I.C.T. ATLAS COMPUTER

SUPERVISOR AND FIXED STORE ROUTINE SPECIFICATIONS

VOLUME 3

ROUTINE 600 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).

JOB DESCRIPTIONS

CONTENTS /2 (VOL 3)

SCHEDULER ROUTINES

1.9.64

R602.

Purpose. Read statements beginning 'Print'

a) PRINT TAPE
(Title)

b) Print TAPE, a/b/c, Title.

No of Instructions.

Cross References.

(30) =	(1/600)	=	Get next Character.
(31) =	(2/681)	=	find beginning of next word.
(32) =	(1/681)	=	Read next in characters.
(33) =	(1/692)	=	Ask for space.
(34) =	(1/682)	=	Read and store title.
(35) =	(2/685)	=	Revise I tape deck.
(36) =	(3/684)	=	Store words 10.0, 11.0, 11.4
(37) =	(1/690)	=	Add title to incomplete job list.
(38) =	(1/621)	=	Translation of beginning of line.
(39) =	(1/686)	=	Read characters a/b/c.
(40) =	(30/619)	=	Store word for word to be put into .10.0

CONNECTIONS.

Enter at 1) if line begins with . PRINT

EXIT. to next line on completion of translation of statement
Can also exit to Error Routine via any of its subroutine.

Notes.

1) Insists that statements be of the form,

a) Print (separator) TAPE (newline)
(TITLE)

b) Print (separator)TAPE(separator)(a)/(b)/(c)(separator)(Title)

2) For (a) forms a 12 word block which has the form.
Words 0-9.4 Title of tape in 6 bit characters
Word 10.0 bits 20-23 = title identifier = * 50
10.4 = Link with next title connected with this job.

3) For (b) forms 12 word block of form.:-

Words 0-9 = = Title of tape in 6 bit characters
Word 10.0 = bits 4-6 = (c)-position in block.
bits 7-19 = (b) block no.
20-23 = identifier = * 40.

10.4 = = Link with next title with this job.
11.0 = = (a) in 6 bit characters.
11.4 =

R603.

Purpose. Read statements beginning -' TAPE'

a) Tape FREE, (n) , (Title)
b) Tape , (n) , (Title)
c) Tape / (m) , (n) , (Title)
d) Tape / (m) , END , (n) , (Title)
e) Tape , (a) /(b) /(c) , (title)

No of Instructions.

Cross References.

(30)	=	(1/619)	=	J marker.
(31)	=	(1/600)	=	Get next character.
(32)	=	(2/683)	=	Read integer.
(33)	=	(31/619)	=	Store register for Word to be stored in
				" " 11.0 of 12 word block.
(34)	=	(30/619)	=	" " 10.0 of 12 word block.
(35)	=	(1/620)	=	Routine 'Error Routine '.
(36)	=	(32/619)	=	Store register for word to be stored in 11.4 OF 12 word block.
(37)	=	(2/681)	=	Routine find beginning of title.
(38)	=	(1/692)	=	Routine 'to ask for store block'
(39)	=	(1/682)	=	Routine 'Read and store Title'.
(40)	=	(3/684)	=	Routine ' Store words 10.0,11.0,11.4.
(41)	=	(1/621)	=	Routine 'find next line'.
(42)	=	(1/690)	=	Routine Entry in incomplete job list'
(43)	=	(1/681)	=	Routine 'get next in characters.
(44)	=	(2/685)	=	Routine 'Add 1 to no of tape decks needed.
(46)	=	(1/686)	=	Routine ' Read (a)/(b)/(c)
(47)	=	(1/691)	=	Routine ' set systems decks'
(48)	=	(40/619)	=	Store used for J marker.
(49)	=	(1/683)	=	

Connections with other routines.

Enter at 1) with statements beginning with TAPE.

Exit - to Translate next line.
Exit - to Error Routine.

Exit - to error routine via any subroutines used.

Notes.

1) Tape (separator) Free(separator)(n)(separator(Title)(newline).
Forms 12 word section,
words 0-9.4 = = Title of Tape in 6 bit characters.

words 0-9.4 = bits 0 - 3 = no of tape document (n)
bits 20 -23 = Identifies = * 24.
words 10.4 = = Link with next title for this job.

Gives Error if a) $n \geq 16$
b) more than 80 characters in title.

NOTES . (continued)

2) TAPE(Separator)(n)(Separator) (Title)
as (1)
except Identifies = * 20.

3) TAPE / (m) (separator) (n) (separator) (Title).

Word	0-9.4 =		=	Title IN 6 bit characters.
Word	10.0 =	bits 20-23 =		Identifier = * 3.
		bits 0 -3 =		(n)
word	10.4 =	link with next title connected to current job.		
word	11.0 =	bits 0 -23 =		(m)
word	11.4 =		=	0

Gives error if
a) n > 16
b) more than 80 characters in title.
c) Incorrect Format.

4) TAPE/ (m) (separator) END (separator) (n)(separator) (Title)
as (3) except word 10, bit 19 = 1 to show END.

5) TAPE (separator) (a) / (b)/(c) (separator) (n) (separator) (Title).

word 0-9.4 = = Title in 6 bit characters.
- here, the title is that of the document stored on tape in block (b)

word 10 = bits 20-23 = Identifies * 14.
bits 0 -3 = (n) no of input document

word 104 = Link with next title

word 11.0 = bits 0 -7 = (a) - here (a) refers to an entry in the Tape section of the job document and is the programmer's tape number not the physical tape identifier.
bits 8-10 = (c) - position in block
bits 11-23 = (b) - block label

Gives error if.
a) n > 16
b) b > 5000
c) more than 80 characters in title.
d) Incorrect FORMAT.

R604.

Purpose.

To interpret statements beginning with an intoger (n). i.e.

a) (n) , (Title)
b) (n) , (peripheral) , (n) Blocks/Line.
c) (n) , TAPE FREE/ (peripheral) , (m) BLOCKS/LINES,(Title)
d) (n) , Tape (b)/ (peripheral) ,m Blocks/LINES, (Title)

No of Instructions.

Cross - References.

(30)	=	(2/683)	=	Alternative entry Routine ' Read integer'
(31)	=	(30/619)	=	Store Location for word to be stored 10.0.
(32)	=	(1/619)	=	Jmarker List.
(33)	=	(1/620)	=	Error Routine
(34)	=	(1/681)	=	Routine'Read next in characters'
(35)	=	(2/681)	=	Routine 'Find beginning of next word'
(36)	=	(1/696)	=	Routine 'Recognize peripheral'
(37)	=	(1/683)	=	Routine 'Read integer'
(38)	=	(1/695)	=	Routine 'Form correct units of length.
(39)	=	(40/619)	=	List of integer switches.
(40)	=	(1/689)	=	Routine'to enter new output device'
(41)	=	(1/621)	=	Routine' Translate next line.
(42)	=	(1/692)	=	Routine' ask for space'
(43)	=	(1/682)	=	Routine ' Read and store Title'
(44)	=	(3/684)	=	Routine 'Set words 10.0,11., 11.4.
(45)	=	(1/690)	=	Routine ' Enter in incomplete job list'
(46)	=	(1/600)	=	Get next character Routine.
(47)	=	(30/619)	=	Store location for word to be stored in 10.0

Cross References.

(48)	=	(2/685)	=	Routine' Add one to no of tape decks needed.'
(49)	=	(33/619)	=	List- 'here,not assigned.'
(50)	=	(1/687)	=	Routine 'Test if title is element of list. '
(51)	=	(1/690)	=	But on incomplete job entry.
(52)	=	(1/694)	=	Give back space.
(53)	=	(34/619)	=	Head of 'not here, assigned' list.
(54)	=	(9/619)	=	Word containing peripheral remotely associated with present input device.
(55)	=	(2/619)	=	Position of job description.
(56)	=	(32/619)		

Connections.

Entered at (1) when first none separator in statements is an integer.
Exit when statements translation is finished.
Exit to error routine.
Exit to error routine via any subroutine used.

Notes.

1) (n)(separator)(Title)(newline)
On being read ,this title is checked against all elements in the'here,not assigned'list of document titles. if in list then is entered in the list of documents associated with this incomplete job.
If not in but, then is entered in the'not here assigned list.

'here , not assigned ' list

Words 0-9	=	=	Title.
Word 10.0	= bits 18-28	=	compiler name (if compiler document)
		=	0 (if data document)
Word 10.4	=	=	link with next title in list.
Word 11.0	= bits 23-3	=	Position of this document on tape.
			" " " " in store.
	bit 0	=	1 if no input tape.
			0 otherwise.
11.4	= bits 10-18	=	No of blocks in this document.

'Not here , assigned ' list

Words 0-9	=	=	Title.
Word 10.0	=	=	Position of job reading this document.
Word 10.4	=	=	Link with next title in list .
Word 11.4	=	=	(n) - input no of this document.

Incomplete job list

If compiler document.

Word 0 - 9	=	=	Title.
Word 10.0	= bit 0-3	=	(n) no of input document.
	4-9	=	Type of compiler.
	10-18	=	No of blocks in this document.
	19	=	
	20-23	=	identifies = * 04.
Word 10.4	=	=	link with next title connected to this job.
Word 11.0	= bits 1 - 0	=	1,1 if document on dump tape.
			1,0 input tape.
			0,0 not on tape.
Word 11.0	= bits 23 - 3	=	position of document on tape.
Word 11.4	=	=	leading block address.

1.9.64

Notes. (continued)

Data document.

As compiler document
except a) no compiler used - word 10.0 bits 4 - 9 = 0
b) identifies word 10.0 bit 20-23= * 10.
Gives Error if input switches no set on reading this format.

2) (n) (separator)(peripheral) (separator)(m)(separator)(BLOCKS)(Lines (newline).
Put entry in output section.
Gives error if a) incorrect format.
b) peripheral tape unknown.
c) n > 16.

3) (n) (separator) TAPE(separator) FREE/(peripheral)(separator(m)(separator)
Blocks Lines (separator)(Title)(newline
Forms entry in incomplete job list.

Words	0-9	=	= Title.
Word	10.0	= bits 0-3	= No of document (n)
		bits 4-7	= type of peripheral.
		20-23	= Identifies - * 54.
	11.0	=	= (m) no of blocks to be printed.

Gives error if a) n > 16.
b) peripheral unknown.
c) more than 80 character in title.

4) (n)(separator) TAPE(separator)(b)/(peripheral)(separator) (m)Blocks Lines (Title

As (3) plus word 11.4 = (b)
Gives error if b> 5000.

8605.

Purpose. To interpret statements beginning ''COPY.''

a) COPY TAPE , (b) ,(title)
b) COPY TAPE FREE ,(title)

No of Instructions.

Cross References.

(30)	=	(2/681)	=	Routine ''Find next character ≥ 20''
(31)	=	(1/681)	=	Routine ''Get next n characters''
(32)	=	(1/620)	=	Error Routine.
(33)	=	(1/600)	=	Routine''Get next character''
(34)	=	(1/683)	=	Routine''Read integer''
(35)	=	(1/692)	=	Routine''Ask for space''
(36)	=	(1/682)	=	Routine''Read and Store Title''
(37)	=	(2/685)	=	Routine''Order and move tape deck''
(38)	=	(1/621)	=	Routine''Translate next line''
(39)	=	(1/690)	=	Routine ''Enter incomplete job list.''
(40)	=	(2/683)		

Connections.

Enter at (1) with statement beginning COPY.
Exit a) to translation of newline.
Exit b) to Error Routine.
Exit to error routine via any subroutine used.

Notes.

1). COPY (separator) TAPE (separator)(b)(separator)(Title).

Incomplete job list entry.

Words 0-9 = = Title.
Word 10.0 = bits 20.23 = identifier = * 34
bits 13-0 = (b)

Gives error if a) b > 5000.
b) incorrect format.
c) more than 80 characters in Title.

2) COPY (separator) TAPE (separator) FREE separator (Title)

as 1).
but identifies = * 44
and word 11.0 = 0.

3606.

Purpose. Translate Lines beginning ''STORE.''

No of instructions.

Cross References.

(30)	=	(37/619)	=	No of blocks allowed to be called for.
(31)	=	(1/600)	=	''Get next character''
(32)	=	(1/680)	=	Routine''Look for separator''
(33)	=	(1/683)	=	''Read integer''
(34)	=	(2/684)	=	''Read until newline''
(35)	=	(2/619)	=	Position of job description.
(36)	=	(1/621)	=	Next line.
(37)	=	(21/619)	=	Marker - no dump tape.
(38)	=	(1/616)	=	Routine to Abandon job.
(39)	=	(50/619)	=	No of free blocks.
(40)	=	(1/620)	=	Error Routine.

Connections.

Enter at (1) with statement begining STOR.

Exit a) To Translate next line.
b)
c) To abandon job and disengage reader
d) when no tapes.

Exit to error routine via any subroutine used.

Notes.

1.) STORE (separator)(m)(separator) BLOCKS.

Enters (m) in word 2.4 bits 12 - 23 of incomplete job entry.

Gives Error if m $\geq$ 220 (normally).

2) If no tapes on machine then care must be taken so that the machine does not become flooded with input well. And then is not sufficient space for execution of jobs. This is achieved by having active and passive sections of store. The size of the active store is the execution space of the largest job in the machine. This is increased if a larger job comes in and there is sufficient free space. If a job is too large then it is refused entry to the machine.

1607.

Purpose. Read statements beginning PARAMETERS.

No of Instructions. 28.

Cross References.

(30)	=	(2/619)	=	Position of job description.
(31)	=	(1/681)	=	Routine Get next n characters.
(32)	=	(2/681)	=	Routine Find beginning of title.
(34)	=	(1/620)	=	Error Routine.
(35)	=	(1/600)	=	Routine Get next character.
(36)	=	(1/621)	=	Routine Translate next line.

Connections with other routine.

Enter at 1) From Translation of Next Line (R621) when find line begining PARA.

Exit a)
b) To Error Routine.
c)

Exit d) On completion of Translation.

Notes.

1. Statements of form PARAMETER (separator)*[upto 8 octal digits](separator)

2. The parameters is stored in word 4.4 of description.

3. Give ERROR if a) Wrong format.
b) Too many octal digits.
c) Letters in middle of octal digits.

R608.

Purpose. Read statements beginning ''EXECUTION ''or ''COMPUTING'',i.e.

a) Execution , p. q HOURS / MINUTES / SECONDS.

b) COMPUTING '' '' '' ''

p . q is a fixed point decimal number.

Cross - References.

(30)	=	(1/681)	=	Routine Read next n character.''
(31)	=	(1/620)	=	Error Routine.
(32)	=	(1/600)	=	Get next character.
(33)	=	(2/684)	=	Routine ''Read until newline character.
(34)	=	0.4(1/619)	=	Position of job description.
(35)	=	(1/621)	=	Routine ''Translate next line''

Connections.

Enter at (1) with word statement begining EXEC or COMPUTIN.
Exit to Translate next line.
Exit to Error Routine.
Exit to Error Routine via any of the subroutines.

Notes.

1.) Execution (separator) p.q (separator) HOURS / MINUTES / SECONDS.
p.q is a fixed point decimal number.

This number is translated into seconds. And stored in half word 6.0 of incomplete job description.

Errors a) Incorrect format.

b) Letters in middle of integers.

2) COMPUTING (separator) p.q (separator) HOURS / MINUTES / SECONDS.

As (1) , except number is stored in halfword 6.4

1.9.64

R609.

Purpose. Re - entry to input master from peripheral routines. Routine is entered when ***, end of tape or end of tape buffer or peripheral fault, are shown on a peripheral.

No of Instructions.

Cross References.

(30)	= (2/697) =	Recognize a peripheral.
(31)	= 0 =	Value of M in R502 on entry which indicates length of input/output complete.
(32)	=0.6(1/620)=	Routine to deal with peripheral faults.
(33)	= 0.4 =	Value of M which indicates ***
(34)	= 1.4 =	Value of M which indicates end of tape.
(35)	= (5/201) =	Restart position.
(36)	= (66/599) =	V store address
(37)	= 16/619 =	Block label of current input block.
(38)	= (17/619) =	P.A.R. value of current input block.
(39)	= (1/318)	
(40)	= (1/312)	
(48)	= (1/622) =	Routine''free a peripheral''
(50)	= (50/599) =	Constant.
(51)	= (51/599) =	Constant.
(52)	= (20/619) =	Half word showing state of peripheral.
(53)	= (1/202) =	Program scan.
(54)	= (10/619) =	No of blocks in present document being input.
(56)	= 2(5/502) =	Routine to restart peripheral on reading ***?
(58)	= (14/619) =	Store for Position when the current document began in current input block(if did not begin in block then = 8.0.)
(59)	= (1/619) =	Jmarker ≠ 0 if J document being entered.
(60)	= (1/611) =	Routine to Process Compiler or Data Heading''
(61)	= (5/619) =	Get next character - position of next character.
(62)	= (6/619) =	'' - no of characters in record.
(63)	= (2/621) =	Routine ''Translate next line''
(64)	= (64/599) =	Constant.
(65)	= (1/610) =	Routine''To start peripheral reading next document''
(66)	= (2/206) =	Enter S.E.R. in queue.
(67)	= (20/502) =	Where to go if peripheral fails.
(68)	= (2/502) =	Routine ''Start reading from a peripheral.
(69)	=0.1(5/568)=	Marker to show used to read in binary.
(70)	= (21/502) =	Marker to show next to be read in non-parity checked binary.
(71)	= (1/627) =	Allocate to new buffer and start input disposer.
(72)	= (12/619) =	Copy of (20/619) showing how present input document should be read.
(73)	= (65/599) =	End of Input buffer.
(74)	= (21/502) =	b104 for non - parity check.
(75)	= (5/568) =	b103 '' '' '' ''
(76)	= 1(5/613)	

Connections.

Enter at (1) From R502 when something has happened on a peripheral with b100 = start of peripheral working space, and this working space is set up as shown in the notes.

Exit a) To Program Scan
b) To read tapes in binary
c) To translate the beginning of a document.

Notes.

There are eleven states of entry to this routine, these are listed and the action taken is shown below.

