabled, nor disengaged.

c) After entry at (1) and if (50) and (56) both contain zero at entry, the exit is immediate, to the address given in (66) with:

B100 containing address of private store (less *7)
(66) containing zero.
Rest of private store unaltered.

<u>Note</u> The last character sent to the buffer of the teletype punch has bit 1 = 1, corresponding to the Stop bit of the punch and is used by Teletype interruption for detecting the end of information in the buffer.

<u>Working space</u> B100 to B110.

<u>Subroutines</u> R201 R202 R312 R318 R511 R512 R514 R515 R516 R517 R518 R520 R521 R522 R523 R527 R599.

R575 Card Punch Fault Testing.

Purpose This is a subroutine of the Peripheral One Second. R508. It examines the control bits of the Card Punch and returns to the Peripheral One Second with the appropriate information. An alternative entry can be made if a check fail or extra L.A.M. was detected during the interruption: to either repeat the faulty card or to return to the Output Master Routine with the appropriate information.

Registers of fixed store 43

Entry at (I) from the Peripheral One Second routine with:

BI02 containing the V-store address of the card punch (less *6)

Entry at (I5) After a check fail or extra L.A.M. has been detected in the interrupt routines with:

BI00 containing the private store address of the card punch (less*7)

(60) containing the beginning address of the card punch buffer (less*7)

(62) containing the present position along the card punch buffer (less*7)

(65) containing the number of times this card has check failed.

(68) containing the V-store address of the card punch (less*6)

Exits

a) if the Card Punch is already started:
to (9/508)

b) if the punch is stopped with no faults:
to (7/508) with:
BI09 containing (47/579) return address to Card Punch P.E.R.

c) if the punch is stopped and disabled:
to (7/508) with:
BI08 containing I.0
BI09 containing entry address to R5I4.
(62) reset to the address (less *7) in the card punch buffer of the first row, of the card, which was being punched when the fault occured.
Card punch stopped and disengaged.

d) if the card punch is stopped with overdue set:
to (7/508) with:
BI08 containing 2.0
BI09 containing entry address to R5I4
(62) as at exit (c)
Card punch stopped and disengaged and overdue reset.

2.9.63

e) if the card punch is stopped and out of cards:
to (7/508) with:
BI08 containing 3.4
BI09 containing entry to R5I4
(62) set to the address (less *7) in the card punch buffer of the next card to be punched.

f) if the card has check failed:
to (48/579) with:
BI00 containing the address of the private store of the card punch (less*7)
(62) reset to the address (less *7) in the card punch buffer of the card that check failed.
(65) increased by 0.I
The card punch stopped.

g) if a card has check failed three times or if a spurious punch or check L.A.M. has occured:
to (3/5I4) with:
(5I) containing 2.4
(62) reset as for exit at (c)
(65) containing 0.4
Card punch stopped and disengaged.

Working space BI02, BI06 to BII0.

Subroutines R508, R5I4, R579, R599.

<u>R576</u> Card Punch, Punch Row Interruption.

<u>Purpose</u> To sort out which type of card punch interruption has occured: if a punch row interruption, punch the next row in the card punch buffer.

<u>Registers of fixed store</u> 30

<u>Entry</u> at (I) in interrupt control when a one appears in bit 30 of line 29, type I4 of the peripheral V-store with:
(62) containing address of next row in the card punch buffer to be punched (less *7).

<u>Exits</u> a) If a punch row L.A.M. has occured.
i) Normally to (I/500) with:
BIII containing private store address (less*7)
(62) containing address of next row to be punched (less *7).
L.A.M. extinguished.
or ii) If a I3th punch row L.A.M. has occured:
to (I4/578) with:
BIII containing private store address (less*7)
BII5 containing 0 if card punch 0
or '' I.0 if card punch I
(62) containing address of present row in the card punch buffer (less *7).
All the L.A.M.'s extinguished and the card punch stopped.

b) If an end of card interruption has occured:
to (I/578) with:
BI23 containing the number of end of card L.A.M.

c) If a check read interruption has occured:
to (I/577) with:
BI23 containing the number of the check read L.A.M.

d) If the most significant L.A.M. in the peripheral V-store type I4, line 6 does not refer to the card punch, to monitor [here labelled (99/599)] with:
BI23 containing the number of the most significant L.A.M.

e) If the peripheral V-store type I4 line 6 is found to contain no L.A.M.'s:
to (I/500)

<u>Working space</u> BIII to BII4, BII6 and BI23.

<u>Subroutines</u> R500, R577, R578, and R599.

2.9.63

R577 Card Punch, Check Read Interruption.

Purpose To check that the appropriate row on the previous card was punched correctly; to store in (53) any discrepencies between the buffer and the card; to avoid checking while the first card is being punched.

Registers of fixed store 24

```
Entry  at I)  in interrupt control after a check read interrupt has
              occured with:
              BI23   containing the number of the check read L.A.M.
              BII5   containing 0 if Card Punch 0
               or       ''      I if  ''    ''  I
              (53)   containing a one in any column which has
                     checked failed on previous rows of this card.
              (62)   bits 23-3  containing address of next row in
                                card punch buffer to punched (less *7).
                     bit 0 = 0  if the first card is being punched.
                           = I  if there is a card available for checking.

Exits   a)    to (I/500) with:
              BIII   containing address of private store (less *7)
              (53)   containing a one in any column which has
                     checked failed on this card.
              (62)   unchanged.
              With the L.A.M. extinguished.

        b)    to (I4/578) if more than I2 check read L.A.M.'s have occured
              on one card with:
              BII5   unchanged.
              BIII   as at exit (a)
              (53)   as at exit (a)
              (62)   as at exit (a)
              With the L.A.M. extinguished and the punch stopped.
```

Working space BIII, BII3 to BII8, and BI23.

Subroutines R500, R578, and R599.

R578 Card Punch End of Card Interruption.

Purpose To check that the end of card has been reached correctly; to test if there are any more cards in the card punch buffer, and if not, to call the card punch P.E.R. to the S.E.R. queue; to test if a check fail has occured and if so, to repeat the card three times before returning to program scan R20I, with (5I) set to 2.4

Registers of fixed store 32

Enter at I) in interrupt control when a one occurs in bit 24 of line 6 type I4 of the peripheral V-store with:

BII5 containing 0 if card punch 0 interruption.
or '' I if '' '' I ''
BI23 containing number of the end of card L.A.M.
(53) containing a one in any column which has check failed.
(62) bits 23-3 containing address of the last row in the card punch buffer (less*7), the first half word of which should contain 6.0 if there are more cards to be punched, and 6.2 if this is the last card in the buffer.
bit 0 = 0 if the first card is being punched.
= I if there was a card to be checked.
(65) containing a count of the number of times this card has already been attempted.

Enter at (I4) in interrupt control if an extra punch row or check read interruption occured with:

BIII containing private store address of card punch (less*7)
BII5 as at entry (I)
(62) containing address of present row in the card punch buffer (less*7)

Exits a) if the end of card was reached correctly, there were no check fails and there is another card to punch: to (I/500) with:

BIII containing address of private store (less *7)
(62) increased to the address of the first row of the next card (less*7)
(65) containing zero.
The L.A.M. extinguished, and the card not offset.

b) if the end of card was reached correctly, there were no check fails and this is the last complete card in the card punch buffer: to (4/20I) with:

BIII containing address of private store (less *7)
BII2 '' entry address to card punch P.E.R. R 579.
(5I) '' -0.4
(62) as for exit (a)
The L.A.M. extinguished, card punch stopped and the card not offset.

c) if the previous card check failed, or the end of card L.A.M. occured in the wrong place, or an extra check or punch L.A.M. occured:
to (4/20I) with:
BIII containing address of private store (less *7)
BII2 '' entry address to card punch fault testing R575 at label (I5)
(5I) containing 2.4
(65) containing the number of times this card has already check failed or 0.4 if an extra L.A.M. has occured.
(62) containing address (less*7) of the position punching has reached in the card punch buffer.
The L.A.M.'s extinguished, card offset & card punch stopped.

Working space BIII to BII4, BII6, BII7, and BI23.

Subroutines R50I, R20I, R575, R579 and R599.

R579 Card Punch P.E.R.

Purpose To convert the characters stored in the (main store) block into the appropriate card code and to store them in the card punch buffer: to start the card punch if the card punch buffer is full or if there are no more characters available in the (main store) block.

Registers of fixed store 277

Enter at I) after all the complete cards in the card punch buffer have been punched with:

BI00 containing address of private store (less *7)

(50) bit 23=0 if there is no carriage control character at the end of the record currently being transfered to the output buffer of the card punch.

=I if there is a carriage control character at the end of the record currently being transferred to the output buffer of the card punch.

bits 5-0 contain the carriage control character or zero.

(5I) containing -0.4 if the card punch is stopped waiting for more information or >0 if it has been stopped with a fault.

(52) bits 23-3 containing address of code conversion table.

bit 0=0 if code conversion is required.

=I if the output is binary and no code conversion is required.

bit I containing 0

bit 2=0 if the last character at the previous entry to this routine was in inner shift.

=I if the last character at the previous entry to this routine was in outer shift.

(56) containing the number of characters in the present record still to be transferred to the card punch buffer.

(60) containing the starting address of the card punch buffer (less *7)

(6I) containing the end address of the card punch buffer (less *7)

(62) bits 23-3 containing the address (less *7) of the row in the card punch buffer for the next card punch interruption.

bit 0 = 0 if there is no card to be checked.

= I if there is a card waiting to be checked.

(64) containing address in the (main) store of the next character to be sent to the card punch buffer.

(66) containing address for return to the Output Master Routine when output completed.

(67) containing the number of characters already stored in the buffer.

(68) containing the V-store address of the card punch (less *6)

Note A restart address in case this routine is interrupted by non-equivalence must be set in (5/20I) before entering this routine.

Entry at (2) After a new (main store) block of information to be punched has been obtained with:

BI00 containing private store address of card punch (less *7)
(50) containing zero.
(5I) containing -0.4
(52) as for entry at (I)
(56) containing zero.
(60) as for entry at (I)
(6I) as for entry at (I)
(62) as for entry at (I)
(64) containing address in the (main) store of the next separator.
(66) as for entry at (I)
(67) as for entry at (I)
(68) as for entry at (I)

Entry at (47) if the card punch was found to be disengaged without a fault, while trying to punch with:
BI00 containing private store address of card punch (less*7)
(62) containing position in card buffer from which punching is to be continued.

Entry at (48) if the previous card check failed and is required to be repunched with:
BI00 containing private store address of the card punch (less*7)
(62) containing the position in the card punch buffer from which punching is to be recommenced.
(65) containing the number of times this card has check failed.

Code conversion The code conversion table provided here converts from internal code to Fortran Card Code. If any other code conversion is required the appropriate table should be provided as shown below and:
(52) bits 23 - 3 should contain the address of the 'look up' table.