1. Peripheral fault. - This exits immediately from this routine.

2. End of tape - this means that end of tape has occurred without a terminating sequence: there are three types of action:-

a) if last terminator read was E read next tape in binary

b) " " " " "B,F then read next tape in internal code.

c) if last terminator read was A,C,T,Z, then expect next tape to begin with a terminator.

3. End of buffer. - gives a new buffer to the peripheral and then carries out any neccessary translation on the block just input. Activates Input Disposer if neccessary

4 ***A. - Abandon document. - abandons the current document, does not disengage the reader.

5 ***B. - Do not disengage reader, read in binary until end of tape, if end of tape, read next tape in internal code.

6 ***C - Do not disengage reader, - end of document. Read next document on same piece of tape.

7)***E - Do not disengage reader, read in binary get end of tape or an ending sequence.

8)***P - Do not disengage reader - read without parity check until reads an end of document terminator ie ***Z,A n C

9)***T - Disengage reader,do not end document if document began in this block, then carry out any translation needed.

10)***Z - Disengage reader end of document, if document began in this block then carry out any translation or storage of titles needed. Also do any end of document processing necessary.

b100 = address of head of storage space

(68/599)	= 50	=	V store address of peripheral (less * 6)
(66)	= 0.4	=	Reason why returning to Input marker Routine (see502)
(60)	= 1.0	=	Beginning of buffer (less * 7)
(61)	= 1.4	=	End of buffer +0.4.(less * 7).-normally will fill buffer and then set next half word zero.
(64)	= 2.0	=	Next character address in store in main store block.
(65)	= 2.4	=	End of store + 0.4 (less * 7) bits 23-12 = * 44667 bits 11-0. end address within block.
(62)	= 3.0	=	Address of buffer which peripheral is going to write to next . (less * 7)
(67)	= 3.4	=	Address in buffer which P.E.R. is going to read from next (less * 7)
(56)	= 4.0	=	Address in store reserved for next separator (less * 7) (0 if record not started)
(51)	= 4.4	=	Reason why peripheral has stopped.
(52)	= 5.0	=	Address of code conversion taken in use. (I.s. bits give shift,)
(50)	= 5.4	=	Last joins characters - each - *.
(53)	= 6.0	=	Count of parity faults since last entry to (1/502) Is.bit = 1 if reader to stop on parity faults = 0 otherwise.

1.9.64

8610

Purpose Entered when the peripheral is engaged by the operator

No of Instructions

Cross References

(30) = (5/201) = Re-entry point
(32) = (20/619) = Peripheral marker word
(33) = (14/619) = Where document begins in block
(34) = (10/619) = No of blocks in current document
(35) = (5/568) = Input table
(36) = (2/502) = Entry to start reading
(38) = (1/626) = Put input block onto input stream
(43) = (2/626) = Acquire another input block.
(45) = (6/999)
(64) = (64599) = Where next character from peripheral to be stored
(66) = (2/248) = Write one half word into block
(67) = (2/697) = Recognize peripheral
(68) = (1/609) = Return address.
(69) = (68/599) = V- store address.

Connections

Enter at (1) with b100 = private store of peripheral.

Exit to (1/502) to read next tape in binary or internal code into a new area of store.
Exit to (5/502) to read next tape as before into the same area of store.

(2/502) ??

Notes

1. On entry the routine first tests if the peripheral was disengaged by the operator, if so it continues to read in the same mode into the same area of store

2. The only ways that the machine will disengage the peripheral are :- A) Tape ends without terminating character
b) Tape ends with ***Z.
c) " " " ***T.
d) " " " ***F.

a) The machine normally expects the next tape to begin with *** unless the last main terminator read was F, B or E. E expects next tape to be read in binary, F and B use end of tape as end of document, thus the next tape will be read in internal code

b) Read in internal code into new area of store
c) Read in internal code same area of store
d) Read in binary into same area of store.

2611.

Purpose. Routine entered when end of a compiler or data document is found.

No of Instructions. 21.

Cross References.

(30) = (2/697) = Identify peripheral
(31) = (4/619) = Position of buffer block
(32) = (14/619) = Identifier of current input block
(33) = (10/619) = No of blocks in current document
(34) = (1/202) = Program Scan
(35) = (41/619) = Head of list''here not on tape''
(64) = (64/599) = Position of last character read
(36) = (5/201) = Re-entry address
(37) = (3/619) = Job marker
(38) = (2/619) = Position of current job marker.
(39) = (49/619) = Head of complete job list.
(40) = (1/626) = Put block onto input stream;
(41) = (4/613)

Connections.

Enter at (1) with b100 = peripheral private store.
Exit to program scan.

Notes.

1. This routine is entered when end of compiler or data document is reached. It puts the title of the document onto the'' here not on tape''list, and if the document is part of a job document,it updates the position of the end of this. Also ,if the complete job list is empty, it puts this document onto tape immediately without waiting for block to be filled.

1.9.64

R612.

Purpose. Start of day routine - Peripherals. At ''start of day'' Operator must feed in message saying ''Peripherals A,B,C, are working,rest are not working '- This operates this routine.

No of Instructions.

Cross References.

(30) = (1/630) = Routine - 'Acquire one block.'
(31) = = Storage for''number of peripheral''
(32) = (20/619) = Head of list of storage space per peripheral.
(33) = = '' '' '' '' V- Store addresses = :
(34) = = Routine '501' to set up private store.
(35) = = Program Scan.

Connections.

Enter 1) - From Operators Request saying.
'Peripherals A,B,C, Working, rest not working'
Exit. a - Program scan on completion of routine

Notes.

1) Operators request must set bit 18 of the peripheral table
= 1 if peripheral in use
= 0 if not in use

2) Operators request which says '' activate a new peripheral - must count peripherals engaged, and enter in 'no of peripherals engaged marker' and must set'active bit'(note 1) of this peripheral.

1.9.64

R613.

Purpose.

To deal with the end of a job document.

Cross References.

(30)	= (2/619)	=	Position of current description.
(31)	= 10	=	Maximum computing time for a short job in seconds.
(32)	= 200	=	Minimum execution time for a big job
(33)	= (1/619)	=	J marker.
(34)	= (1/202)	=	Program Scan
(35)	= (16/619)		
(36)	= (1/318)		
(37)	= (1/312)		
(38)	= (64/599)		
(5)	= (5/627)		

Connections.

Entered at (1) from R624 on finding a partial separator under such circumstances as to indicate the end of the job document.

Exit to program Scan.

Notes.

1. Forms the type of job currently being input the basic rules are that a job is 'short' if computing time < 10 secs.
a job is 'tape' if uses tape
a job is 'long' if execution time > 200 secs.

2) These types are tape - type 1
short - type 2
long - type 3

3. Initially type of job and priority are set equal.

1.9.64

R614.

Purpose. to form processed job description of any jobs that are completely in the machine.

No of instructions.- 110

Cross References.

(30) = (5/201) = Re-enty Address
(31) = (35/619) = Head of incomplete job list.
(32) = (21/619) = Marker word (sub.store)
(33) = (1/202) = Program Scan
(34) = (52/619) = Working space-holds position of current job document
(35) = (8/619) = Head of input stream 0
(36) = (2/630) = Find block
(37) = (3/203) = Block status directory.
(38) = (48/619) = Position where next entry is to be stored
(39) = (1/318) = Call to cores
(40) = (1/312) = Set P.A.Rs.
(41) = (2/626) = Acquire another input block
(42) = (1/694) = Give back space
(43) = (1/625) = Deal with slash documents
(44) = (3/626) = Put block onto input stream
(45) = (3/627) = Activate R699 - Input disposer.
(46) = (49/619) = Complete job list
(47) = 0.4(48/619)
(48) = 1(5/613)

Connections.

Enter at (1) from R628 via S.E.R. queue.
Exit a) to Program if no more complete jobs in machine.
Exit b) to R699 via S.E.R. queue if some blocks ready to go onto input tape.

Notes.

1. bit 2 of subsidiary store marker word = 1 if this routine is active, or there is an entry in the slow S.E.R queue for it.

2. Routine tests if there are any complete jobs waiting to have processed job description formed. Does this by testing bit 0 of word 4 of incomplete job entry.

3. Processed job description are stored exclusively in input stream 0 , begining in a new 64 block each time.

1.9.64

Processed Job description - each item linked in word 10.4

First section. words 0 - 12.0

words 0-9.4 - Title of job.

word 10.4 - Link to next title

word 11.4 bits 12-23 - Title identifies

bits 3-12 - No of titles in machine

Second section word 12.4 - 24.4

word 12.4-14.4 - A direct copy of job entry in main store.

12.4(0), bit 0 = 1 if unusual job.
0 otherwise

bit 1 = 1 job document entered on a remain device.

bits23-2 = link with next entry in complete job list in main store.

word 13.0.(0.4), bits 0-3 = type of output peripheral.

4-11 = amount of output in blocks.

12-23 = position of leading block of job on tape.

word 13.4(1.0). bit 0 = 1 if job on tape
0 otherwise, ie. no input tape.

bit 1

bit 2

bits 3-7 Input stream of job document.

bits 8-23 Position of processed job description on tape or in store

Processed job description.

Second section (continued.)

word 14(1.4) = bit 0 : = 1 if private tapes are needed
= 0 otherwise.

bit 1 : = 1 if archive tapes are needed
= 0 otherwise.

bit 2 : = 1 if archive tapes are used (used by spare alloc.)
= 0 otherwise.

bit 3 : = 1 if long job.
= 0 otherwise.

bits 4-6 : priority of job (high, low medium)

bits 7-11 : Numbers of tape units needed at beginning of job.

bits 15-17: Type (short, long, tape.)

bits 18-23: Compiler

word 14.4(2.0)

bit 0 = 2 On Active list B.
= 0 otherwise.

bit 1 : = 1 On active list A.
= 0 otherwise.

bit 2 : = 1 if swing overfrom.
= 0 otherwise.

bits 3-10 : Archive tape identifier.

bits 11-23: Job Identifier.

Processed job description.

word 15 (2.4) :	bit 0-7	=	no of input sections needed in processed job description.
	bits 8-11	=	no of archive documents called for.
	bits 12-23	=	no of blocks called for.
word 15.4 (3.0):	bits 0-23	=	Time in seconds.
word 16 (3.4)	bits 0-7	=	no of tape sections needed in processed job description.
	bits 8-11		no of documents not yet entered (= 0 when put on complete job list)
	bits 12-15	=	no of self document.
	bits 16-19	=	Decks.
	bits 20-22.		unused.
	bit 23	=	Tape marker.
word 16.4(4.0)	bit 0	= 1	when all document to this job are on the input tape , and it is waiting to have the processed job description formed.
	bits 23-1	=	position of title of job.
word 17.0(4.4)		=	Parameter.
word 17.4(5.0)		=	Execution time.
word 18.0(5.4)		=	Computing time.
words 18.4(6.0) to 22.0 (9.4)		=	Outputs.
word 22.4(10.0)		=	Link to block if there are more than 7 output streams. (=0 if otherwise)
word 23.0.(10.4)		=	Link to next entry in processed job description.
word 23.4 (11.0).	bit 0	= 1	when job document has been put onto the input tape.
		= 0	otherwise.
	bit 8-10	=	position in block
	bits 11-23.		position on tape.
word 24.0 (11.4.			link with rest of jobs in incomplete list.

R615.

Purpose. To translate statements beginning *

No of Instructions.

Cross References.

(30):	=	=	Get next character.
(31)	=	=	Hartran marker
(32)	=	=	Hartran compiler recognition in bits 18.23.
(33)	=	=	Position of job description.
(34)	=	=	Alternative entry to End of job document.
(35)	=	=	Read integer.
(36)	=	=	Get next on characters.
(37)	=	=	Error Routine.
(38)	=	=	Binary switch, sometimes.
(44)	=	(1/502) =	Entry to R502
(45)	=	(1/613) =	Routine to enter complete job list.
(46)	=	=	Program Scan.
(47)	=	=	
(48)	=	=	Job marker.
(49)	=	=	Routine'find first non-separator.
(50)	=	6/621 =	Alternative entry to Next List.

R616. -

Purpose - Implement ***A.

No of Instructions.

Cross References.

(30) = = Beginning marker.
(31) = = Position of Peripheral Rivets Store.
(32) = = Program Scan
(33) = = Routine to free peripheral x.
(34) = = J marker
(35) = = Operators Routine.
(36) = = Buffer block which holds title of compiler and Data documents during input of the documents.
(37) = = Routine to give back block of store.
(38) = = Position of job description.
(39) = = Head of 'not here,assigned' list -(10.4)

Connections-

Enter at (1) from R621 upon finding a line beginning Comp. with b100 = peripheral private store.

Enter at (3) from R618 upon finding a simple Data statement.

Exit.(a) to R608. on finding a line beginning computation.

Exit (b) to read rest of document.

Enter at (5) from assembly routine with b100 = beginning of document.(ie position of 1st internal separator)

Exit to. with b100 = position of 12 word block allocated to this document.

1.9.64

R617.

Purpose. Read statements beginning ''Compiler''

No of Instructions.

Cross References.

(30) = (1/681) = Routine ''Read next n characters into B102'' when B105 = n.
(31) = (1/692) = Routine ''ask for 12 word block''.
(32) = (4/619) = Store used, shows position of buffer 12 word block.
(33) = (2/608) = Routine to translate line beginning ''computation''
(34) = (36/619) = Head of list of compilers.
(35) = (2/681) = Routine find beginning of Title or compiler name.
(36) = (2/688) = Routine ''form checksum of compiler name''
(37) = (1/684) = Routine ''search list of half words''
(38) = (1/619) = Job marker ≠ 0 if reading job document.
(39) = (1/682) = ''Routine'' Read and store title''.
(40) = (1/621) = Alternative entry to R624 to find end of document.
(41) = (2/619) = Position of job description now being entered on this peripheral.
(42) = (3/619) = Used to show if present document has been defuried by a self statement.
(43) = (5/619) = Position of last character read
(44) = (6/619) = No of characters in present separator.
(45) = (7/619) = Position sub store private store
(46) = (13/619) = Position of last separator read.
(47) = (20/619) = Position of peripheral marker word
(48) = (1/627) = Put current block onto stream and get new block and contains reader.
(49) = (1/611) = Routine to deal with end of document.
(50) = (1/682) = Read integer.

Connections

Entered at (1) from R621 on reading COMP.
Entered at (5) from R621 on reading USE
Entered at (3) from R618 on reading data.
Exit to R611 if document begins and end in same block
Exit to R608 if heading is not compiler
Exit to R202 (Program Scan) if document opens more than one block.
Exit to R627 to put input block into input stream if new input block needed.

Notes.

1. If compiler or data document then forms the following block of information.

words.	0.9.4	=	=	Title
word	10.0.	= bits 20-23	=	type of entry =*04 compiler *1 Data.
		bits 14-18	=	Input peripheral stream.
		bits 4-13	=	compiler identifier.
		bits 0-3	=	no of input document.
Word	10.4	= bits 2-22	=	Link to next document
		bit 1	=	1 if last document in processed job description (set in R614)
		bit 0	=	
Word	11.0	= bits 12-22	=	Position of document in store if not on tape (block label.)
		or		
		bits 11-23.	=	Position of document on input tape (block label).
		bits 8-11	=	Position of beginning of document in block
		bits 0-7	=	Archive tape no if document on archive tape.
Word	11.4	= bits 11-23	=	Position of leading block of document on tape, or in store.
		bits 10	=	1 if document on archive tape 0 otherwise
		bit 9	=	1 if document on input tape 0 otherwise.
		bits 0-8	=	No of blocks in document

2) Data / documents.
All data/ documents with the same title are read in and the documents are treated as a normal simple document.

Data / List.

Words	0-9.4	= Title of document
word	10.0	= No of elements needed. ie. total no of documents with title shown in 0-9.4 This is initially to * 2 and is reset to (n) on reading Data / (n) END.
Word	10.4.	Link with data / list
word	11.0.	No of elements entered. (regards document as fully in when word 11.0= word 10.0.
Word	11.4	Link with rest of continuation document.

Document 12 word block - as data document.

R618.

Purpose. Translate Lines beginning 'DATA'

No of Instructions.

Cross References.

(30)	=	(2/681)	=	Routine Find beginning of title.
(31)	=	(1/692)	=	Routine ask for 12 word block
(32)	=	(4/619)	=	Position of buffer block b list
(33)	=	(3/617)	=	Alternative entry to 'compiler'routine
(34)	=	(1/683)	=	Routine 'Read integer into b 102.
(35)	=	(1/619)	=	Job marker.
(36)	=	(1/685)	=	Routine '' Duplicate this block in buffer b.
(37)	=	(1/600)	=	Get next character
(38)	=	(1/681)	=	Routine ''Read next on characters''.
(39)	=	(1/620)	=	Error Routine.
(40)	=	(3/619)	=	End marker.
(41)	=	(1/627)	=	Read main program.
(43)	=	(1/682)	=	Routine ''Read on store title.''
(44)	=	(13/619)	=	Separator at beginning of current record

Connections.

Entered	at	(1)	from R621 on reading data
Exit	To	(3/617)	if normal data document
Exit		(1/627)	if Data / document
or	to	(1/620)	Error routine.

R619.

Purpose. - Constants and parameters associated with each peripheral.

Associated with each peripheral.

(1) = = Job marker
≠ 0 if job document is being input.
= 1 if input section.
= 2 if output section.
= 3 if tape section.

(2) = = Position of job description currently being input.

(3) = = Self marker - used if there is a ''self'' statement within a job document. - also there is always an inplied ''self''statement in the common job description

(4) = = Position of buffer block used in Routines R617 and 618. when inputting compiler or Data document.

(5) = = Re-start position- position of last character read

(6) = = Re-start 2 - no of characters in record.

(7) = = Restart 3 = (b100- substore add for present peripheral).

(8) = = Head of input stream

(9) = = Remote - holds type of peripheral that this input device has been made remote to.

(10) = = No of blocks in present document being input on this device.

(11) = = Last four characters input in previous input block (used when making sure that peripheral has stopped on *** and not ccc)

(12) = = Store for (20) in 609.