' Look up' table consists of 7.0 words of store containing the information described below:
Half word 0.0 contains the address of the inner shift table.
0.4 contains the address of the outer shift table.
5.0 contains the address of the carriage control table.
The rest of the table contains address's required by the routine and specified in R5I7 and R527.

R579 continued...

The code conversion table must contain in each half-word:-

bits 23-2I   as explained in R5I7.

bits 20-I5   contain a number 2 n where n is the row containing the first hole in the particular column of the card:-
+ zone if required is the first row so contain 2
3 zone '' '' '' '' sixth row '' '' I2
6 zone '' '' '' '' ninth row '' '' I8

bits I4-9   contain a number 2 m where m is the row containing the 2nd hole in the column.

N.B.   bit 9 Will always be 0 when calculating 2m and consequently can be used to define a further 'hole' if required.

bits 9-0   contain a one in the relevant position for subsequent holes commencing at the least significant end: ie
bit 0   contains a one if a hole is required in row (m+I)
bit I   contains a one if a hole is required in row (m+2)
all unused positions containing zero.

Example

The code for is a -zone, a 3 zone and an 8 zone ie

						n					m												
	23	22	2I	20	I9	I8	I7	I6	I5	I4	I3	I2	II	I0	9	8	7	6	5	4	3	2	I	0
bits 23-2I	I	0	0																					
- zone				0	0	0	I	0	0															
3 zone										0	0	I	I	0	0									
8 zone															0	0	0	0	0	I	0	0	0	0
	I	0	0	0	0	0	I	0	0	0	0	I	I	0	0	0	0	0	0	I	0	0	0	0
		4			0			4			I			4			0			2			0	

bits 23-21 ~~see~~ R517
bits 20-16 : n = row containing 1st hole
bits 15-10 : m = " " 2nd hole
bit 0 = 1 if hole reqd. in row m+1
bit 1 = 1 " " " " " m+2
⋮ m+10
bit 9 = 1 if " " " "

2.9.63

Exits a) if all the information in the (main) store up to a zero separator has been transferred to the card punch buffer and there is at least one complete card ready to punch:
to program scan R202 with:-
BI00 containing address of private store of card punch (less *7)
(50) containing zero.
(5I) containing zero.
(52) bits 23-3 unaltered.
bit 2 updated.
bits I-0 unaltered.
(53) containing zero.
(56) containing zero.
(60) unaltered.
(6I) unaltered.
(62) bits 23-3 containing address of the first row of the next available card in the card punch buffer.
bit 0 unaltered.
(64) updated.
(65) containing the number of times this card has already been attempted.
(66) unaltered.
(67) updated.
(68) unaltered.
The Card punch started provided that it is not disabled, out of cards or overdue.

b) if all the information has been transferred as in exit (a) but there is not a complete card in the punch buffer:
to address in (66) with:
BI00 and private store set as for exit (a) except:-
*(66) contains zero.

c) if the card punch buffer is full:
to Program Scan R202 with:
BI00 containing private store address of card punch (less *7)
(50) updated.
(5I) containing zero.
(52) as at exit (a)
(53) containing zero
(56) updated.
(62) as at exit (a)
(64) updated.
(65) containing zero.
(66) unaltered.
(67) updated.
(68) unaltered.
The card punch started provided that it is not disabled, overdue or out of cards.

Note	The last complete card in the card punch buffer has a one in bit I, corresponding to the stop bit of the punch and is used by the end of card interruption to detect the end of buffer.
Working space	BI00 to BII0 Four half words of supervisor extracode working space labelled (80),(8I),(90) and (9I).
Subroutines	R20I, R202, R3I2, R3I8, R5I2, R5I4, R5I5, R5I6, R5I7, R5I8, R520, R52I, R523, R527, and R599.

R585 Teleprinter fault test.

Purpose This is a subroutine of the Peripheral One Second, R508. It examines the control bits of the teleprinter to find if the equipment needs attention, and returns to the Peripheral One Second with the appropriate information.

Registers of fixed store None:

This routine is identical with the Teletype Fault Test, R570.

R586 Teleprinter interruption. (Teleprinters 0-7)

Purpose To send the next character from the teleprinter's buffer to the teleprinter; and to stop the printer when the last character is reached.

Registers of fixed store: 13

Enter in interrupt control at (1) when a one appears in bit 25 of peripheral V-store, type 15, line 13, with:

(62) containing the address (less *7) of the next character in the teleprinter buffer.

Exits in interrupt control.

a) Normally to R500 with:

(62) updated.
The L.A.M. extinguished.

b) to R201 after a character has been found in the buffer with bit 1 containing 1, with:

B111 containing address (less *7) of teleprinter's private store.
B112 containing entry address to teletype P.E.R. R573.
(51) containing -0.4
(62) updated.
The teleprinter stopped.
The L.A.M. extinguished.

c) to R500 if the peripheral V-store, type 14, line 1, contains no L.A.M.'s.

Working space B111 to B114, B123

Subroutines R201, R500, R573, R599.

R591: Select input/output

Purpose: An SER in main store entered via R590 to select a new input or output stream for the current program. Also entered to step on next block of the current stream, with an immediate exit to R592. Working registers in subsidiary store are preserved for the previous stream and set up for the new stream, and the current blocks are protected in core store. Exit is to extracode control with the current block of the new stream in core store.

Registers of main store: 47

Instructions obeyed: 2 if exit to step to next block
Around 70+ search of block directory (7 orders per program entry) if new select.

Parameters Used: (1) to (19)

(7) = Location of working registers in current block relative to start of block .7 half words for storage of current parameters, in area 0 - 7.4 of block. Alternate half words from word 1.