(13) = = Store word for position of the separator at the beginning of current line (also word in R600.)

(14) = = Position where current document began in current input block. if did not begin in current block - 1.0

(15) = = V - store address per peripheral.

(16) = = Current input block label. (Re - start

(17) = = Block label used by input routines (*36 - - 3603 AT moment)

Associated with each input device. (cont.)

(20): bit 0 = 1 if end of statements reached - set by R621., word by R600.

bit 1 = 1 if this an end document - used in Data / n documents

bit 2 = Integer switch - used in R604

bit 3 = beginning marker, = 1 if document begins in current input block

bit 4 = 1 if peripheral engaged.

bit 5 = 1 if operators input

bit 6 = 1 if next tape to be read in non - parity checked binary.

bit 7 = 1 if new input block needed in, previous document ended in last 64 word block, so entire input block was put onto tape - a new input buffer is needed before anything can be done with this device.

bit 8 Used by R600- 1 if next re-entry begins at the start of block, ie no re-entry points 1or2 set

bit 9 = 1 if Hartan being input. (R615)

bit 10 = 1 if last directive was A.

bit 11 = 1 " " " " " B

bit 12 = 1 " " " " " C

bit 13 = 1 " " " " " E

bit 14 = 1 " " " " " F

bit 15 = 1 " " " " " P

bit 16 = 1 " " " " " T

bit 17 = 1 " " " " " Z

bit 18 = 1 if previous tape ended without a terminator.

bit 19 = 1 if peripheral disabled.

bit 20 = 1 if peripheral needs reengaging when buffer space has been re-allocated (operators request has said - disable on input device- must re-allocate buffer space,).

bit 21 = 1 if next tape must begin with ***, (only set if bit 18 is also set.

(20): bit 22 = 1 if next tape is to be read in parity directed binary.

bit 23 = 1 if *** read in previous record. (used by R600)

General words - not particular to each device.

(30)	=	=	word to be stored in register 10.0 of 12 word block
(31)	=	=	" " " " " 11.0 " " " "
(32)	=	=	" " " " " 11.4 " " " "
(33)	=	=	Head of ''here - not assigned'' list
(34)	=	=	Head of 'assigned - not here' list.
(35)	=	=	Head of incomplete job list.
(36)	=	=	Head of list of compiler.
(37)	=	=	No of blocks allowed to be called for in machine. - normally - if there is a dump tape, this will be 220, however if there is no dump tape, then this will depend on the other job at present in the machine. (see R607 ''store'').
(38)	=	=	Store word used in R624.
(39)	=	=	Store word showing position of block that has just been put onto magnetic tape by input disposal (R699) used in R628.
(40)	=	=	Store word used in R603.
(41)	=	=	Head of ''here not on input'' list.
(42)	=	=	Current peripheral being processed (holds working space).
(43)	=	=	Position of next separator working space
(44)	=	=	Position of last character read for R600
(45)	=	=	Store word used in R628
(46)	=	=	Head of data / list.
(47)	=	=	No of peripherals.
(48)	=	=	Store word used in R614
(49)	=(12/652)=484*3504=		Head of compiler job list.
(50)	= 375*7	=	No of free blocks
(51)	=	=	Where last processed job des. ended.
(52)	=	=	Store word 614.
(53)	=	=	Head of list of titles
(54)	=	=	store in 625
(55)	=	=	Head of operators request - OPS request (b100 in word 0.0.4 = beginning of request.)
(56)	=	=	Beginning of current record
(57)	=	=	No of titles in machine.
(58)	=	=	Position of first block of title on dump tape.

Sub store half - word (21/619). in sub store

bit	0	=	1	No dump tape
bit	1	=	1	Input disposal active
bit	2	=		Peter warn active
bit	3	=		R641 active
bit	4	=		R614 active

R621.

Purpose. - Translate the next line of input.

No of Instructions.

Cross References.

(30) = = Position of beginning of Record.
(31) = = No of characters in Title.
(32) = = Store word for b100.
(33) = = Restart store word 1.
(34) = = " " " 2
(35) = = " " " 3
(36) = = Routine find beginning of Title
(37) = = Routine Error.
(38) = = List of peripherals
(39) = = Routine to enter S.E.R. in Queues.
(40) = = Operators Input Routine.
(41) = = Program scan.
(42) = = Read next n characters.
(43) = = Get next character.
(44) = = Routine - Search table of half words.

Connections.

Enter at 1) - having set positions of last character read and no of characters in record before entry.

Enter at 5) - normal entry to routine to read the beginning of next record.

Exit. a) if line beginning with unrecognizable character. [ie,(,),£,£,?,£,/.]

Exit b) if operators input - via S.E.R. queue.

Exit c) To routine to deal with heading

Notes.

1. Routine reads characters until finds first non-separator if integer,*, XorJ then Takes applicable action. Otherwise reads in next B characters and jumps to routine signified by there characters.

2. Contains a Table , beginning at (4) of all possible characters that can read line.

3622.

Purpose.

To disengage peripheral.

No of Instructions.

Cross References.

(30) = (20/619) = Peripheral marker word
(31) = = Private store of this peripheral.
(32) = = Free this peripheral.

Connections.

Enter at (1) to disengage peripheral.
with b109 = peripheral identifies.

R623.

Purpose. Translate lines beginning '' SELF'',''Input'',''Output''

No of Instructions.

Cross References.

(30) = (1/619) = J marker = 1 if input switch
= 2 if output.
(31) = (1/620) = Routine ''Errors''
(32) = (2/681) = Routine ''find next non - separators''
(33) = (1/600) = Get next character.
(34) = (1/683) = Routine '' Read integer''
(35) = 0.4(1/619) = List - position of job document
at present being read in.
(36) = (1/621) = Routine ''Next Line.''
(37) = (1/680) = Routine ''Test for separator''

Connections.

Enter at (1) statements beginning SELF.
exits to Routine ''Translate next line'' R621.

Enter at (2) statements beginning INPUT.
exit as (1).

Enter at (3) statement beginning OUTPUT
exit as (1)

Notes.

1) Expects statement SELF = (n)
gives error if incorrect one word
stores (n) in word 3.4 ,bits 12 to 15

2) Input sets Job marker = 1.0 (1/619)
this indicates that input section is being read in.
3) Output sets Job marker = 2.0.

R624.

Purpose.

Entered when a peripheral becomes free or a program is completed. It tests if there are any complete jobs. And if so, forms the processed job description and puts this into the input stream ready to go on tape.

No of Instructions.

Cross References.

(30)	= (1/627)	=	Get another input block
(31)	= (43/619)	=	Pos. in next separator
(32)	= (1/613)	=	End of document
(33)	= (2/626)	=	Get another input block
(34)	= (20/619)	=	Peripheral marker word
(35)	= (1/318)	=	Call to cores
(36)	= (1/312)	=	Set PAR
(37)	= (3/502)	=	Read from peripheral
(38)	= (38/619)	=	Store word
(39)	= (56/619)	=	Beginning of current record
(40)	= (1/609)	=	Re-entry after reading
(41)	= 0.4(56/619)		
(42)	= (2/600)		

Connections.

Enter at (1) When n peripheral becomes free, or a job is completely in the machine.

Exit. to program scan, when there are no more complete jobs to be dealt with

Notes.

This routine only chooses which job to deal with, and sets the relevant markers, it uses as subroutines R614 and R625. R614 forms the processed job description , and R625 gets the necessary space, in an input stream, in which to form this job description if no peripheral is free, Item R625 is the routine which abandons everything.

R625.

Purpose. Allocate store for periphal
This is a subroutine used by R614, it looks for a free peripheral and then takes enough store in this peripherals stream list as is needed by R614.

No of Instructions.

Cross References.

(30)	=		= No of peripherals.
(31)	=	(20/619)	= Peripheral descriptive word.
(32)	=		= Alternative entry to R624 if no peripherals are disengaged.
(33)	=		= Position of private store of peripheral.
(34)	=		= Store word for position of processed job description in block
(35)	=	(3/626)	= Put present input block into stream list, but mark as ''being processed''
(36)	=	(2/626)	= Acquire another input block.
(37)	=		= Re-entry to R614
(64)	=	(64/599)	= Peripheral private store, beginning of buffer.
(38)	=		= Put present input block onto tape, but mark as ''waiting to go onto tape''

Connections.

Enter	at (1)	with b103 =	no of consecutive blocks of 64 word blocks called for.
Exit	(a)	to R624	if no peripherals are free
	(b)	to R614	when sufficient space has been found.

Notes.

This routine gets enough 64 word blocks of space for R614 to put a processed job description into. Each entry block of store needed is put onto The input stream, bit s marked as ''being processed'',This prevents input disposal trying to put it onto magnetic tape.

R626.

Purpose. Input block full

- 1) to put the present input block onto the stream list, mark as waiting to go on tape
- 2) to acquire a new input block.

3) To put present input block onto stream list, mark as ''being processed.(used by R625)

Number of Instructions.

Cross References.

(30) = (2/697) = Recognize Peripheral.
(31) = (2/630) = Find block.
(32) = (8/619) = Reads of peripheral streams.
(33) = (3/203) = Status directory
(34) = (1/630) = Aquire one block
(35) = (1/318) = Call to cores
(36) = (1/312) = Update PAR's
(37) = (3/627)
(38) = (5/613)
(64) = (64/599)
(39) = (5/201)
(40) = (1/999)
(41) = (2/999)
(42) = (3/999)
(43) = (15/229)
(44) = (1/213)

Connections.

Enter at (1) To put current input block on peripheral x at the head of the input stream marked as ''waiting to go on tape'' enter with b100 = peripheral x working space
b110 = link.

Exit at (a) with b100 = peripheral x working space.
b109 = peripheral x identifier
b101 = status entry for head block.

Enter at (2) To acquire a new input block for peripheral x
b100 = x working space
b110 = link.

Exit at b) with b100 = peripheral x
b105 = block label.
b109 = status entry
-block label in bits 22-12.

Enter at (3) to put current input block on peripheral x at the head of the input stream, marking it as ''being processed''.
b100 = peripheral private store.
b110 = link.

Notes.

This routine is normally entered at (1) or (2): it is entered at (3) when a processed job description is overflowing into more than one page. In this case, the block must not be put onto tape until it has been filled by the job description.

R627.

Purpose. Buffer block full
Entered at end of buffer block from input routine. Puts present buffer onto input stream list, updates necessary constants and get a new buffer block.

No of Instuctions.

Cross References.

(30)	=	(64/599)	=	position in private store
(31)	=	(2/697)	=	Recognize peripheral.
(32)	=	(11/619)	=	Store word,(per peripheral) to hold the last 4 characters read in previous block (used to check for ***)
(33)	=	(1/626)	=	Put input block onto input stream, mark as waiting.
(34)	=	(2/626)	=	Acquire another input block.
(35)	=	(16/619)	=	No of blocks in document currently being input.
(36)	=	(1/318)	=	Peripheral information word
(37)	=	(1/312)	=	1 bit = 1 if input disposal active.
(38)	=	(0.1)	=	Mask for (37).
(39)	=	(1/699)	=	Input disposal.
(40)	=	(2/206)	=	Put S.E.R. in queue.
(41)	=	(1/609)	=	Re-entry point to R609
(42)	=	(3/502)	=	Start reading from a peripheral. (alternative entry)

Connections.

Enter at (1) from R609 with b100 = peripheral private store.

Exit to start reading from peripheral (3/502) - May activate input disposal

IR623.

Purpose. Entered from input disposal when a block has been put onto tape.

No of Instructions.

Cross References.

(30)	= (41/619)	=	Head of ''here,not on tape'' list.
(31)	= (39/619)	=	Position where block has been put on tape.
(32)	= (35/619)	=	Incomplete job list head.
(33)	= (1/699)	=	Entry points to input disposal.
(34)	= (34/619)	=	Head of ''not here-assigned''list.
(35)	= (42/619)	=	Store for b101 when searching.
(36)	= (1/687)	=	Test if title is element of a list.
(37)	= (1/694)	=	Give back free space.
(38)	= (33/619)	=	Head of ''here not assigned''list.
(39)	= (46/619)	=	Head of data /
(40)	= (46/620)		
(41)	= (1/692)		
(42)	= 2(1/685)		

Connections.

Enter at (1) with b100 = identifier of block just put on tape
and (31) = block label on tape where block has been recorded.

Exit to Input disposal when all information connected with this block has been processed.

Notes.

1) Routine resets all leading block markers from correct block identifier to tape address.

2) Routine resets all relevant incomplete job entries to correct tape addresses, and puts each job on the complete job list if nessary.

3) Routine resets all relevant block addresses of document entered, and puts them on the correct list.

ATLAS SCHEDULER

12th FEBRUARY, 1964.

SCHEDULER ROUTINES

R650	Enter job in execute list
R651	Job assembly complete action
R652	Scheduler parameters and tables
R653	Set scheduling bits
R654	Selection routine
R655	End of job routine
R656	Set up stream directory
R657	Set new priority
R659	Activate scheduler (fixed store routine)
R660	Scheduler entry and exit
R661	Active list selection routine
R662	Enter job in active list
R663	Semi-emergency entry to scheduler
R664	Select assembly type documents
R665	Job assembly routine
R666 (R66, R766 R866 and R996)	Re-entry routine
R667	Shift n places
R668	Set up job size
R670	Select document for assembly
R671	Find input well block
R672	Remove / duplicate routine
R673	Assembly control routine
R675	Document complete action
R676	Prepare to read in
R677	Return from read in
R678	List job description
R980	Compiler assembly
R982	Find / free halted program
R983	Tape assembly for execution jobs
R984	Tape assembly for active list jobs
R985	Obtain a free deck

19th October, 1964.

INDEX

SCHEDULER NOTES

1. The scheduler routines occupy four blocks (*3423, *3437, *3440 and *3441) and perform five main tasks.
   a. choose jobs for assembly and enter them in the active list.
   b. choose jobs for execution and set up execution directories with the appropriate time sharing bits set.
   c. assemble input documents and compiler in main store.
   d. control the assembly of magnetic tapes.
   e. allocate external priorities.

2. To ensure only one branch of the scheduler is active at a time entry to it can only be made through R659.

3. Reentry addresses are most important in the scheduler routines and reentry routines (R66, R666, R766 and R866) which recover B100- B110 after non-equivalence are provided in each block. Where possible modifications to a routine should go in the same block as the routine.

4. R652 contains all scheduler parameter and label settings.

5. To recompile feed in R652 followed by the four scheduler tapes.

6. There are groups of instructions at present in the scheduler routines which may be incorporated at some future date but play no active part at present.

12.2.64

JOB DESCRIPTION ENTRIES

A programmer's job description is processed by the input master routine into $12\frac{1}{2}$ word entries and then expanded, when the job is assembled for execution, into $15\frac{1}{2}$ word entries.

1st ENTRY

Words	0-9.4	=	job title
	10.0	=	spare
	10.4	=	link to next entry
	11.0	=	spare
	11.4(23-12)	=	title identifier
	(11-3)	=	after information
	12.0	=	spare
	12.4(23-2)	=	active entry address
	(0)	=	if in job description assembly
	13	=	job entry address
	13.4-15.0	=	spare

2nd ENTRY

Words	0-2.0	=	copy of job entry
	2.4-15	=	further information not used by schedule

These first two entries are standard, other entries are

JOB DOCUMENT ENTRY

Words	0-9.4	=		document title
	10 (23-20)	=		entry type
	(19-14)	=		input stream type
	(13-4)	=		compiler identifier (0 if data, 511 if compiler not listed)
	(3-0)	=		number of input document
Word	10.0(23)	=	1	if this entry is last one in block (this bit is only set in the $12\frac{1}{2}$ word entry)
	(22-2)	=		link to next entry
	(1)	=		spare
	(0)	=	1	if assembly type document (set at assembly time)

12.2.64

JOB DESCRIPTION ENTRIES CONTD.

word	11.0	(23-12)	=		block label of last document block
	or	(23-11)	=		tape block number of last document block if the document is on magnetic tape
		(10-8)	=		position in block where document begins
		(7-0)	=		archive tape number (or if not on tape, 0.2 if on system input tape)
word	11.4	(23-12)	=		block label of leading document block
	or	(23-11)	=		tape block number of leading document block if the document is on magnetic tape.
		(10-9)	=		spare
		(8-0)	=		number of document blocks
word	12.0	(23-12)	=		ax+b space in blocks required by job when it is being compiled
		(11-3)	=		count of number of blocks assembled
		(2)	=	1	if document refused space after being partially assembled
		(1)	=	1	if document being assembled
		(0)	=	1	if document in main store
word	12.4		=		spare
word	13.0	(23-2)	=		job description address
		(1)	=		spare
		(0)	=	1	if next block to be assembled isn't in store
word	13.4	(23-12)	=		block label of next block to be assembled
	or	(23-11)	=		next block on tape to be assembled
		(10-8)	=		position in block where document begins
		(7-0)	=		archive tape number
word	14.0		=		spare
word	14.4	(23)	=	1	if space refused during assembly
		(22-12)	=		block label of last block assembled
		(10-0)	=		block label of last but one block assembled
word	15.0		=		spare

EXECUTION DIRECTORIES

STATUS (½word entry)

bit	(23)	= 1	if halted for a block
	(22-12)	=	reason for halt
	(11)	= 1	if tapes used
	(10)	= 1	if clearway job
	(9)	= 0/1,	full recover / only recover if in supervisor
	(8-2)	=	link to next job
	(1)	= 1	if in supervisor
	(0)	= 1	if free to run

STORE (½ word entry)

bits	(23-13)	=	number of blocks reserved less one
	(12-2)	=	start of job area in block directory relative to start of block directory
	(1)	= 1	if not in processing mode
	(0)	= 0/1	clear new blocks / do not clear new blocks

SWITCH (½word entry)

bit	(23)	= 1	if branching
	(22-20)	=	location of dump in dump block
	(19)	= 1	if compiling
	(18)	=	spare
	(17-15)	=	job type
	(14)	= 1	if dynamic time shaving
	(13)	= 1	if own clock
	(12-7)	=	instruction counter for scheduling
	(6-4)	=	job priority
	(3-2)	=	spare
	(1-0)	=	monitor description