(18) = Monitor marker "Input not defined" = 11.6

Cross references:

(2)	=	(1/592)	Step input block
(3)	=	(70/595)	Input working registers in subsidiary store.
(4)	=	(70/596)	Output working registers in subsidiary store
(5)	=	(13/590)	Input block label
(6)	=	(14/590)	Output block label
(8)	=	(2/312)	Contents PAR
(9)	=	(6/203)	Page directory
(11)	=	(4/203)	Program store directory
(12)	=	(2/203)	Block directory
(13)	=	(14/700)	Monitor
(14)	=	(12/227)	Input/output blocks : Directory record
(15)	=	(5/201)	Re-entry address
(16)	=	(2/590)	Call in/out block to core store
(17)	=	(4/247)	Exit and reset full recovery switch

Connections with other routines:

Entry at 1) via R590 with
B96 digits 21 = 00 Select new input
10 Select new output
01 Step input
11 Step output
B119 = new stream (digits 7-3). Remainder irrelevant
B100 = 0 (Input) 0.4 (Output)
B106 = Page no. of current block of current stream
B107 = Location of this block relative to start of block directory
B108 = block label (input or output) with digit 23 = 1
Current block of current stream in core store with PAR locked out.
Full recovery switch set.

R591: continued

Exit: If "step to next block", exit to (1/592) with
B101 = 0.2
Remaining registers unaltered

Otherwise to monitor if new stream not defined, at (14/700)
B100 = 11.6 (Input)
12.2 (Output)
B126 odd
Exit thence to monitor sequence

To exit to main program when new stream selected, via (4/247) to reset full recovery switch. Exit thence to Extracode control as in B97.
Only B96 of extracode registers is altered.

Alternative entry at (19) to call new stream without preserving old stream (used at start of program)
B96, 119, 100 as for entry at (1)
Exit conditions as for entry at (1)

Subroutines:

"Call current block to cores"

Entered at (2/590) with
Digit 2, B96 = 0 (Input) : (Output)
B101 = Return address
Exit to resume at re-entry address if block not in cores
If in cores, to return address with
B100 = (Input) 0.4(Output)
B106 = Page number of block (digits 23-3)
B107 = Location in block directory relative to start of block directory
Remaining B registers irrelevant

Temporary working space: B100 - 110

Notes:

1. The current stream numbers are recorded in two successive half words of the program working area in subsidiary store (70/595)

2. Half words 70/595 to 76/595 (input) or 70/596 to 76/596 (output) are preserved in alternate half words 1, 2, ...7 of the current block, and are reset in subsidiary store from the current block on new selection.

3. R591 resets the record of the location of the current block in the program working area.

4. The current block of input/output n is recorded in the program area of the block directory as having block label

Digits 22-19 = 1
Digits 18 = 0
Digits 17-13 = stream n
Digits 12 = 0 (Input)
1 (Output)

R591: continued

5. On conclusion of the routine, the current block of the new stream is in core store with the correct (locked out) page address register. The page directory of the current block of the previously selected stream is locked out and PAR set to *4. It will not be altered by the program changing sequence.

6. It is assumed that words 1 to 7 of the leading block of each stream are set up appropriately at the time the stream is defined.

7. On exit to monitor with stream not defined, the full recovery switch remains set. B119 digits 7-3 specifies the stream number, the remaining digits are irrelevant.

R592: Step input block

Purpose: An SER in main store entered via R590, 591, to step to the next block of input or output. An immediate exit is made to R593 for output. For input, the next block is obtained from the input well or from a buffer block supplied by a fast direct input device. In the latter case a routine is initiated to refill the buffer block. The current block of input is lost from main store. An analysis routine is entered when the end of a document is reached to select the next section or to monitor.

Registers of main store: 56

Instructions obeyed: 5 if entered for output
57 if entered for input stream (normal)
50 if entered for input stream direct.

Parameters used: (1) to (24)

(2) = 0: Location within block of key word "Next block"
(3) = 7.4: Location within block of key word "Counter or peripheral"
(4) = 6.4: Location within block of key word "Supply routine (see notes)

Cross references:

(5)	=	(1/593)	Step output
(6)	=	(2/203)	Block directory
(7)	=	(3/203)	Block timers
(8)	=	(6/203)	Page directory
(9)	=	(2/312)	Contents PAR
(10)	=	(4/203)	Store directory
(11)	=	(8/227)	Monitor directory
(12)	=	(5/201)	Re-entry address
(13)	=	(1/318)	Call to cores
(14)	=	(8/303)	Page timers
(15)	=	(13/590)	Input block label
(16)	=	(5/203)	Current program
(17)	=	(24/227)	Find BD of input
(18)	=	(5/202)	Program scan exit
(19)	=	-3(23/205)	Amend free block count
(20)	=	(4/247)	Reset full recovery switch and exit
(21)	=		End input stream
(23)	=	(75/595)	Current addresses for extracodes
(24)	=	(76/595)	

Connections with other routines:

Entered at (1) via R590, 591 for "step to next block" with
B96 digit 2 = 0 (Input) 1(Output)
B100 = 0 (Input) 0.4(output
B106 = Page of current block
107 = Location in block directory relative to start of block directory of current block
108 = block label + *4 : (13/590) *4

1.2.64

R592: continued

Exit a) If output block, to (1/593) with
B101 = block label of output
B102 = "Next block")
103 = "Counter") recovered from current block
104 = "Supply routine")
Other B register unaltered

b) If end of stream (see note 1) to "End input stream"
With full recovery switch set,
B registers as for exit (a)

c) Otherwise to program scan, via (4/247), to reset full recovery switch.
Program Scan exit = 0 (supply from input well)
= address of supply routine (23-3);
queue base (1,0) if direct supply.