MONITOR (½word entry)

bit	(23)	= 1	if in offline trap
	(22)	= 1	if page lockdown
	(21-12)	=	off line monitor description
	(11-12)	=	page number if lockdown trap
	(1)	= 0/1	master process / no master process
	(0)	= 1	if waiting off line trap

12.2.64

COMPILER DIRECTORY ENTRIES

```
A1      (11/291 - ½ word entry)
  bits  (23-0)  =   first four characters of compiler name

A2      (12/291)- ½ word entry)
  bits  (23-0)  =   second four characters of compiler name

B       (16/290 - ½ word enrty)
  bits  (23-19) =       a
        (18-11) =       b
        (10-3)  =       compiler size
        (2)     =   1   if common compiler
        (1)     =   1   if protected
        (0)     =   1   if keep in store

C       (13/291) - ½ word entry)
  bit   (23) =      space
        (22-13) =   compiler number
        (12-0)  =   position on library tape

D       (20/291) - ½ word entry)
  bits  (22-13) =   space
        (12-0)  =   position on dump tape

E       (14/290  - ½ word entry)
  bits  (23-2)  =   address in store
        (1)     = 1 if assemble
        (0)     =    space
```

NOTE

Working space allocated to a compiler is AX + b
where X = number of blocks on document and

A = 2 to the power - a if a = 0,1,2 or 3 and
A = 2 to the power 24-a if a = 23-12

12.2.64

PERIPHERAL ENTRIES

PERIPHERAL SCHEDULER ENTRY (½ word entry)

bit	(23)	= 1	if remote
	(22)	= 0/1,	if normal / semi-remote
	(20)	= 1	if tape out of action
	(19-15)	=	print well backlog
	(14-13)	=	spare
	(12)	= 1	if full backlog
	(11-3)	=	actual output backlog
	(2)	= 1	if idle
	(1)	= 1	if semi-emergency
	(0)	= 1	if emergency

STREAM DIRECTORY (½ word entry - entry n for stream n)

bit	(23)	=	spare
	(22-15)	=	stream full parameter
	(14-11)	=	space
	(10-3)	=	stream output backlog
	(2)	= 1	if peripheral available
	(1)	= 1	if semi- emergency
	(0)	= 1	if emergency

PERIPHERAL TABLE (½ word entry)

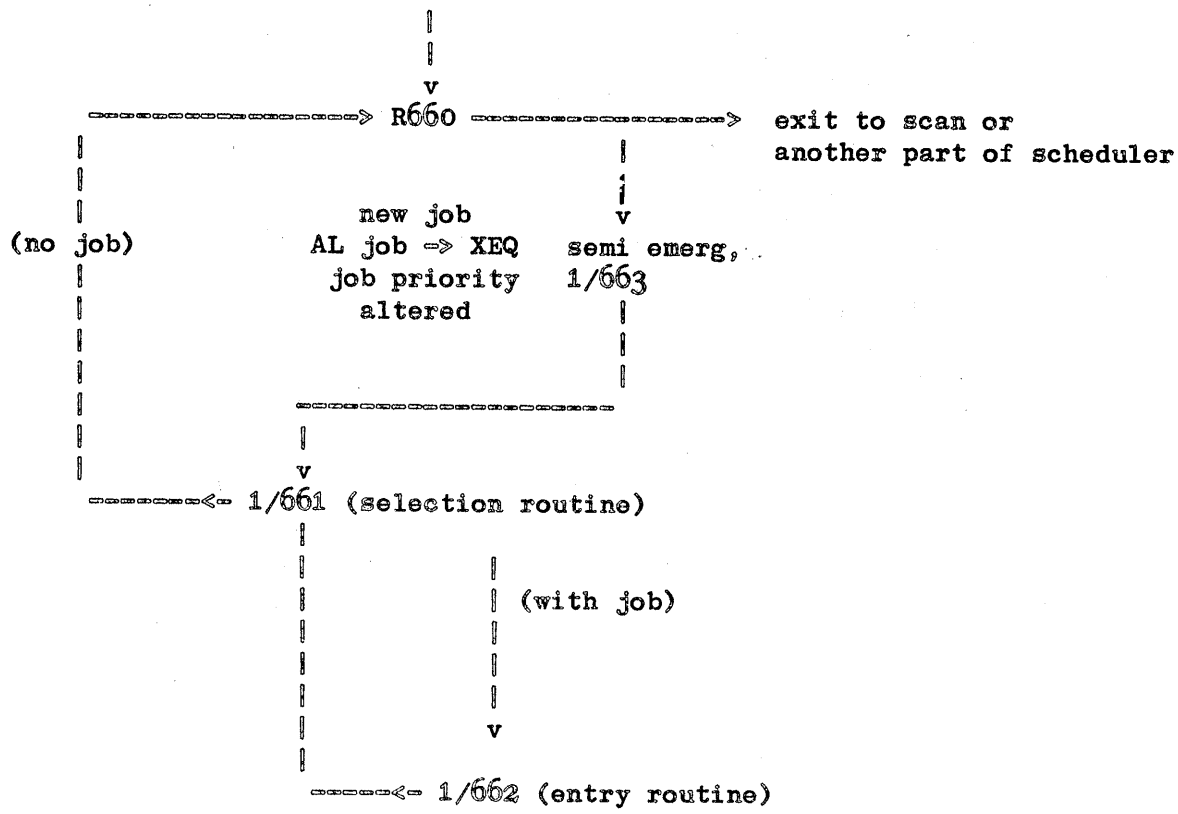
bits	(23-2)	=	address private peripheral store (less*7)
	(1)	= 0/1,	if input / output device
	(0)	= 1	if not available

STREAM TYPE TABLE (½ word entry - entry n for output peripheral n)

bits	(22-8)	=	spare
	(7-3)	=	stream type

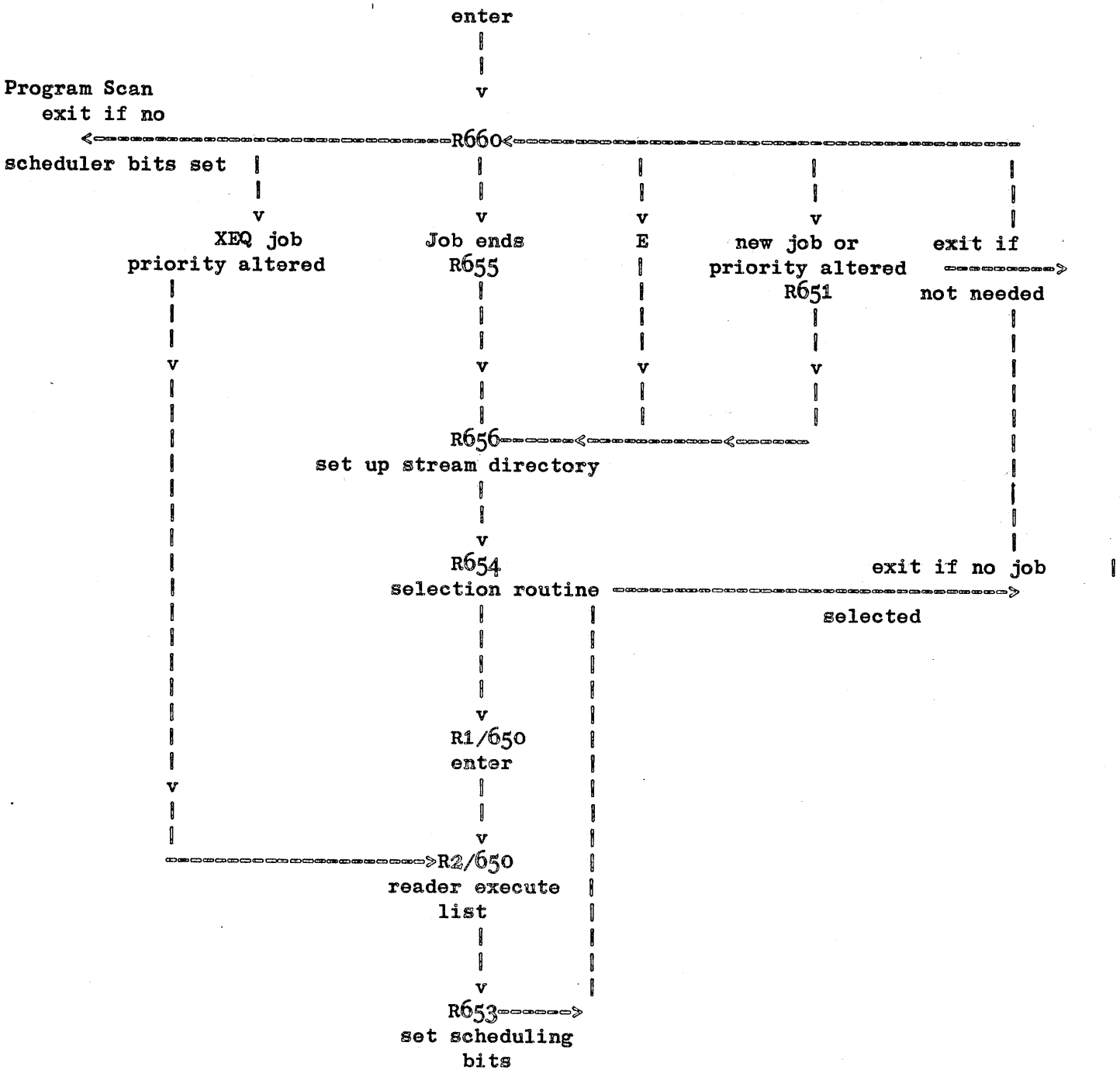
12.2.64

Active List Scheduler. (106)



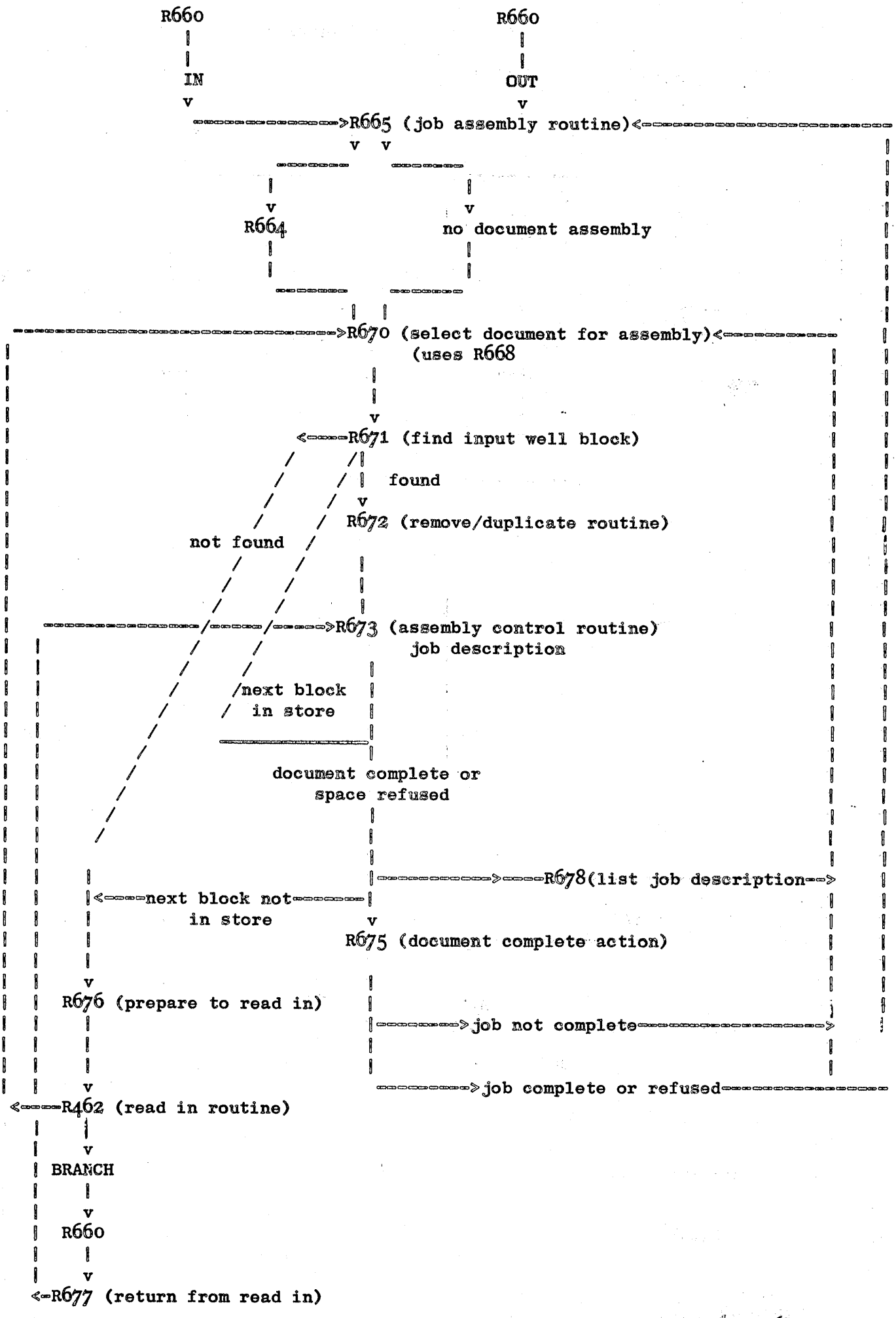
12.2.64

Execute Scheduler. (250)



12.2.64

Document Assembly Flow Chart.



12.2.64

R650. Enter job in execute list.

Purpose: To enter a new job in it's correct position in the execute list.

Connections with other routines.

a) Entered at (1) from R654 with
B100 (1,0) = 1.
(11) = 1/0, tapes/no tapes.
other digits zero.
B101 (19) = 1 compile.
(17-15) = type.
(6-4) = priority.
other digits zero.
B102 job description address.
B103 (23-12) number of blocks required less one.
B104 active list entry address.
Exits to 1/660 if there are no free entries otherwise to 1/653.

b) Entered at (2) from R657 with
B101 (6-4) = priority.
B106 = entry to be ordered.
Exits to B110.

Also the new priority must have already been set in the switch directory and the entry removed from the list linking with the next job bits set to zero.

Notes: 1. The routine places the new job or the one with altered priority immediately above the leading job of the next priority rating.

2. A free entry in the execute list is detected by the fact that the store directory entry is zero.

3. The routine sets up new entry as follows:

a) Status directory.
(11) = 0/1 1 tapes used.
(8-2) = link to next job.
(1-0) = 1.

b) Switch directory.
(19) = 0/1 1 compiler mode.
(17-15)= type.
(6-4) = priority.
These bits must be added in to the entry.

c) Main program store (mps) dump.
mps 0 PJD address.
mps 1 (23-12) max (ax+b) -1.
mps 4 active list address.
mps 5 SER entry address (1/280)

4. The execute list contains 8 entry positions but the first entry is always used by the supervisor leaving 7 for main programs.

5. There are 3 basic types of jobs indicated by the following type values: 1 for tape jobs, 2 for common or peripheral jobs, 3 for computer jobs.

6. Priority values are as follows
0 top job, 1 tape job, 2 common job, 3 computer job, 4 low job.

12.2.64

R651. Job assembly complete action.

Purpose: To determine when a new job is cleared on an active list stream whether the selection routine should be entered.

Connections with other routines:
Entered at (1) from R660 and exits either to 1/656 or 1/660.

Notes:

1. The aim of this routine is to prevent the selection routine being entered when a new peripheral job which is cleared on the active list, is not required on the execution list. By this means interruption of computer-tape job time sharing does not occur every time a new peripheral job appears.

12.2.64

R652. Scheduler parameters and tables.

Purpose: To set up scheduler parameters and tables and also to set any unset tables during development period.

Notes:

12.2.64

R653. Set scheduling bits.

Purpose: To set own clock and dynamic time sharing (DTS) bits where necessary.

Connections with other routines:
Entered at (1) from R650 & R660, and exits to 1/654.

Notes:

1. The rules for setting these bits are:

   a) If top priority job present give own clock.
   b) If no top priority job present, give all tape jobs and top peripheral job own clock and D.T.S. If no peripheral job present, give own clock and D.T.S. to one computer job which is using tapes.
   c) Jobs not mentioned above are given a general clock and no D.T.S. occurs.

2. If there is only one or no jobs for D.T.S then *4 is set in time criteria to prevent D.T.S., otherwise the time plus scheduler period is set as the time criteria.

3. Associated with each job is a ''clearway'' bit (digit 10 status directory) which has the effect of ensuring that execution of the job will continue so long as it is not held up. If the bit is not set a job is likely to be interrupted as soon as a job of higher priority becomes free. At present this bit is not set for any jobs.

12.2.64

R654. Selection Routine.

Purpose: To select a job for the execute list from those available on the active list.

Connections with other routines:
Entered at (i) from R656 and exits to 1/660 if no job selected and to 1/650 with the entry conditions set up for this routine if job selected for entry to execute list.

Notes:

1. The rules of selection are:
   a) Only active list jobs whose documents and compilers are in main store are considered for selection.
   b) All except peripheral jobs are entered in execution list as soon as available.
   c) Peripheral jobs are entered when execution list is O.K. if
      i) Stream output well is not full.
      ii) No peripheral job on execute list and either
      iii) emergency
      or
      iv) space available and output fits into well.
   d) If execute list now O.K. then a peripheral job is entered providing the output well is not full.

2. The space parameter used for scheduling is the job size plus max (ax+b) ie maximum compiling space and not the size estimate given in the job description which is allocated to the job after compiling is complete.

12.2.64

R655. End of Job Routine.

Purpose: To lose the active list, job and job document entries when a job finishes execution.

Connections with other routines:

Entered at (1) from 1/660 and exits to 1/656.

Subroutines:

a) Enter 1/684 to lose $2\frac{1}{2}$ word entry with
B104 = entry address.
B110 = link.
Uses B100, 104, 107 and B108.

b) Enter 6/694 to lose 62 word entry with same entry conditions and usage as a).

Notes:

1. It is assumed that job and active entries are in the same stream.

12.2.64

R656. Set up stream directory.

Purpose: To update the stream directory.

Connections with other routines:

Enter at (1) from R653, R655 and R660 and exit to 1/654.
Also entered at (1) from R663 and exits to 3(1/663)

Notes:

1. Associated with the active stream list is a stream directory, each stream having a $\frac{1}{2}$ word entry. At present only the entries for peripheral streams are used. Each entry consists of:

(0)	=	0/1	/emergency.
(1)	=	0/1	/semi-emergency.
(2)	=	0/1	/not full.
(3-10)	=	Output backlog blocks.	

Bits 0, 1 and 2 are set when any of the peripherals in this stream category are in a state of emergency, semi-emergency or not full respectively. Similarly the stream output backlog is the sum of the backlogs of all the peripherals in the group.

2. For each entry in the peripheral scheduler table there is a corresponding $\frac{1}{2}$ word entry in the stream type table

(2-7) = stream type.

12.2.64

R657. Set new priority.

Purpose: To allocate an operator's priority request to a job.

Connections: Entered at (1/657) from R660 with
B100 (23-11) = job identity.
(2-0) = priority 0 top, 0.1 high,
0.2 normal, 0.3 low.
Exits to (1/660) after having put an SER to the slow queue to return to operator routine with B100 as on input plus
B100 (3) = 1 if job on JL
(4) = 1 " AL
(5) = 1 " execute list, which will call in R651, R653&R661, and also after setting 2.5 in the scheduler line.

Subroutines: R650 to reorder execute list.

Notes:

1. The effect of the priorities are:

a) job list entries.
the new priority is entered in the job entry.
top jobs are put into the top stream, and
high jobs are put to head of this stream queue.

b) active list jobs.
top jobs are put into the top stream.

c) execute list jobs.
the new priority is noted in the switch directory and the list reordered.

2. Note that with high and normal jobs the priority equals the type.

R659. Activate Scheduler.

Purpose: To set marker bit in scheduler line and enter in SER to the slow queue if scheduler not already active.

Connections with other routines:

Entered at (1/659) with
B109 = marker bit
B110 = link.

Exit to link.

Subroutines:

a) R206 entered via. R230 to put SER to queue
Entry at (2/206) with

B107 = queue number.
B108 = entry information
B109 = entry address.
B110 = link.

Notes:

This routine provides the only method to be used for activation of the scheduler. The routine in combination with R660 ensures that the scheduler is only active in obeying instructions and waiting for a drum transfer on one task at a time.

12.2.64

R660. Scheduler Entry and Exit.

Purpose: To examine the scheduler line in the working store and exit to appropriate routine.

Connections with other routines:

Entered at 1/660 from SER queue entry or from scheduler routines.

Exit via jump table to appropriate scheduler routine with B100 unaltered and active bit set, and to program scan if scheduler line zero.

Notes:

1. Entry to scheduler routine is made through R660 using the scheduler line to indicate the routines required to be entered and a jump table giving their addresses. They are dealt with in priority order 0, 1, 2 ..... 22. Digit 23 is set to 1 when the scheduler is active and no entry can be made whilst this bit is set.

12.2.64

R661. Active List Selection Routine.

Purpose: To select a job from the job list for the active list.

Connections with other routines:

Entered at (1) from R660 and exits to 1/660 if no job selected or to 1/662 with
B100 (6-2) = stream.
B101 (23-2)= job entry.

Notes:

1. Rules of selection (in order of priority).

a) There is a limit to the number of jobs that can be put on the active list, when this limit is reached no further jobs are added till a position becomes vacant.

b) Only one job undergoing private tape assembly is allowed on the list at a time.

c) Only one unusual job is allowed on active list at a time.

d) Top jobs and jobs with ''put on AL'' bit set are entered to the list.

e) Only 1 computer and tape stream job is allowed on list.

f) Peripheral jobs in a stream with semi-emergency are entered if no other jobs of that stream are already on list.

g) There is a minimum number of jobs required on list to ensure an O.K. condition. If this number is not reached after first pass of the selection routine a second pass is made.

h) Jobs which are selected on the second pass are peripheral jobs whose output will fit in the well and also there must be a peripheral job on the execute list (this indicates that the output wells are in a state of being filled).

2. The order of peripheral streams is critical, the order should be one of importance and popular usage, viz, the ''any'' stream should be first with the least used peripheral stream last.

R662. Enter Job in Active List.

Purpose: To enter a job in the active list.

Connections with other routines:

Entered at (1) from R661 with
B100 (6-2) = stream address.
B101 = job entry.
and exits to 1/661. The routine sets the assembly routine bit in the scheduler line.

Subroutines:

a) Enter R692 at (2/692) to obtain $2\frac{1}{2}$ word entry
with B110 = link
Exits to link with B104 = new entry address.
This subroutine may destroy all B lines except B100.

Notes:

1. Obtains free entry and sets link to this entry and link to job entry, also sets an active list bit in job entry and arranges for assembly.

12.2.64

R663. Semi emergency entry to scheduler.

Purpose: To set up the stream directory according to latest peripheral information before selecting jobs for active list.

Connections:

Entered at (1) from R660 and exits to (1/661)
R656 is used as a subroutine.

Notes:

1. Semi-emergency on a peripheral indicates that jobs for this stream should be assembled in preparation for entry to the execute list when emergency occurs.

2. The stream directory is only set up by R656 and this routine must be used before entering the active scheduler proper.

12.2.64

R664. Select assembly type documents.

Purpose: To search through the document entries for a job and mark those which require assembly before execution.

Connections with other routines:

Entered at (1) from R665 & R678 with
B100 (23-2) = AL entry
Exits to (1/670)

Notes:

1. The documents which are assembled in store before execution are compiler, data, archive, private tape and slash data document 1.

12.2.64

R665. Job Assembly Routine.

Purpose: At initiate the assembly of job documents on the active list.

Connections with other routines:

Entered from R660 & R678.
and exits to
a) (1/660) if no jobs for assembly, otherwise to
b) (1/670) with
B100 (23-2) = active entry.
(0-1) = 0

Subroutines:

a) R668, entered at (1/668)

Notes:

1. The rules of selection are clear from the flow diagram, however is should be emphasised that though jobs are not normally assembled in store before their tapes have been mounted archive documents of active list jobs are assembled as soon as the relevant archive tape mounted.

2. The ''no archive tape'' bit is set when trying to assemble archive documents whose tape is not mounted. This bit is cleared on all AL jobs when any archive tape is mounted.

12.2.64

R666. Re-entry Routine.

Purpose: To preserve B lines 100 - 110 when non-equivalance occurs.

Connections with other routines:

Entered at (1) from a routine by the instructions

```
113   126   0   0.4(3/666)
121   126   0     (1/666)
```

Entered at (2) after non-equivalance.

Notes:

1. This routine must be stored with each block of scheduler routines.

2. When only B100 needs to be preserved it is best to set the re-entry address direct into 5/201.

12.2.64

R667. Shift n places.

Purpose: To shift the bits in B105 n places to the right.

Connections with other routines:

Enter at -n (2/667) with
B105 = word to be shifted.
B110 = link.

12.2.64

R668. Set up Job Size.

Purpose: To determine the number of blocks a job occupies whilst in the active list and set this number in the active list entry.

Connections with other routines:

Entered at (1) from R665 with
B102 (23-2) = PJD address.
B110 (23-2) = link.
(1-0) = 0
Exits to link with
B110 (0) = 1 if job size exceeds free space.

Notes: 1. There are three space parameters associated with a job, these are

a) the actual space the job occupies whilst sitting in store. This space is used when selecting jobs for the active list. It is obtained by summing the various document sizes, which are themselves obtained either on entry to the store through normal peripherals or when they are read from an archive tape at assembly time.

b) compiler working space ax+b, where a & b are compiler parameters and x is the size of the document. This parameter is used when selecting jobs for the execute list. When there are a number of documents the max ax+b is used.

c) The programmers own space estimate. This is the space allocated to a job when it is ready to commence execution after compilation.

12.2.64

R670. Select document for assembly.

Purpose: To select the appropriate job document for assembly.

Connections: Entered at (1) from R665 with
B100 (23-2) = Active list entry, and
at (8) from R462 with
B100 (23-2) = doc address.

Exit with
B100 (23-2) = document address
to (1/676) if document uses special or dump tape
(1/671) otherwise, or
to (1/665) if no document selected.

Subroutine: R693 to obtain 62 word section. Enter 1/693 with
B110 = link , bit 0 = 1. Returns to link with
B104 = address of section. May use B101-110.

Notes:

1. If private tapes are still needed only archive documents may be assembled. These are assembled for AL jobs as soon as archive tape is mounted. Normally documents are not assembled in store until all the tapes have been mounted.

2. To incorporate job description assembly into the normal document assembly framework a 62 word section is obtained and set up as though it was really the job description, ie. words 12.4 & 13.0 of the first section contain the active list and job list entry address and in the second section word 13.0 contains address of the 1st. section (or temporary PJD) and the block label of the real PJD will be placed in 11.4 when it is assembled. Later this section will be used as the first section of the expanded PJD.

R671. Find input well block.

Purpose: To search the input stream to find block for assembly.

Connections with other routines:

Entry at (1) from R670 or R673 with
B100 (23-2) = document address.

Exit with
B100 preserved
to (1/676) if block not found and
to (1/672) with
B100 (23-2) = document address.
B102 (10-8) = section document begins.
B103 if (0) = 1, (7-2) = input stream
if (0) = 0, (22-12) = BL of previous block in stream.
B106 = BSD entry.
B108 (22-12 = BL
B109 (2-0) = priority, 0 = supervisor 0.1 = top.
0.5 = normal.

Subroutines:

a) R667
b) R630 entered at 2/630 with
B108 (22-12) = block label.
B110 = link.

Return with
B105 (12-2) = block position.
B106 = BSD entry.
B108 (22-12) = block label.
Uses B105, 106 and 108

R672. Remove/Duplicate Routine.

Purpose: To either remove or duplicate a block in the input well.

Connections with other routines:

Entry at (1) from R671 with
B100 (23-2) = document address.
B102 (10-8) = section document begins.
B103 if (0) = 1, (7-2) = input stream.
(0) = 0, (22-12)= BL of previous entry.
B106 = B.S.D. entry.
B108 (22-12)= B.L.
B109 (2-0) = priority 0 = supervisor 0.1 = top
0.5 = normal.

Exit to (1/673) with
B100 as on entry.

Subroutines:

a) R630 to find position 11 in BD of BL
Enter at 2/630 with
B108 (22-12) = a
B110 = link.
Exit with
B105 (12-2) = b
B106 = BSD entry.
B108 (22-12) = a

b) R630 to acquire free block.
Enter at 1/630 with
B109 (2-0) = reason & priority.
B110 (23-2) = link.
(0) = 1
Exit with
B105 (12-2) = a if available.
B105 = 0 if not available.
B109 (22-12) = BL
remainder 1's.

c) R315 to duplicate block.
Enter at 1/315 with
B100 (12-2) = a rest zero, (21) = 1.
B101 (23-3) = link, (2-0) = 0.6
B102 (12-2) = BL old block.
(22, 21, 13,0)= 1, rest zero.
note: 22=1 sets lock down.
B103 = (14/314)
Exit with B100 preserved.

d) R248 to read or write isolated word
Enter at (2/248) with
B102 = word to be written.
or B102' = word to be read.
B104 = address within block.
B109 (22-12)= block label.
(8-2) = program number.
digits 9,0 irrelevant, remainder zero.
B110 = return address.
if (2) = 1 read, = 0 write
Uses B105 - 110, Bt.

Exit with B100 or B100-104 preserved according to re-entry address.

12.2.64

R672 Cont.

Notes:

1. The block status directory entry (BSD) is
   (23) = 1 if break in stream.
   (22-12) = next BL
   (11-3) = tape block number.
   (2) = 1 if being processed.
   Bits 0 & 1 used by input master & should not be altered.

R673. Assembly control routines.

Purpose: To link and count blocks, unlock any where necessary, set reject bits and exit to appropriate routine.

Connections with other routines:

Entered at (1) from R672
with B100 (23-2) = doc address.

Exits to 1/671, 1/676, 1/678 & 1/675 with
B100 as on entry.

Subroutines:

a) R630 entered at 2/630.
b) R462 entered at 2/462 to activate tape.

12.2.64

R675. Document Complete Action.

Purpose: This routine is entered when the assembly of a document is complete to determine future action.

Connections with other routines.

Entered at (1) from R673 with
B100 (23-2) = document address
Exit to (1/670) and (1/462) with
B100 (23-2) = AL entry.

Subroutines

a) R630 entered at 2/630.

b) R678 entered at 3(12/678) to lore blocks and exit to R660 with
B101 (22-12) = BL of first block to be last
(other blocks are linked in BSD)
B103 = link (R660)

Notes:

1. When all the documents of a job are in store the blocks are linked in the BSD with (22-12) = next BL, ans (23) = 1, if last block of document.

12.2.64

R676. Prepare to read in.

Purpose: To set up the entry conditions for the routine which reads a block into store.

Connections with other routines:

Entered at 1/676 with
B100 (23-2) = document address.

Exits to read in routine with
B100 as on input.
B101 (23) = forward/backward linking.
(7-3) = tape deck number.
(2-0) = priority.
B103 = tape block number.
B110 = (8/670) immediate link.
B102 = 1(1/677) return link after obtaining details of block just read in.

Subroutines: at R667, shift in places.

Notes:

1. The input and archive tapes are assumed to have backward linking whilst dump and satellite tapes have forward linking.

12.2.64

R677. Return from Read In.

Purpose: To set up the entry conditions for the assembly control routine after reading in a block from tape.

Connections with other routines:

Enter at 1/677 from read in routine
with B108 (23-2) = document address.
B101 (12-0) = tape number.
B102 (22-12)= block label.
(23) = /block refused.

Exit to 1/673.
with B100 as an input
and words 13.4 and 14.4 of document entry set up.

Notes:

1. The procedure with reading tape blocks is that (1/462) is entered to read in tape block. This routine puts an entry in the tape queue and returns to link. When a block has been read in a bit is set in the scheduler line and R677 is entered via R660. At this stage no further blocks are assembled from tape till the tape routine is activated either when a document is complete by entry to (2/462) or when another block has to be read in. Details of the block just read in cannot be passed via R660 and these are obtained by entry to (10/462) At this point the block is locked down and the entry is lost from the tape queue.

R678. List job Description.

Purpose: To copy the original processed job description from $12\frac{1}{2}$ word entries to $15\frac{1}{2}$ word entries and lose original entries from store.

Connections with other routines:

Entered at (1) with
B100 = job description address.
Exit to R660
Entered at 3(12) to free blocks with
B101 (22-12) = block label
B103 = link
and any further blocks to be lost must be linked in the BSD.

Subroutines:

a) R692 entered at (1/692) to obtain 62 word section.
b) R248 entered at (2/248) to read or write isolated word.
c) R205 entered at -3(23/205) to update free block counter with
B110 = link.
Uses B105 - 110.
d) R317 entered at 1(3/317) to lose block with
B101 (22-12) = block label
B102 = program number (0)
B100 = link
Exits with B105 = BD position.

Notes:

1. At the beginning of R678 the links in the document entries must be reset to take account of any blocks which have been duplicated.

12.2.64

R680.

Purpose. - Two subroutines a) Test if separator present.
(comma or space or newline)

b) Test if separator present.
(comma, space, full stop or newline)

No of Instructions.

Cross References.

(30) = (1/600) = Get next character.
(31) = (1/621) = Error Routine.

Connections with other routines.

Entry 1a) - To test for separator class 'a'(comma, space, newline)
b110 = link.

Entry 2a) - To test for separator class 'b'(comma, space, newline, full stop)
b110 = link.

Exit. a) To link. if space, tab, newline.
b) To link if comma.
c) Error if character $\geq$ 10. not full stop or comma.
d) To link if full stop.

Notes.

1. Test next character - gives error if $\geq$ 10 and not full stop or comma.

- Ignores shifts.

Subroutines used.

2. Get next character.

1.9.64

m681.

Purpose. Subroutines a) read next n characters.
Entry with b105 = n
b110 = link.

b) Find beginning of next word or integer.
ie. Find next non-separator.

No of Instructions.

Cross - References.

(30) = (1/600) = Get next character.

Connections With Other Routines.

Entry 1) To read next n characters.
b105 = n
b110 = link

Entry 2) To find beginning of next word or integer.

Exit a) When having read next n characters in b102

Exit b) With b101 = first character of next word

Notes.

1.) Reads next n characters , keeping last character read,
if n $\geq$ 4 loses first x characters

2.) Assumes word beginning with character $\geq$ 10, not
comma or full stop.

1.9.64

R683.

Purpose. - Read Integer

No of Instructions. 15

Cross - References.

(30) = (1/600) = Get next character
(31) = (1/620) = Error Routine.

Connections with other Routines.

Enter 1) if beginning of integer not yet formed.
with b110 = link - b102 = 0

Enter 2) if beginning of integer formed with b101 = first character
b102 = 0
b110 = link

Exit a) if reads letter or character $\geq$ 32.

b) to link with integer in b102

Notes.

1) Assumes integer ends with a separator of class (a)

m634.

Purpose. - Amount of Subroutines

a) Search a list of half words for an element
enter with b 102 = element.
b 105 = begining of list.

on exit b 104 = identifier.
Error if not on list.

b) Read until new line.
Ignores all characters up to newline.

c) Update words 10.0, 160,11.4 of
document entries.

Cross References.

(30) = (1/620) = Error routine.
(31) = 0.4(5/600) = Exit link in ''Get next character.''
(32) = 0.4(3/600) = ''Get next character''no of characters in current statement.
(33) = 0.4(2/600) = ''Get next character''position of last character read.
(34) = (4/600) = Alternative entry to ''Get next character''.
(35) = 0.4(3/619) = Stored for word 11.0 of 12 word block.
(36) = (4/619) = Store for word 11.4 of 12 word block.
(37) = (3/619) = Store for word 10.0 of 12 word block.

Connections With Other Routines.

Entry 1a) - FOR routine to search a list of altemate half words
b102 = element searching for.
b110 = link.
b105 = begining of list.