Exit thence to supply routine (which occupies the same block as R592) with

B100	=	Directory location (relative to start of BD) of block to be filled
B101	=	0 if continue reading, non zero if end
B102	=	Block label of block to be filled (digits 22-12) program number (digits 8-2) rest zero
B103	=	Peripheral number
B107	=	Program number (digits 20-14). Other digits irrelevant.

Subroutines:

a) "Call to cores", entered at (1/318) with

B110 = Return address
B109 = block label (22-12)
Program number (8-2) (0 or current program)
p 0 = 1 (operand)
p 9 = 1 (do not change timer)

Exit to recover at re-entry address if block on drum or if locked out, not program 0

Otherwise to return address with
B109 = Page number of block
PAR = *4

b) "Find directory entry of input block" entered at (24/227) with

B110 = Current program number
B109 = Return address
B100 = 0

Exit (block allocated) with
B107 = location of current input block relative to start of block directory.
B106 Altered

<u>R592:</u> continued

c) "Amend free blocks" entered at -3(23/205) with

B110 = Return address
Exit to return address with B105-109 altered.

<u>Temporary working space:</u> B100 - 110

<u>Notes:</u>

1. Three half words in the current input block (2) (3) (4) hold

   a) If supply from input well

   (2) = block label of next block within supervisor area, (22-12) remaining digits zero.
   If current block is end of document <u>or</u> end of part of document held in store, digit 23 = 1

   (3) = Counter, number of blocks read (digits 23-3)

   (4) = 0

   b) If supply directly (e.g. from Ampex tape)

   (2) = block label of next block in program area (22-12) (= block label of current less *01).
   Program number (8-2) rest zero

   (3) = Peripheral no. (e.g. tape deck no. etc)

   (4) = Address of SER to supply block, digits 23-3
   Digits 0,1 = SER queue on which based

   11 = Slow
   01 = Tape etc.

   These half words are up dated to the next block by R592, and are assumed to be set up correctly in the first block of each stream.

2. The supply routine should set (4) to 0 normally, non zero if end of document. It should also lock out the block involved in the transfer until the block is available to the main program.

3. The local process switch is set for the program whilst the next block is located, and is then reset from the master process switch.

4. The current block is lost and one added to the number of free blocks if a block of input well in supervisor store.

5. Start addresses for the extracodes are both set to word 8 of the new input block. This block is in core store on exit with the P.A.R. set to the appropriate locked out block label.

R593: Step output

Purpose: An SER in main store to pass to the output well a completed block of output from an object program and to supply the program with a new block. When necessary the system output tape is activated or the peripheral is restarted. An alternative entry provides similar facilities for SER's.

Registers of Main Store: 64

Instructions obeyed: 33 to 58 when entered from object programs
9 to 34 when entered from SER's

Parameters Used: (1) to (37)

(25)	=	Reason for space: Block of output well
(27)	=	Bit in Output State Record denoting Idle (one digit) = 0.4
(30)	=	Monitor reason "Output exceeded" = 12.6
(31)	=	Start of output within block = 8.0
(32)	=	End of output within block = 511

Cross references:

(5)	=	(2/203)	Block directory
(6)	=	(3/203)	Block timers
(7)	=	(6/203)	Page directory
(9)	=	(5/203)	Current program number
(10)	=	(5/201)	Re-entry address
(11)	=	(35/203)	Block location table
(12)	=	(5/202)	Program scan exit
(13)	=	(2/592)	(
(14)	=	(3/592)	(Location of link parameters in block
(15)	=	(4/592)	(
(16)	=	(14/590)	Output block label
(17)	=	(1/630)	Aquire block
(18)	=	(24/227)	Locate current output block
(19)	= 1	(2/590)	Call output to cores
(20)	=	(12/503)	Store location and lock out
(21)	=	(1/251)	Analyse output tape
(22)	=	(2/700)	Monitor
(23)	=	(4/247)	Reset full recovery switch and exit
(24)	=	(43/599)	"Output state record" in private store of peripheral
(26)	=	(1/263)	Output recorder
(28)	=0.3	(1/269)	Restart peripheral
(29)	=0.1	(1/470)	Start output tape
(33)	=	(72/596)	Extracode working space for start
(34)	=	(75/596)	
(35)	=	(76/596)	Extracode working space for end
(36)	=	(2/253)	Exit for output from SER

R593: continued

Connections with other routines:

Entered at (1) from R592 when output block supplied by object program.

with B102 = "Next block")
B103 = "Counter) As recovered from current
B104 = "Supply routine") output block
B100 = 0.4

Exit to Program Scan by jump to (4/247) to reset full recovery switch.
Program scan exit zero or set to

a) Output recorder Digits 1, 0 = 00. "Output recorder" entered at (1/263) with
B101 = Block label of block produced in output well, p 22-12. Rest zero
B100 = 0.4
B102 = contents of word (13) of block (see notes)
B103 = Count of blocks less 1, digits 23-3
B104 = Contents of word (15) of block (see notes)

b) Start output tape. Digits 1, 0 = 01. "Start output tape" entered with
B101 = Block label of block produced in output well, p 22-12. Rest zero
B102 = Block label of previous block of document, p 22-12. Rest zero.
B103 = Counter of no. of blocks to go, digits 23-3
B104 = Contents of word (15) of block
B105 = Location in block directory relative to start of directory of previous block, digits 12-2. Rest zero.

c) Restart peripheral. Digits 1, 0 = 11 "Restart peripheral" entered with
B96 = Block label of block produced in output well, p 22-12. Rest zero.
B101 = Block label of previous block of document, p 22-12. Rest zero.
B103 = Counter of no. of blocks to go, digits 23-3
B104 = Peripheral stream (word (15) of block)
B105 = Location in block directory relative to start of directory of previous block, digits 12-2. Rest zero.

d) Private output control Digits 1, 0 = 11 or 01. Entered with
B96 = Block label of block produced in output well, p 22-12. p23 = 1. Rest zero
B100 = 0.4
B101 = B102 = Contents of word (13) of block. Odd if first block; even if continuation (see notes)
B103 = Peripheral stream (e.g. channel number)
B105 = Location on block directory relative to start of directory of previous block, digits 12-2. Rest zero.