Exit a) If element not in list.
b) when element joined with b104 = Identifier.

Entry 2) - Read until newline.

Exit a) To link when newline read.

d)

Entry 3) To store halfwords into 12 word block
positions 10.0 ; 11.0 ; 11.4.
b104 = position of 12 word block.

Exit e) When completed.

Notes.

1) Assumes lists of the form.

a0 / a,
b0 / b,
c0 / c,.

where x0 = element.
x, = identifies.

List terminated by zero entry.

R685.

Purpose. - Subroutines = to duplicate a title into a buffer block.
= to add 1 to no of tape decks needed.

No of Instructions.

Cross Refences.

(30) = = Position of job description.
(31) = (1/620) = Error routine.
(32) = *00002000 = Maximum no of tape decks allowed x 2^8

Connections.

Entry 1) - To duplicate a job title.

b104 = 12 word block to be stored in.
b110 = link.
b109 = peripheral identifier.

Exit a) -To link when transfer completed.

Entry 2) - To add one to no of tape decks needed.
b110 = link.
b109 = peripheral identifier.

Exit b) - Error , - too many decks requested.

c) - to link on completion.

Notes.

1). Transfers complete 10 words.

R686.

Purpose. Read (a) /(b) /(c)

No of Instructions.

Cross References.

(30) = (1/620) = Error routine.
(31) = (2/683) = Read integer
(32) = (31/619) = Word to be stored in 11.0
(33) = (30/619) = Word to be stored in 10.0
(34) = 2(2/681) = Find next non separator.
(35) = (1/683) = Read integer.

Connections.

Enter at (1) with b101 = first character of (a)
b110 = link.

Exits to Error Routines
or link.

Notes.

1) accepts format :- (A)(separator?)b/(separator?)(b)(separator?)/(separator?)(c)
(a) can be of form :- A(integer)
or (integer)

2) Stores in word to be stored in 11.0 in following names.

bits 0-7 = integer part of (a)

bits 8-10 = (c)

bits 11-23 = (b)

word to be stored in 10.0 = identifier = *14 if (a) = A (integer)
= * 7 if (a) = (integer)

3) Gives Error if 1) (b) >5000
2) (c) >7
3) incorrect format.

R687.

Purpose. - Subroutine - Test if title is element of a list.

No of Instructions. 12.

Entries.

1) To search the list.

b 104 = position of beginning of title.
b 105 = " " " " list.
b 107 = (" " " " ")- 10.4
b 110 = link.

Exit a) if title is part of the list b105 = position of title in list.
->link. b104 unchanged.
b107 = position of previous element in list.

Exit b -> link +16.0, if title is not part of the list.

b104 unchanged.

Notes.

Search assunes that table ends with zero link.

1.9.64

R688.

Purpose. - To form checksums

a) of title in 12 word block

b) of title not yet put into 12 word block.

No of Instructions.

Cross References.

(30) = (1/692) = Routine'' Get a 12 word block''
(31) = (1/682) = Routine'' Read and store title''
(32) = (1/694) = Routine'' Give back 12 word block of space.

Connections.

Enter 1) To form checksum of title in 12 Word block
on entry b104 = position of title.
b110 = link.

on exit b102 = checksum.

uses b107, 102, 106, 108,

Enter 2) To form checksum of title not yet in 12 word block.

on entry = b 110 = link.
b 101 = first character of title.
on exit = b 102 = checksum.

uses. = all b - registers.

Notes.

1) Entry (1) assumes that title has been stored in a 12 word block and forms a halfword checksum of 6 bit characters until it encounters a blank halfword.

2) Entry (2) Acquires a 12 word block, stores the title in it, forms a checksum (using first entry of routine) and then gives back space.

3) Routine assumes that all titles have $\leq$ 80 characters and end with a newline.

R689.

Purpose. - To record the details of a new output stream.

No of Instructions.

Cross References.

(30) = (32/619) = Word to be stored in 11.4
(31) = (2/619) = Position of job description.
(32) = (1/620) = Error Routine.
(33) = (1/692) = Routine '' Get 12 word block''

(34) = (30/619)
(35) = 5(2/604)

Connections.

Enter at (1) to record the details of a new output stream with :-
b 102 = (m) = no of blocks of output.
b 104 = type of peripheral.
b 110 = link.
b 109 = peripheral number.
(30/619) = no of output document.

Exit a) to link.

b) to error routine if too many output document defined (>16)

Notes.

1) Routine allows maximum of 16 documents. (physical maximum is 31)

2) Routine notes the major output document in job description.

3) Information is stored in one half word.

bits 0 - 3 = type of output peripheral.
bits 4 -17 = amount of output in blocks.
bits 18 -23 = no of output document.

4) The first 7 output documents are entered in half words 6.4 to word of job description, and word 6.0, bits 2 - 7, contains the number of output streams.

R690.

Purpose. - To enter document or title in incomplete job list.

No of Instructions.

Cross References.

(30) = 0.4(1/619) = position of job description.

Connections.

Enter 1) to enter document in incomplete list.

enter with b110 = link

b104 = position of 12 word block holding the title.

exit to link.

uses b 101,108.

8691.

Purpose. To request another archive tape.

No of Instructions.

Cross References.

(30) = (2/619) = Position of job description.
(31) = (1/620) = Error Routine.
(32) = (31/619) = Word to be stored in 11.0.

Connections.

Enter at (1) with b 110 = link.
no of archive tape in working register 0.4(3/619)
b 109 = peripheral identifier. bits 0-7

Exit (a) to link on completion of task.

Exit (b) to Error Routine if requested too many tape decks ($\geq$ 9)

Notes.

1. Adds 1 to no of systems decks required in job desciption word 2.4 bits 8 - 11.

2. Resets the number of the major archive tape needed in job entry word 1.4 bits 12 - 17.
Assumes that last archive tape requested to the major one.

3. Sets word 1.4, bit 0, = 1 ; This signifies that archive tapes are needed.

4. Gives Error, if request too many tape decks.

R692.

Purpose. - a) Ask for 12 word block of space - Entry (1)
b) " " $2\frac{1}{2}$ " " " " - Entry (2)

No of Instructions. - 24.

Cross References.

(30) = = Pseudo -Label for position of head of '512' list of blocks. (position of head of list - 1.4).
(31) = = Routine - ask for '62' word block of space.

b registers used. 101, 108, 104,

Entries and Exits.

Entry 1). with link in b110 to acquire 1 '12' word block.

Exit a) To routine - get one 62 Word section.

Exit c)

Exit b) To link with position of block in b 104

Entry 2) With link in b 110 to acquire one '$2\frac{1}{2}$' section.

Notes.

1) Input master space is divided into 4 groups.

a) 512 word blocks

b) 62 word blocks

c) 12 word blocks

d) $2\frac{1}{2}$ word blocks

Each 512 word block is divided into 62 word sections, beginning at word 16.0. also the first words are :-

word 0.0. head of free '62' word blocks.
0.4 head of 62 word blocks used for 12 word entries.
1.0 " " " " " " " " $2\frac{1}{2}$ " "
1.4 link with next 512 word block.
2.0. no of used 62 word blocks.,

2) On asking for a 12 or $2\frac{1}{2}$ word block the routine searches through the first 212 word section for an appropriate sized block.

2) (cont) and if there is a free 12 or $2\frac{1}{2}$ word block. it takes it, if not then moves on to next 62 word block, or if none are left then the next 512 word block. if cannot find space then ask for another '62' word block, to get this if necessary, a new 512 word block is taken.

3) Whenever a block is freed we attempt to free the 512 word block holding it. In this way the no of 512 blocks used is kept to a minimum.

4) Each 62 word block is divided into either '12' or $2\frac{1}{2}$ word sections begining at word 2.0. also

word	0.0	head of free section.
	0.4	no of used blocks.
	1.0	link with next section in this '512' section of this type.

R693.

Purpose. - Subroutine used by R692. and Peter Jones ask for '62' word block.

No of Instructions. - 52.

Cross References.

(30) = = Pseudo - entry for head of '512' list.
(35) = 1.0(1/692)= Alternative entry to R692.
(35) = (1/630) = Routine - acquire one free block.

Connections.

Entry 1) For '62' word block, divided into 12 word sections.
Link in 110.
10 odd if from Peter Jones.

7) For '62' word block, divided into $2\frac{1}{2}$ word sections.

Exit with position of 62 word section in b 104.

R694.

Purpose. - Give back '12' or $2\frac{1}{2}$ word block of space.

No of Instructions. 38.

Cross References.

(30) = = Routine - lose block b.

Connections.

Enter 1). with position of block in b104
link in b110
Exit a) - when completed to link.

Enter at 6) from Peter Jones. - to lose 62 word block.
b104 = position of block.

Notes.

- see R.692.

Working space. Uses b 100, 104, 107, 108.

R695.

Purpose. To find length of output in BLOCKS / LINES.

No of Instructions. 16.

Cross References.

(30)	=	(2/681)	=	Find next non separator.
(31)	=	(1/620)	=	Error Routine.
(32)	=	(31/619)	=	Word to be stored in 11.0
(33)	=	(1/600)	=	Get next character.

Connections.

Enter at (1) with b 102 = m
b 110 = link.

Exit to Error routine
or link.

Notes.

1) Forms amount of output in blocks in b 102 , bit 0 - 23 and in word to be stored in 11.0.

2) Only considers the first letter in each word ie. B or L.

3) Error if word does not begin B or L.

3696.

Purpose. = Recognize an output peripheral name.

No of Instructions. 17.

Cross References.

(30)	= (1/620)	=	Error Routine.
(31)	= (1/600)	=	Get next character.
(32)	= (2/683)	=	Read integer
(33)	= (1/695)	=	Blocks or lines
(34)	= (1/689)		
(35)	= (6/604)		

Connections.

Enter at (1) with b 102 = first 3 characters of output peripheral name

b 110 = link.

Exit to Error Routine if does not recognize peripheral.

or to link with b104 = peripheral identifier.

3 registers used. - 107, 108.

Notes.

1. There are four types of output peripheral.

any	=	0.1
Interprinter	=	0.2
creed 3000	=	0.3
teletype	=	0.4.

2. All output peripheral name must consist of one word only, ie routine searches for next separator as end of name.

R697.

Purpose. Find next non separator.

No of Instructions. 9.

Cross References.

(30) = (1/600) = Get next character routine.

Connections.

Enter at (1) with link in b 110.

Exit to link with character in b 101.

Notes.

1) Defines separator as - all internal characters < 10, full stop and comma.

Enter at (2) to Identifier a peripheral.

R700 Program monitor interrupts and SER

Purpose: To record in B91 any errors resulting in interrupts (EO,DO, Sacred Violation, Unassigned function) and any errors detected in S.E.R. when the program is in store control. Main program controls are forced so that on exit from supervisor control is transferred in extracode to R701.

Registers of fixed store: 29

Number of instructions obeyed: Varies from 9 to 22
A maximum of 14 orders obeyed with interrupts inhibited.

Parameters used: (1) to (14)

(7) = "Current program trap address"
One half word in subsidiary store in the current program working area holding
Digits 23 - 2 Address of trap vector. If no trapping, P23 = 1
Digit 0 0 if no errors detected
1 if error recorded by R700 (rest by R701)

Cross references

(3) = 2727*4 Table *4/*2 *1/*04 etc.
(4) = (4/201) "Enter Supervisor" slow queue
(5) = (31/500) Table n1, n2, n4 ..... n128 in alternate half words
(6) = (10/201) In Supervisor switch
(9) = (6/201) Main program controls
(10)= (2/701) Entry to "On line monitor extracode and trap"
(11)= (1/202) Entry to "Program Scan"

Connections with other routines

Entry at (1) from R500 under interrupt control with
B123 = 0 (UF)
1 (DO)
2 (SVI)
3 (SVO)
5 (EO)

Exit to 2048*4 if previous error has been recorded
to "Enter Supervisor" at 4/201 if first error with
B112 = (12/700) = Re-entry address.

Re-entry at (12) from R201

Entry at (2) from SER detecting error in current program in control of store, under extracode control, with
B126 odd (digit 0 = 1)
B100 Error marker digits 10-2 Digit 1 = 1 if non trap error,
or in one of digits 23-11 0 if trap error
B102 Return address R digits 23-3

Exit to Return Address R with error recorded in B91 and with B101 altered.

Entry at (14) : As entry at (2) only B102 not set on entry, and exit is to program scan with B101, 102 altered. Note again B126 odd on entry.

16.7.62

Temporary working space: Entry (1) : B111, 112
Entry (2) : B111, 112, 101
Entry (14): B111, 112, 101, 102
Entry (12): B101, 102

Notes:

1. This routine builds up in B91 a record of detected faults. The faults are of two types, those which can arise simultaneously and those of which only one can occur at once. The former (including interrupt faults) are recorded as digits in 22-11 of B91 (1 = fault, 0 = no fault). The latter are recorded as a counter in digits 10-2, with digit 1 = 1 if error is not trappable, 0 if error can be trapped. The assignation of digits and counters is to be determined later (see note 4).

2. The program is set up so that on exit from supervisor, R701 will be entered in extracode control at (2) to decode B91.

3. In order to prevent alteration of extracode control by a non-equivalence SER after it is set by R700 (during an SER), the non-equivalence look at me is put cut on entry (1) if the In Supervisor switch is negative.

4. The temporary assignment of digits in the record in B91 is

Digit	Fault	Digit	Fault
22	UF	17	Local timer
20	DO	21	Overall timer
18	SVI	19	Execution timer
16	SVO	15	Page lock down
12	EO		

R 701 On line monitor extracode and trap

Purpose: Entered after R700 has recorded faults, or from extracode routines discovering faults. Analyses whether the program has trapped the error or errors, and if so enters the trap. Otherwise enters R702 to monitor the error(s). An alternative entry is from the trap extracode.

Registers of Fixed Store: 22

Number of Instructions Obeyed: Varies from 2 to 30

Parameters used: (1) to (11)

(6) = Fault number of "No trap set" digits 10-2. Digit 1 = 1 Remainder zero.
(10) = Mask of non-trap error. Digits 23-11 = 1 for any error recorded in B91 by R700 which cannot be trapped. Digit 1 = 1. Digits 10-2, 0 = 0
(11) = Number of interrupt type errors which can be trapped, digits 8-2. Rest zero.
(7) = Table in fixed store, consecutive half words. Only digits corresponding to trappable interrupt type error are significant; entry i as 1 in i the number of this set zero remainder.
(9) = Trap mask; one in digits corresponding to trappable interrupt type error, zero elsewhere.

Cross references:

(4) = (7/700) Current program trap address
(5) = (1/702) Entry to Enter Monitor

Connection with other routines

Entry at (1) from trap extracode (1111)with
B121 = trap number n, digits 8-2
Exit to monitor, 1/702, if no trap table set or trap n is not set with
B91 = fault number "No trap set" digits 10-2. digit 1 = 1
Exit if trap set with B127 = contents of word n.0 of trap table
previous B127 in B register specified in word n.4 of trap table. B91 - 93, 121 altered.

Entry at (2) from R700 with B91 = Record of faults

Entry at (8) from extracode fault detection with
B91 = fault number, digits 10-2
Digit 1 = 1 (Non-trappable), 0 (trappable)

Exit: To R702 with B91 unaltered if any non-trappable error is recorded in B91.
If only trappable errors are recorded, the top priority error is selected and a trap is attempted as in entry (1). If unsuccessful, monitor is entered with B91 unaltered.

Temporary working space: B92, 93, 121

Note

1. On entry (2), the Current Program Trap address is reset to have digit 0 = 0 permitting further monitor entries via R700.

16.7.62

Purpose: An extracode routine to control entrance to the standard monitor sequence or to a private monitor sequence. Entrance is made via R701, when the error is not trapped. The routine prevents multiple entry to private monitor routines in the event of errors occuring in these and prevents interruption of monitor routines by off line failures such as magnetic tape failures.

Registers of Fixed Store: 20

Number of instructions obeyed: From 9 to 15

Parameters used: (1) to (6)

(4) = "Current programm monitor address".
One half word subsidiary store in the current program working area holding the address to jump to for private monitoring, digits 23 - 3
Digits 1,0 = 00 Enter after printing fault description
01 " before " " "
10 " after printing description and standard post mortem
If negative, no private monitor requested.

Cross references

(2) = (5/203) Current program member
(3) = (15/204) Programm switch drectory
(5) = (1/710) Entry of "Main Store Monitor"
(6) = (1/233) "Enter processing"

Connections with other routines

Entry at (1) with
B91 odd: On line machine error
B91 even, negative: off line error (e.g. tapes)
B91 even, positive: On line program error

Exit: a) To private monitor in main control
B92 = old value of M
B93, 121 altered
B91 digit 0 = 0 otherwise unaltered.

b) To "Enter Processing"
with B97 = Address of Main Store Monitor digite 23 - 3 Digit 0 = 0
B126 odd.
B91 unaltered except that
digit 0 = 0 if private monitor allowed
1 if " " not allowed.

Temporary working space: B121, B92, B93

Notes:

1. Private monitor is only entered if the current program is not in monitor already. Digits 1,0 of the program switch directory describe the monitor status of the program and read
0 0 Not in monitor
0 1 Monitoring on-line program error
1 0 " off-line error or failure
1 1 " on-line failure.

16.7.62

2. If R702 is entered for on-line failure, the monitor description is forced to 11. If for program failure or off line failure, it is set to 01, 10 if formerly 00, but otherwise is unaltered.

3. If the program is not in monitor already, private print is entered immediately if the Current program monitor address is positive and odd. Otherwise the standard monitor routine is entered.

R703 Block monitor

Purpose: To monitor the current program by entry to R700 when Store Location and lock out detects use of a reserved block label or detects an overflow of store allocation. In the latter case, monitor is not entered if the program is in compilation or has been temporarily reduced in store allocation by the supervisor.