R593: continued

Alternative entry at (37) from R253 with

B102, 103, 104 = contents of link word (13) (14) (15)
B100 = Working area of peripheral less *7 (digits 23 = 1)
B96 = Block label of new block, digits 22-12.
Rest irrelevant.

Exit to (2/253) with program scan exit set up as for entry (1)
B100 unaltered
B96, 103, 104 = new contents of link words (13) (14) (15)

Subroutines:

(a) Aquire block entered at (1/630) with

B101 = Reason for request (Output well)
B110 = Return address
Exit to re-entry address when block made available, after halting, and finally to return address with
B105 = Directory location of new block relative to start of block directory, digits 12-2
Block directory = block label, p 22-12. Rest 1.
B106 - 110 Bt altered.

(b) Locate current output block entered at (24/227) with

B100 = 0.4
B108 = *4
B109 = Return address
B110 = Current program number, digits 8 - 2
Exit with B107 = Directory location of current output block, digits 12-2. Rest zero
B106 = Page number of block (digits 11-3). Rest zero if in cores
= Negative if on drum
If in cores, PAR set to *4
No other B register altered

(c) Call output to cores entered at 1(2/590) with

B100 = 0.4 (output)
B101 = Return address
Exit to program scan and thence to re-entry if block not in cores.
If in cores, exit to return address with
Full recovery switch set
B100 = 0.4
B106 = Page no. of output block (current block of selected stream), digits 11-3
B107 = Directory location of block relative to the start of the block directory, digits 12-2
B110 = Current program number

R593: continued

(d) "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 block directory (digit 12-2)
B105 - 110, Bt altered

(e) "Analyse system output tape" entered at (1/251) with

B126 even
B110 = Return address

Exit to return address with

B109	=	0	if tape present, busy
		non zero and even	if tape not present
		odd	if tape present, idle.

(f) "Monitor" entered at (2/700) with

B126 odd
B100 = reason (output exceeded, = 12.6)
B102 = return address

Exit to return address with B lines unaltered

Temporary working space: B96, B100 - 110, Bt

Notes:

1. Extracode working space in subsidiary store is reset by this routine to the beginning and end of the new block.

2. One and only one of the various routines listed in the exit conditions is entered on conclusion. The start of these routines must be in the same block as R593 if B105 is required. Exit (a) continues on the same SER base. The remaining exits change the base, leaving the object program free to proceed.

3. If this is the first block in the output stream, output recorder is entered. Otherwise the block is linked to the previous one in store by setting the block timer of the previous block to the block label (digits 22 -12). Digits 23 = 0 if this is a normal peripheral stream, 1 if a fast stream, not using the system output tape (e.g. a private Ampex tape). The remaining digits are zero. The block timer of the new block created is set to zero. The lockout digits are set to 2 if no output tape, 3 if output tape in use.

R593: continued

4. After linking the block, a "private" SER may be entered e.g. to control an Ampex tape. This is only entered if the block timer of the previous block is non-zero (i.e. has been altered by the tape control routine).

5. For normal streams, the output tape is analysed. If present and busy no action is taken. If present and idle, it is activated via exit b. If the tape is not present, the peripheral being supplied (if known) is activated via exit c.

6. The course of action is dictated by the link words recorded in the current block and copied and updated by R593. These contain the following

(2/593): Digits 23 = 1 if private (bypass output tape),
0 otherwise
Digit 0 = 1 if 1st block of stream. 0 otherwise
Digits 22-12 = block label of previous block, in supervisor area (if digit 0 = 0)

(3/593): If private, peripheral no. (e.g. channel no.)
Otherwise = count of number of blocks permitted (digits 23-3). The program is monitored when this reaches zero but after supplying the new block. Thus monitor output may continue into the new block. If this is filled, standard monitor output may extend into another new block.

(4/493): If private, address of SER to deal with output, digits 23-3. Digits 0, 1 = SER queue number.

Otherwise, = 0 if not coupled to peripheral
= private store of peripheral otherwise.

It is assumed that these are set up initially when the output stream is defined. They occupy three half word in the output block in addresses 0, 7.4, 6.4 respectively.

7. The alternative entry at (37/593) is used by SER's to set up the same exit conditions as described above, entry and exit being via R253 (which loads and unloads link words from B registers).

R594: Use tapes as slow peripheral

Purpose: To arrange input and output via direct magnetic tapes in conjunction with input-output extracodes. The information is assumed arranged within each block as for peripheral input-output blocks.

Registers of main store: 44

Instructions obeyed: Input : 15
Output : Around 26 per block plus subroutines (see below)

Parameters used: (1) to (19)

Cross references:

(6)	=	(2/203)	Block directory
(7)	=	(3/203)	Block timers
(8)	=	(4/203)	Program store directory
(9)	=		Variable tape directory
(10)	=	(5/201)	Re-entry
(11)	=	1(16/421)	Order to tape queue
(13)	=	6(14/317)	Lose block (alternative entry)
(14)	=	-3(23/205)	Accumulate free blocks
(15)	=	(1/216)	Assign tape exit
(16)	=	(12/203)	Store location and lock out
(17)	=	(35/203)	Block location table
(18)	=	(1/202)	Program scan
(19)	=		Output controller : Private tape free

Connections with other Routines:

Entered at (1) via R592 via Program scan exit with
B100 = Location of block to be filled relative to start of block directory, digits 12-2. Rest zero
B101 = 0 if continue input
Otherwise non zero if end of document on previous block.
B103 = Absolute deck number, digits 7-3. Remainder zero
B107 = Program number, digits 20 - 14.
Remainder irrelevant
B102 = Program number, digits 8 - 2
Remainder irrelevant

Exit: To enter to tape queue at 1(16/421) with re-entry address set
B100 = deck number
B101 = program number 21-15
directory location of block relative to start of program area, 13-3. digits 23 = 1. Remainder zero
B104 = *000606 (Read forwards)
Exit thence to program scan.

R594: continued

Entered at (2) from R593 via program scan exit with

B96 = block label of block produced in supervisor area, digits 22-12.
digits 23 = 1 if normal
23 = 0 if last block of document
Remaining digits irrelevant

B102 = digit 0 = 0 (continuation)
1 (start)
Remaining digits irrelevant

B103 = deck number, absolute, digits 7-3

B105 = location in block directory (relative to start of block directory) of previous block produced (in Supervisor area), digits 12-2.
Remaining digits zero.

Exit: a) When last block concluded, to output controller at (19) with B100 = Deck number, digits 7-3

b) Normally : To program scan when no more blocks to be transferred.

Subroutines:

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

B100 = Return address
B105 = Directory location of block relative to start of block directory.

Exit to return address with B101-103 unaltered

b) "Accumulate free blocks" entered at -3(23/205) with

B110 = Return address
Exit to return address with B105-109, Bt altered

c) "Store location" 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 directory location of block relative to the start of the block directory in block location table. B100-104 preserved.

d) "Assign tape exit" entered at (1/216) with
B100 = deck number
B110 = return address from R216
B109 = exit address after tape transfer, digits 23-3.
digits 2-0 = 1

Exit to return address in B110. B100 - 105 preserved.

R594: continued

e) "Order to tape queue" entered at 1(16/421) with re-entry address set in case of halt for full tape queue

B100 = deck number, digits 7-3
B101 = block label, digits 22-12. Rest zero
B104 = *2006054 (Write + exit at end of transfer to supervisor).

Exit to program scan. Eventual exit to assigned address with B100 = deck no.

R595 Input extracodes.

Purpose To transfer characters from the input well to the main programme. Includes extracodes 1050, 1051, 1052, 1053, 1054, 1056, 1057.

Registers of fixed store 193 excluding initial jumps.

Entries at	(40)	for	extracode	1050
	(43)	''	''	1051
	(44)	''	''	1052
	(45)	''	''	1053
	(1)	''	''	1054
	(20)	''	''	1056
	(21)	''	''	1057

Parameters in subsidiary store

(70) bits 23-3 contain number of currently selected input stream.
bits 2-0 contain zero.

(71) contains a copy from the input well of the half word currently being unpacked. The next character due to be passed to the main programme is at the most significant end.

(72) bits 23-3 contain an address in this routine depending on the position within a record of the half word which is due to be unpacked next.
bits 2-0 contain bits depending on the position within a half word of the character which is due to be passed to the main programme next.

(73) bits 23-3 contain an address in this routine depending on whether there is a carriage control character at the end of this record, or whether the record continues elsewhere.
bit 0 = 1 if the record currently being transferred to the main programme is a binary record.

(74) contains the V-store address of the reader from which the input originated.

(75) contains the address of the next half word in the input well.

(76) contains the end address of the current record.

Subroutines

R590. Entered after non-equivalence to call the current input block to cores. (Entered with l.s. octal digit of B126 containing zero) Return to address in B97.

R591. Entered from extracode 1050 'Select input stream' when the newly requested input stream number does not correspond with the present input stream number.
Return to address in B97 with (70) to (76) replaced with values appropriate to the new input stream.
If this is the first time the stream has been selected then:
(76) contains the same as (75):
(72) contains the address (14/595) 0.7:
(70) bits 23-3 and (74) are set appropriately.

R592. Entered after a zero separator has been encountered (indicating the end of information in the current block) and the main programme requests still more characters.
Returns to address in B97 with (75) and (76) containing address of first separator in new input block.

Note

These extracodes cause non-equivalence in interrupt control if the current input block is not in core store. Before doing this, B126 is copied into the half word of subsidiary store labelled (79) and the cause of interruption can be checked by comparing these two registers. If equal, extracodo control should be resumed at this address.

Working space

Extracodes	1050	B91, B95 to B97, B119	
	1051	Ba only.	
	1052	Ba only.	
	1053	B94 to B97, B119, Ba	
	1054	B94 to B97, B119, Ba	
	1056	B91 to B97, B119, Ba)	Also extracode
	1057	B91 to B97, B119, Ba)	working space in subsidiary store.

Exit on extracode control

Extracode 1050, 1053 and 1054 can be used as subroutines of other extracodes provided the return address is set in B127 and has bit 23 = 1.

Extracodes 1056, 1057 cannot be used in this way.

Extracodes 1051, 1052 are trivial one order extracodes and can be copied into other extracode routines if required.