Registers of Fixed Store: 30

Number of instruction obeyed : From 9 to 17

Parameters used: (1) to (20)

(3) = Monitor marker "Store exceeded" digits 8-2 Digit 1 = 1 if non trap
0 if trap
Remaining digits zero

(6) = Bits in program switch directory indicating compiler or supervisor store allocation. Provisionally digits 18, 19; (6) = *03

Cross references

(4)	=	(7/299)	Entry to SER "stop on block allocation"
(5)	=	(15/201)	Program switch directory
(7)	=	(4/204)	Non equivalence marker
(8)	=	(8/314)	Contents of (7) if non equivalence
(9)	=	(27/205)	Entry to Unlock Store Block
(10)	=	(14/700)	Entry to Monitor Interrupt and SER
(11)	=	(9/205)	Current program in store control
(12)	=	(7/202)	Current Program branch indicator
(13)	=	(1/232)	Entry to "Branch block monitor"
(14)	=	(1/202)	Entry to Program Scan
(15)	=	(1/215)	"Set full recovery switch"
(16)	=	(18/204)	Number of main programs
(17)	=	(7/204)	Main program short dumps
(18)	=	(9/204)	Program Status Directory

Connections with Other Routines

Entry at (1) from R203 if store allocation exceeded
B108 = Current program number
B109 = block label digits 22 - 12

Entry at (2) from R203 if block label limit exceeded with
B108, 109 as for entry (1)
B100 = monitor marker for this error.

Exit a) To "Set & reset full recovery switch" at (3) with
B100 = Fault marker digits 8-1 remainder zero as on entry
B109 = Entry address (14) or R700 digits 23-3; digit 0-1
(causes entry to R700 and thence to program scan)

b) To Program scan, with current program in control set up to resume in supervisor with full recovery of B line and working space, and to resume at 6(2/703) with
B100 = Fault marker digits 8-1, compiler/supervisor bits as in program switch directory after entry (1)
B101 = 0 if fault follows non equivalence, non zero otherwise
SER re-entry address = 6(2/703)

16.7.62

c) To SER "Step on block allocation" if entered at (1) compiler/ supervisor bits non zero.
B100 = as exit b
Main control reduced by 1 if fault does not follow non-equivalence

Subroutines:

(a) "Unlock store block" entered at (27/205) to free program with
B110 = 4(2/703)
B108 = Digit 23: 1
Digits 22 - 12 = block label
Remaining digits irrevelent
B106 = Current program number digits 8-2. Remainder zero.

Return to 4(2/703) with
B100, 101, 106 unaltered.

(b) "Branch block monitor" entered if program is in branching with
B100 as exit (b)
B101 as exit (b)
B106 = current program number.
Return to 10(2/703) with B100, 106, 109 unaltered.

Temporary working space: B100 - 102, 106 - 110, Bt

Notes:

1. If the fault follows non equivalence, the current program or branch is made free to proceed. Otherwise, main control is reduced by 1 before exits a,c. The non-equivalence switch in subsidiary store is reset.

2. If the program has been resumed In Supervisor and is not in store control, the program status directory is set to cause resumption with store control in supervisor, and re-entry is then made to R703.

3. The full recover switch is reset on exit to R700.

4. If store allocation is exceeded legally (when in compile phase or if the supervisor has reduced store allocation) a main store SER is entered to supply an extra block. The programm may be resumed without alteration to the control registers.

```
T
FIXED STORE COLUMN 40014

(0) = 159*40014



R703                                        |Block monitor


(3)  = 2.0                                  |Monitor mark-store exceed
(6)  = *03                                  |Compile & supervisor bits
(4)  = (1/708)                              |exit for extra block
(5)  = (15/204)                             |Switch directory
(7)  = (4/204)                              |# marker
(9)  = (27/205)                             |Free program
(10) = (14/700)                             |Monitor
(11) = (9/205)                              |Current program in store
(12) = (7/202)                              |Branch in dicator
(13) = (1/232)                              |Branch block monitor
(14) = (1/202)                              |Program scan
(15) = (1/215)                              |Set full recover switch
(16) = (3/204)                              |Main programs
(17) = (7/204)                              |Short dumps
(18) = (9/204)                              |Status directory



1)  121,   100,   0,     (3)                |ENTRY FOR STORE EXCEED
    101,   101,   108,   (5)
    164,   100,   101,   (6)                |)COLLECT COMPILE SUPERVISE BITS
2)  101,   101,   0,     (7)                |ENTRY FOR LABEL EXCEED
    165,   106,   108,   63.4               |)EXIT IF NON EQUIVALENCE
    210,   126,   101,   (20)               |)
    152,   106,   0,     (11)
    225,   126,   0,     (19)               |)EXIT IF PROGRAM NOT IN STORE CONTROL
    101,   102,   0,     (12)
    216,   126,   102,   (13)               |)EXIT IF PROGRAM BRANCHING
    210,   126,   101,   2(0)
    122,   127,   0,     1                  |)REDUCE M BY 1 IF NOT =/
    165,   101,   100,   (6)
    215,   126,   101,   (4)                |)EXIT IF EXTRA BLOCK PERMITTED
    121,   109,   0,     0.1(10)
    121,   126,   0,     5(15)              |)RESET FULL RECOVER SWITCH AND EXIT TO

                                            | R700 AND THEN PROGRAM SCAN
19) 113,   100,   106,   (17)
    113,   101,   106,   (16)(17)           |)SET UP PARAMETERS FOR RESUMPTION
    121,   100,   0,     5(2)               |)
    113,   100,   106,   (16)(16)(16)(16)(16)(17)
    121,   108,   0,     64.2               |)
    147,   108,   106,   (18)               |)RESUME IN STORE CONTROL AND IN SUPERVISR
    113,   108,   106,   (18)
    121,   126,   0,     (14)               |)
20) 113,   0,     0,     (7)                |RESET =/ SWITCH
    121,   108,   109,   0
    167,   108,   0,     *4                 |B108 = BLOCK LABEL p23=1
    121,   110,   0,     3(2)               |FREE PRODRAM
    121,   126,   0,     (9)

                                            |30/8/63.
```

Purpose: An SER entered when timers are updated and the instruction counter timer of the current program in store exceeds a check value. The routine causes monitoring if the local timer or absolute timer are exceeded; otherwise the check value and current counter are reset and block timers adjusted to keep the timers within range.

Registers of fixed store: 39

Number of instructions obeyed: 8 to 14 if monitor
Maximum of 18 + 10b, normal 15 + 8b if continue, where b is the number of blocks used by the current program in store

Subsidiary store:

(7) = "Overflow check timer" One half word in current program working area. Normally holds zero. If local timer is set, overflow timer holds the former check timer, if local timer is less than this, or the difference between the check times, digits 22 - 3

(8) = "Overall timer count" One half word in current program working area holding
Digits 23-15 N: No. of times overall check is to be reached before conclusion
Digits 14-3 N: No. of times overall check has already been reached.
Digits 2-0 L: No. of addition to check by monitor

Parameters used (1) to (14)

(11) = Extra time allowed for monitor, (digits 22-3)
(12) = Lower limit for blocktimers T
(13) = Monitor marker local time exceeded
(14) = Monitor marker total time exceeded

Cross references

(2) = 9(3/303) Return address to update timers
(3) = (2/700) Entry to Monitor Interrupt and SER
(4) = (21/303) Check time of current program
(5) = (20/303) Program Instruction Counter timers
(6) = (9/205) Current program number in charge of store
(9) = (4/203) Program store directory
(10) = (3/203) Block timers/status directory

Connections with other Routines

Entry at (1) from R303 when current timer exceeds check timer

Exit (a) to "Program Monitor Interrupts and SER" at 2/700 with
B126 odd
B102 = 9(3/303) = Return address
B100 = Monitor marker for local or overall time exceeded
B101 altered

Exit (b) : To 9(3/303) if more time allowed.
B104, 109, 110 unaltered

Temporary working space: B100-103, B105, Bt

Notes:

1. This routine is entered from "Update Timers" when the timer of the current program in store exceeds the check timer (also of the current program). Return is to continue updating timers.

2. If the local check time has been reached (check timer digit 0 = 1), it is replaced by the overall check timer and monitor is entered.

3. If the total allowed time has been reached, the check time is increased by a fixed amount to permit monitoring, and monitor is entered. This error cannot be trapped.

4. If further units of time are allowed, the overall timer count is adjusted, and the check value C is subtracted from the current timer and also from all timers T in the block timer directory. A lower limit (negative) is placed on these timers, to prevent overflow when subsequently computing page timers T. If a local timer is still in force, the check timer is adjusted accordingly.

```
T
FIXED STORE COLUMN 40034

(0) = 210*40034


R704                    |Instruction counter monitor

(11) = 1000                                 |4 sec. extra time
(12) = *6                                   |Lowest block timer
(13) = *01                                  |Local time
(14) = *1                                   |Total time

(2)  = 9(3/303)                             |Return to update timers
(3)  = (2/700)                              |Montior
(4)  = (21/303)                             |Check time
(5)  = (20/303)                             |Timers
(6)  = (9/205)                              |Program in store
(9)  = (4/203)                              |Store directory
(10) = (3/203)                              |Block timers

1)  101,    100,    0,      (4)             |<- ENTRY
    211,    126,    100,    7(0)            |JUMP IF NOT LOCAL CHECK
    121,    100,    0,      (13)            |SET MARKER LOCAL TIME EXCEEDED
    101,    101,    0,      (7)
    113,    0,      0,      (7)             |OVERFLOW CHECK TO ZERO
    113,    101,    0,      (4)             |RESET CHECK TIMER
    121,    102,    0,      (2)
    121,    126,    0,      0.1(3)          |)EXIT TO R700 AND THENCE TO R303
    101,    101,    0,      (8)
    165,    102,    101,    *777            |)JUMP IF COUNTER NOT EXHAUSTED
    215,    126,    102,    6(0)            |)
    124,    101,    0,      0.1
    113,    101,    0,      (8)             |)ACCUMMULATE ADDITIVE COUNT
    121,    101,    100,    (11)            |MODIFIED CHECK TIMER
    121,    100,    0,      (14)            |MARKER OVERALL TIME EXCEEDED
    121,    126,    0,      4(1)            |TO MONITOR
    124,    101,    0,      1*777
    113,    101,    0,      (8)             |)STEP TO NEXT SECTION
    101,    101,    0,      (6)
    101,    102,    101,    (9)             |)SET UP COUNTERS TO SCAN BLOCK TIMERS
    165,    103,    102,    1023.4          |)
    127,    102,    0,      *7776           |)
    101,    105,    103,    (10)
    122,    105,    100,    0               |SUBTRACT CHECK TIMER
    170,    105,    0,      (12)
    227,    126,    0,      3(0)            |)IF TOO SMALL REPLACE BY STANDARD
    127,    105,    0,      0.7             |)
    124,    105,    0,      (12)            |)
    113,    105,    103,    (10)
    124,    103,    0,      0.4             |)STEP THROUGH TIMERS
    122,    102,    0,      1024            |)
    215,    126,    102,    -9(0)           |)
    110,    100,    101,    (5)             |FORM NEW NUMBER OBEYED
    101,    101,    0,      (7)
    214,    126,    101,    4(0)            |)FORM NEW CHECK AND OVERFLOW CHECK
    124,    101,    0,      0.1             |)
    113,    101,    0,      (4)             |)
    113,    100,    0,      (7)             |)
    121,    126,    0,      (2)             |RETURN TO R303

                                            |18/3/63
```

R706 Set, read instruction counter

Purpose: To perform the extracode to set a local instruction counter (trap after n instructions) and to read the number of instructions remaining to be done.

Register of Fixed Store: 23

Instructions obeyed: 9 - 10 (Read)
9 - 18 (Set)
Interrupts inhibited for a maximum of 17 instructions.

Parameters used: (1) to (8)

Cross references:

(4)	=	(5/203)	Current program number
(7)	=	(20/303)	Program instruction counters
(6)	=	(7/704)	Program timer overflow
(5)	=	(21/303)	Check timer

Connections with other routines:

Entry at (1) from "read" with B126 odd (digit 0 = 1)
from "write" with B126 even
with B119 = n (write) { n = 0 : Reset local counter to master counter }
B121 = ba (read).

Exit: To main control with B119 altered, and ba holding no. of instructions yet to be done (digits 23-3) in case of read.

Temporary working space: B91, B101, B119

Notes:

1. The unit counter is 2048 instructions.

2. "Set I C" causes any previous local setting to be removed. If entered with n = 0, the counter resets to the overall counter set in the job description.

3. It is not possible to exceed the overall counter by use of "set". If n is in excess, "read" will subsequently give the number to go before the local trap, although in fact the overall trap will occur first.

4. If "read" is used before a local timer has been set, the result will be the number of instructions to go before the Supervisor check value is exceeded. In the case of a program of less than 30 mins.duration, this will be the number of instructions to be obeyed before an overall trap and monitor is entered.

R708 | Acquire blocks for compiler

```
(2) = *001                                  |4 excess blocks
(3) = (4/203)                               |Store directory
(4) = (5/201)                               |Re-entry address
(6) = (1/215)                               |Set fr switch
(5) = (21/261)                              |Alt. entry to End Compiling
(7) = (4/247)                               |Reset fr switch and exit

1)  121,   100,   0,     (7)                |Link for main store
    101,   102,   106,   (3)
    124,   102,   0,     (2)                |Step block counter
    113,   126,   0,     (4)
    121,   109,   0,     (5)
    121,   126,   0,     (6)                |Exit setting full recovery switch
```

R709 Off line program monitor

Purpose: To trap a program for failure occurring off line, when the program may not be currently in control of store e.g. looked down page when using extracode "Read to Page P", errors and faults in on-line peripherals etc.

Registers of Fixed Store: 27

Instructions obeyed: 5 to 9 on entries 1,2
13 on entry 3 (7 with interrupts inhibited).

Parameters used: (1) to (14)

(4) = Monitor marker for page trap [a single digit in range P22 to P12]
(10) = Mask of all other digits in range P22 to P11 = *3777 - (4)

Cross references:

(5) = (6/315) Alternative entry to drum queue
(6) = (8/227) Program monitor directory
(7) = (9/205) Current program in control of store
(8) = (16/202) Supervisor exit trap
(9) = (18/202) Normal setting of (8)
(11) = (7/201) SER base
(12) = (8/299) Entry to Main Store Off line monitor
(13) = (14/700) Alternative entry to Program Monitor
(14) = (5/201) SER entry address

Connection with other routines:

Entry at (1) from R331 (Page extracode) if page P locked down with
B100 = Page number P digits 11-3 remainder zero
B106 = Program number digits 8-2
at (2) from other off line error detection with
B100 = Error marker (digit in range 22-12 plus 0.1)
B106 = Program number
B110 = Return address from R709

Exit to Return address with B100 altered

Entry at (3) from R202 under interrupt control to trap before resumption of program.

Exit (a) to Main store off line monitor if any fault other than a page fault is detected with
B100 = Program monitor directory of current program.
Digit 23 of this set to 1 in store, otherwise unaltered
B102 = Current program number, digits 8-2

Exit (b) to Program Monitor Interrupt at (14/700) if only page error detected with
B100 = marker for page error
B119 = page no. P digits 11-3 Remainder zero.
Program monitor directory of current program set to zero except digit 1 unaltered.

Temporary working space: Entry 1 : B100, B110, Bt
Entry 2 : B100, Bt
Entry 3 : B100-103, Bt

16.7.62

R709 continued

Notes:

1. On entry 1,2 the program monitor directory of the program is set to hold the fault markers, digit 0 is set to 1, digit 1 remains unaltered. If the program is currently in control of store, the supervisor exit trap is set to cause entry to (3) before resumption of the program.

2. Before resumption of the program, if only a page trap has been detected, exit is to R700 and thence to normal trapping and monitoring. If any other fault has been detected, exit is to a main store routine to preserve working registers and to enter an off line trapping routine.

R710 Main store monitor

Purpose: A routine in main and extracode control, obeyed from core store, which prints a description of all detected faults (program and computer faults, where the latter affect an object program) on output stream 0 of the object program. On conclusion, exit is to a private monitor sequence or to the standard monitor post mortem.

Registers of Main store (program): 92

Instruction obeyed: Approximately 11X(no. of characters printed)

Main store tables and working space:

(16) = "Standard messages"

A table containing standard messages in internal code 6 bit characters, packed four to a half word each message ending with character (octal) 00 or 77. Total length and composition undefined as yet.

(15) = "Message locations for multiple faults"

A table of successive half words, entry n giving start relative to (16) of the monitor message for fault recorded as digit 9 + n in the record description in B91 on entry to this routine. 15 half words. Messages may start at character addresses within the table (16), not only at the beginning of words or half words.

(14) = "Message locations for single faults"

As (15) only half word n corresponds to fault no. n recorded in digits 8-2 of B91 on entry to this routine.

(12) = "Temporary dump area"

11 words holding original contents of B127, B119, B80 - 99 for recovery on exit from this routine.

(17) = "Powers of 10" 6 consecutive half words, entry n = 10^n, digits 23-3. Digits 2-0 zero.

Parameters used: (1) to (22)

(13) = Number of multiple fault indicators (maximum) in B91 on entry in digits 23-2 = 7.0 (digits 22-9)

Cross references:

(7)	=	(7/202)	Current program branch indicator
(8)	=	(6/203)	Page directory
(9)	=	(5/203)	Current program number
(10)	=	(15/204)	Program switch directory
(21)	=	(1/711)	Entry to standard monitor post mortem
(22)	=	(1/236)	Exit from processing

R710 continued

Connections with routines:

a) Entry at (1) from "End Program" extracode via "Enter processing" in extracode control.
Exit to (1/712) in extracode control with program branching killed if it was in use.

b) Entry at (2) from "Enter Monitor", R702, via "Enter processing" with B92 even.
B91 = Fault record
Digit 0 = 0 if private monitor allowed
1 if private monitor not allowed

Exit to (1/711) "Standard monitor post mortem" under main control if no private printing after descriptive print, with original contents of B127, B119, B80 - 99 recorded in successive half words from (12/710)

or to "Exit from processing" at (1/236) to enter private print with B119, B80-99 as on entry (B92 = original main control, B91 = fault record.