Entry and Exit conditions are:

1050: Enter at (41) with:
B91 containing number of input stream required.
B127 containing return address (Bit 23=1)
Exits in extracode control to address in B127
Uses B91, B95 to B97, B127.

1053: Enter at (45) with:
B127 containing return address for internal code (bit 23=1)
B91 containing return address for binary.
Exits in extracode control to addresses in B127 or B91.
Uses B91, B94 to B97, B127.

1054: Enter at (1) with:
B127 containing normal return address (bit 23=1)
B91 " return address at end of record.
Exits in extracode control to addresses in B127 or B91 with:
B94 containing character in bits 5-0
Uses B91, B94 to B97, B127.

R596 Output extracodes.

Purpose To transfer characters from the main programme to the output well. Includes extracodes 1060, 1061, 1062, 1064, 1065, 1066, 1067.

Registers of fixed store 189 excluding initial jumps.

Entries at	(60)	for	extracode	1060
	(43)	''	''	1061
	(44)	''	''	1062
	(1)	''	''	1064
	(45)	''	''	1065
	(20)	''	''	1066
	(21)	''	''	1067

Parameters in subsidiary store

(70) bits 23-3 contain number of currently selected output stream.
bit 0=0 if current selection is internal code.
=1 if binary.

(71) contains the half word currently being packed. The next available charcter space in this half word is at the m.s. end. The last available character space contains 4.0 until it is overwritten by the last character.

(72) contains the address in the output well reserved for the separator at the beginning of the record which is currently being filled.

(73) contains the count of complete records sent out by the main programme so far,

(74) contains V-store address of the output peripheral which is to be used,

(75) contains the address of the next half word space in the output well.

(76) contains the end address of the space in the output well.

Subroutines

R590 : Entered after non-equivalence to call current block to cores. (Entered with l.s. octal digit of B126 containing 4)
Returns to address in B97.

R593 : Entered from extracode 1060 'Select output stream' when the newly selected output stream number does not coincide with present output stream number.
Returns to address in B97 with (70) to (76) replaced by values appropriate to the new output stream. If this is the first time this stream has been selected, then:
(72) contains the same as (75):
(73) contains zero:
(70) bits 23 to 3 and (74) are set appropriately.

R594 : Entered when the end address of the space in the output well has been reached.
Returns to address in B97 with new values set in (75) and (76)

Note

These extracodes cause non-equivalence on interrupt control if the current output is not in core store. Before doing this B126 is copied into the half word of subsidiary store labelled (79/595) and the cause of interruption can be checked by comparing these two registers. If equal, extracode control should be resumed at this address.

Working space

Extracode	1060	B91, B94 to B97, B119, & first half word of extracode working space in subsidiary store.
''	1061	Ba only.
''	1062	Ba only.
''	1064	B94 to B97, B119.
''	1065	B94 to B97, B119.
''	1066	B91 to B97, B119, Ba) Also extracode
''	1067	B91 to B97, B119, Ba) working space in subsidiary store.

Exits on extracode control

Extracodes 1060, 1064, 1065 can be used as subroutines of other extracodes provided the return address is set in B127 and has bit 23=1.

Extracodes 1066, 1067 cannot be used in this way.

Extracodes 1061, 1062 are trivial one-order extracodes and can be copied into other extracode routines, if required.

Entry and exit conditions are:

1060 : Enter at (61) with:
B91 containing number of output stream required.
B127 containing return address (bit 23=1)
Exits in extracode control to address in B127.
Uses B91, B94 to B97, B127 and first half word of extracode working space in subsidiary store.

1064 : Enter at (2) with:
B94 bits 5-0 containing character for output.
bits 23-6 '' zero.
B127 containing return address (bit 23=1)
Exits in extracode control to address in B127.
Uses B94 to B97, B127.

1065 : Enter at (42) with:
B94 bits 5-0 containing character for output.
bits 23-6 '' zero.
B127 containing return address (bits 23=1)
Exits in extracode control to address in B127
Uses B94 to B97, B127.

R599 Working space for peripheral routines.

Purpose No programme is obeyed. This routine number is used only for convenience in specifying the private store and working space used by peripherals.

Private store

The addresses (50) to (68) are successive half words of (subsidiary) core store, addressed relative to the first half word.

(68) = 0.0 : contains V-store address of peripheral (less*6)

(66) = 0.4 : contains return address when input/output finished or information on reason for return has been made.

(60) = 1.0 : contains starting address (less *7) of input/output buffer.

(61) = 1.4 : contains end address (less *7) of input/output buffer.

(64) = 2.0 : contains address of next space available in store (input); address of next character to be output from store (output)

(65) = 2.4 : contains end address of store (reserved block number less *7)

(62) = 3.0 : contains next address in buffer (less *7) for interrupting routines.

(67) = 3.4 : contains next address in input/output buffer for P.E.R's (less *7)

(56) = 4.0 : contains address (less *7) reserved in store for next separator (input); number of characters remaining in current record (output)

(51) = 4.4 : contains information on present state of peripheral. M

(52) = 5.0 : contains information on code conversion required.

(50) = 5.4 : used for test *** (input); contains carriage control character at end of current record (output)

(53) = 6.0 : contains information on action required when a punching fault is detected by an input peripheral.

Working space (80)(81)(90)(91)(92) three half words. Used during a P.E.R. to augment the B lines available. Contain no useful information when the P.E.R. is finished.

Miscellaneous constants

(5) = starting address in subsidiary store table where peripheral private store addresses are stored.

(6) = number of such addresses in the table.