B127 = Private print address
B126 odd
Entry to R236 is under extracode control, entry to private print under main control.

c) Entry at (3) in main control to print message with
B85 = Start of message relative to start of table, (16)
B87 = Return address
Exit to return address with message printed up to character 0 0 or 77. If the terminating character is 0 0 only the message is printed. If it is 77, the message is followed by contents of B119 on entry, printed as "signed decimal integer.octal fraction". No layout is included. The selected output is used.

Working space: B81 - 83, B85 - 86, B91 - 97
If B119 is not printed, B85 hold next character position in table to return.

d) Entry at (4) in main control to print signed B line, B86
Entry at (5) " " " " " unsigned B line (digit 23=0)
with B86 = quantity to be printed
B85 = no. of figs/spaces required digit 5-2.
Remainder zero.
B87 = return address

No layout is printed. The selected output is used. For style of printing, see note (3). Working space: B81 - 83, B85 - 86, B91 - 97

e) Entry at (6) to re-enter private monitor sequence. Entry in main control. B80-99, B119 are recovered. B127 set and the private sequence entered under main control

R710 continued

Subroutines

Uses extracodes

1060	Select output
1064	Print 1 character
1066	End Record
	Pick up one character

Temporary working space: B81 - 93, B_t, B121

One exit to private print. B81-99 are restored but not B_t, B121.

Notes:

1. If program branching is in use, it is killed (made inoperative by setting a switch).

2. Output stream 0 is selected, and 3 "NL" are printed, followed by the fault description, followed by B119 if applicable, followed by "NL". If multiple faults have occured, descriptions are printed on separate lines.

3. B119 is printed as
   Sign (Space or Minus)
   Digit 22-3 as a decimal integer to at most 6 figs. Leading zeros are omitted and not replaced by spaces. If digits 22-3 are all zero, "0" is printed. If digits 2-0 are zero, spaces are printed so that the total number of characters excluding sign is 8
   If digits 2-0 are not zero, "." is printed followed by digits 2-0 (octal) and spaces are then printed so that the total number of characters excluding sign is 8.

The subroutine used to print can be used generally to print to any total number of figures by entry with B85 = number figures, digit 6-2. Here B85 = 4.0 The sign, if required, is additional to this number.

4. The routine operates in main store and requires modification to use fixed or subsidiary store. A temporary version, R720, can operate in subsidiary store.

R711 Standard monitor post mortem

Purpose: To print standard post mortem information on output stream 0 of an object program after detection of a fault. The instructions in location M,M-1 are printed, together with the contents of the B line used in these instructions. The contents of B lines 1 to 10 are then printed, followed by the accumulator (single length, unstandardised, as fraction followed by the octal exponent), if private magnetic tapes are in use, the position of each tape is listed. The routine finally tests whether the program is to be dumped, and exits to the dump routine or to End Program.

Registers of main store (program) 206

Instructions obeyed Principally print instruction, 11×(no. of character printed)

Subsidiary store

(44) = "Current program dump description"

On half word in the current program working area describing action on dump.
Digit 23 = 1 No dump required. Remainder irrelevant
Digits 23-0 = 0 system dump where applicable
Otherwise dump on private tape B block n
Digits 8 - 2 = B
Digits 21-9 = n

(45) = "Dump tape description"

One half word holding
0 if no dump tape
Digit 23 = 1 In use
Digit 0 = 1 Required for further use.
Remaining digits describe tape no. and current position (not used by R711).

Parameters used (1) to (47)

(13) = Limit of public block labels = *34
(19) = Number of B lines to be printed (digits 23-3)
(24) = Number of figures in accumulator print (digits 23-3)
(28) = Number of tape decks (digits 23-3)
(34) = Minimum instruction counter for dumps (digits 23-3)
(27) = Address of floating point zero, exponent -128
(26) = 1 word working space (main store)
(25) = Address of fixed point 10 : 0/1.2
(20) = Address relative to start of table of standard message (16/710) of "ORDER"
(21) = Address relative to start of table of standard message (16/710) of "UNALLOCATED"
(22) = Address relative to start of table of standard message (16/710) of "ACCUMULATOR"
(23) = Address relative to start of table of standard message (16/710) of "TAPE"

Cross references

(3)	=	(3/710)	Subroutine Print message
(4)	=	(4/710)	Subroutine Print signed B line
(5)	=	(5/710)	Subroutine Print unsigned B line
(8)	=	(1/201)	Enter Supervisor
(9)	=	(1/202)	Program Scan
(10)	=	(5/203)	Current program number
(11)	=	(4/203)	Program Store Directory
(12)	=	(12/710)	Working space
(14)	=	(2/203)	Block directory
(32)	=	(6/710)	Re-enter private monitor
(33)	=	(2/712)	Alternative entry to End Program
(35)	=	(3/204)	Halt main program
(36)	=	(1/715)	Entry to Dump object program
(37)	=	(5/221)	Deck allocation directory
(38)	=	(5/201)	SER re entry address
(46)	=	(8/704)	Current program overall times
(47)	=	(20/303)	Program timers

Connections with other routines

Entry at (1) from R710 with temporary dump area, (12/710) containing original M, B119, B80-99 in consecutive half words. Entry is in Main Control

Exit a) To End Program at(2/712)in main control if no dump required
b) To Dump object program at (1/715). Entry In Supervisor in extracode control with
B100 = Current program dump description (digit 23 = 0)
or Dump tape description, digit 23 = 1
c) To Halt Main Program at (1/204) if dump tape in use with
B110 = Program
B109 = Dump tape number (digits 8-2). Digit 0 = 1
Remainder zero.
Entry in Supervisor, using extracode control.

Alternative Entry for subroutine "Find decks"

Entry at (30) in extracode control (not in supervisor) with
B126 odd
B90 = one less than number of decks, digits 23-3
B84 = Return address in Main Control when deck directory has been scanned.
B96 = 2(30/711)

Exit (a) : Current main control if deck found with
B89 = Programmers deck label, digits 8-2
B90 = Absolute deck number, digits 8-3

(b) to Address in B84 in main control when no more decks found.

Re-entry after Exit (a) to continue search,
at (30) in extracode control with
B126 odd
B90 unaltered
B84 Return address when deck-directory
B96 = 10(30)
B89, 97, 100, 101 used as temporary working space

Subroutines

(a) "Print message" Entry at (3/710) in main control
B85 = Start of message relative to start of table.
B87 = Return address in main control.
Uses B81-83, 85-86, 91-97

(b) "Print signed B line" Entry at (4/710) in main control
B85 = no. of figures, digits 5-2
B86 = quantity to be printed
B87 = return address in main control
Uses B81-83, 85-86, 91-97

(c) "Print unsigned B line" Entry at (5/710) in main control
B lines as for (b)

(d) Enter Supervisor: Entered at (1/201) in extracode control with interrupts inhibited.
B96 = Entry address in supervisor
Exit to entry address in supervisor

(e) Program Scan: Entered at (1/202) to exit from supervisor to main control

(f) Re-enter private monitor: Entered at (6/710) in main control to recover B lines and exit to private monitor

Temporary working space B81 - 97, B_t

Extracode used:

Uses extracodes 1066 End of Record
1064 Print 1 character
1157 Exit to Extracode

Notes:

1. Printing includes current instructions, B lines Accumulator, and Tapes as follows:-

a) Current instructions Instructions M-1, M printed on separate lines.

ORDER (address M-1/M) (fn), (Ba), (Bm), (S) B(a)=(ba) B(m)=(bm)

24 bit registers printed as sign, digits 22-3 as decimal integer, digits 2-0 as octal digit following "." . These omitted if digit 2-0 = 0.
Function printed in octal, 4 figures leading zero omitted for basic codes Ba, Bm printed as decimal integers, 3 figures
B(a) = (ba) omitted if a = 0
If control address is in private store, or is ≥*34, or if the store block is not defined, the address is followed by "UNALLOCATED".

b) B lines Contents of B lines 1 to n are printed as signed decimal integers (digit 23-3) followed by "." (octal digit) unless digits 2-0 = 0. 4 B lines are printed per line of print.

c) Accumulator Printed on a separate line as ACCUMULATOR (Contents of M as fraction, signed unstandardised)/ (octal exponent as a decimal integer, signed). The exponent guard digit is ignored.

d) Tapes If private tapes are in use, these are listed on separate lines as TAPE (program label) at (Block number)/(Word no.) the word number being omitted if zero (e.g. if fixed block transfer used).

2. If private printing is allowed and has been requested at this tage, it is entered after the standard printing, recovering B119, B80-91, and setting B92 = original main control.

R712 End Program

Purpose: A routine obeyed from main store, entered from End Program and after standard monitor printing to terminate an object program. The number of instructions obeyed and tape waiting time if any are printed on output stream 0, all output streams are closed, all tapes if any disengaged and rewound, any reserved bands are lost and exit is then made "in supervisor" to the routine "Clear Program".

Registers of Main Store (program): 115

Instructions obeyed:

Subsidiary store:

(25) = "Effective tape engage time"
One half word in the current program working area holding effective time for which all private tapes have been engaged in units 0.1 secs. digits 23-0. Updated on engage, disengage tapes.

Parameters used: (1) to (38)

(15) = 2 words working space in main store, first half of each zero
(16) = 1 word working space in main store
(17) = Address of constant 0/500000
(18) = Address of fl.pt. 0.1, exponent zero
(20) = Address relative to start of table of messages (16/710) of "INSTRUCTION COUNTER"
(27) = Address relative to start of table of messages (16/710) of "TAPE TIME"
(32) = Parameter no. of "output stream record" in parameter list
(33) = 2 words working space for output stream = (1/712)
(34) = Number of output streams (digit 23-3)
(36) = Number of bands on drums

Cross references:

(3) =	(3/710)	Print message
(4) =	(4/710)	Print signed B line
(5) =	(5/710)	Print unsigned B line
(7) =	(1/201)	Enter supervisor
(8) =	(1/202)	Program scan
(9) =	8(6/710)	"Exit from processing" instruction in extracode
(10) =	(5/203)	Current program number
(11) =	(20/303)	Program timers
(12) =	(21/303)	Check timer
(13) =	(8/704)	Overall timer
(14) =	(11/704)	Time allowed for monitor
(19) =	(15/711)	3 figure print entry
(22) =	(3/303)	Update timers
(23) =	(28/711)	Number of tape decks
(24) =	(30/711)	Alternative entry to R711
(26) =	(4/702)	One word in Current Program working area to hold Inst. Counter.

R712 continued

(35) = (45/314) Number of drum transfer
(37) = Band directory
(38) = (1/215) Recover main
(39) = (1/714) Clear program

Connections with other Routines:

Entry at 1) in Extracode control from R710 on End Program
at 2) in Main Control from R711 or R715 on Monitor with output stream 0 selected.

Exit to Recover main at (1), in supervisor, with
B109 = Return address 1/714. In supervisor
Return to 1/714 to clear program.

Alternative entry from Extracode "Read Instruction Counter to Acc." via R233 (Enter processing)

Entry at 6)
Exit to R236 (Exit processing) via R710 with Acc. = Instruction counter, B91, 96-97 and B100-110 altered.

Subroutines:

(a) "Print message". Entry at (3/710) in main control
B85 = start of message relative to start of table
B87 = return address in main control
Uses B81-83, 85-86, 91-97

(b) "Print B line" Entry at (4/710) (signed) (5/710) (unsigned) in main control with
B85 = no. of figures, digits 5-2
B86 = quantity to be printed
B87 = return address in main control
Uses B81-83, 85-86, 91-97

(c) "Enter supervisor" Entered at
(1/201) with B96 = Entry to Supervisor routine,
return to M 1(1/201) with B96 = Entry to SER
B97 = Return address in extracode

(d) "Program Scan" Entered at (1/202). Exit to main or extracode

(e) "Update timers" Entered at (1/303) in extracode, in supervisor.
B110 = Return address in supervisor.

(f) "Find decks": Alternative entry to R711 at (30/711)
Entered in extracode control with
B126 odd
B90 = Deck counter
B84 = Return in main control when no more found
B96 = 2(30/711) on first entry
10(30/711) on subsequent entries

(g) Uses extracodes 1060 Select output
1066 End record
1064 Output one character
1157 Exit to extracode
1140 Read parameter
Pick one character
1067 Break output
1016 Disengage tape
Lose band

R712 continued

Temporary working space:

B81-97, Bt
Words (15), 1(15) (16) (33) 1(33) in main store block holding this routine.

Notes:

1. Printing occurs on output stream 0 of the object program and is proceded by 3 "NL"

2. The total number of instructions obeyed is printed in units of 2048. If in excess of 5.10^5, the number is scaled by 10 until it is within this range and the resulting value and exponent are printed
   e.g. INSTRUCTION COUNTER 4321
   or INSTRUCTION COUNTER 12345/2

3. If private tapes are in use, the total effective tape engage time is printed in hours, mins, secs. etc.
   e.g. TAPE TIME 12M 5.2 S

4. All tapes in use are disengaged (the extracode routine will print operator instructions) all output streams are terminated and any reserved bands are "lost". Note digit 0 of "no. of drum transfers" = 1 if any bands reserved.

5. The routine operates in main store, sharing a block with R710, 711. Should extra space be required, R712 can be broken. The block forms part of the object program during its use. Exit is to a supervisor block holding R714 "Close program".

R720 Temporary main store monitor

Purpose: A temporary version of R710 (main store monitor) for Supervisor Version 1. Prints a description of all detected faults on the output stream of the object program and exits to a private monitor sequence or to the standard monitor post mortem.

Registers of Main Store (program): 107

Instructions obeyed: Approximately 11X(no. of characters printed)

Specification: As for R710 in effects with the following alterations

a) Parameters (60) (64) (66) are used:

(60) = Enter for Extracode subroutine "Select output 0 for internal code". Entered in extracode control. Exit to return address in B99 in extracode.

(64) = Entry for Extracode Subroutine "Print 1 character"
Entered in extracode control with
B98 = character (digits 5-0)
B99 = Return address in extracode

(66) = Entry for Extracode subroutine "End record"
Entered in extracode control with
B98 = Carriage control information (as in extracode 1066)
B99 = Return address in extracode.

(21) = (1/721)

b) Temporary extracode 1477 is used

c) No notice is taken of program branching

d) The routine uses extracode control, except for subroutine (61) (64) (66), which enter main control and use extracodes 1060, 1064, 1066, 1477. Replacement of this section enables the routine to be tested in subsidiary store.

R721 Temporary standard monitor post mortem

Purpose: A temporary version of R711 for the intermediate supervisor prints a standard post mortem on the output stream of the object program and exits to a private monitor sequence or to End Program (R722)

Registers of Main store (program) 174

Instructions obeyed Principally print instructions, 11 x (no. of character printed)

Parameters used (1) to (45), (64), (66)

(13) = Limit of public block labels = *34
(19) = Number of B lines to be printed (digits 23-3)
(24) = Number of figures in accumulator print
(27) = Address of floating point zero, exponent -128
(26) = 1 word working space (main store)
(25) = Address of fixed point 10: 0/1.2
(20) = Address relative to start of table of standard messages (16/720) of "ORDER"
(21) = Address relative to start of table of standard messages (16/720) of "UNALLOCATED"
(22) = Address relative to start of table of standard messages (16/720) of "ACCUMULATOR"

Cross references

(3) = (3/720) Subroutine Print Message
(4) = (4/720) Subroutine Print signed B line
(5) = (5/720) Subroutine Print unsigned B line
(8) = (1/201) Enter supervisor
(9) = (1/202) Program scan
(10) = (5/203) Current program number
(11) = (4/203) Program store directory
(12) = (12/720) Working space (temporary dump area)
(14) = (2/203) Block directory
(32) = (6/720) Re-enter private monitor
(33) = (2/722) Alternative entry to End Program
(40) = (6/201) Main program controls
(64) = (64/720) Print one character
(66) = (66/720) End record

Connections with other routines

Entry at (1) in extracode control from R720 with temporary dump area (12/720) containing original M, B119, B80-99 in consecutive half words.

Exit: (a) to End Program at (2/722) in extracode control

(b) to (6/720) and thence to private monitor if private printing required.

Subroutines

(a) Print message: Entry at (3/720) in extracode control.
B85 = start of message relative to start of message table
B87 = Return address
Uses B81-83, 85-86, 91-99

(b) Print B line : Entry at (4/720) (signed), (5/720) (unsigned)
B85 = no. of figures, digits 5-2
B86 = quantity to be printed
B87 = return address
Uses B81-83, 85-86, 91-99

(c) Enter Supervisor: Entered at 1(1/201) in extracode control with interrupts inhibited.
B96 = Entry address is supervisor
B97 = Extracode address for exit from supervisor

(d) Program Scan: Entered at (1/202) to exit from supervisor to extracode control.

(e) Re-enter private monitor: Entered at (6/720) in extracode control to recover B lines and exit to private monitor.

(e) Temporary extracodes: R720 entered at (64), (66) to replace extracodes.

(64): Print character from B98 (digit 5-0) and return to address in B99 in extracode control.

(66): End record, B98 = Carriage control, and return to address in B99 in extracode control.

Temporary working space: B84-99, Bt

Notes:

1. Printing is as described for R711 except that section (d) (Note 1) is omitted.

2. Private monitor may be entered as in R711.

3. No dumping is permitted.

4. Extracode control is used throughout. Replacement of subroutines (64), (66) allow testing of this routine in subsidiary store.

R7 22 Temporary end program

Purpose: A temporary version of R712 (End Program) for use with the Intermediate Supervisor. After end program or monitor, punches run out characters on the output stream and enters an S.E.R. to the queue to start the punch. Clears the input well, disengages the reader where appropriate, and if required enters an S.E.R. to the queue to start the reader. Prints a message, including time, on operators output and finally exits to "Clear program".

Registers of Main store: (program): 69

Instructions obeyed: Around 136 + time to output characters

Main Store Tables:

(6) = "Current control number"
One half word, digits 4-3 = control desk in current control, remaining digits zero

(12) = "Punch control table"
4 alternate half words in main store, entry n for punches n,
Digit 0 = 1 if punch in action, 0 if idle
Digit 1 = 0 if waiting to be engaged, 1 otherwise
Digit 23= 1 if buffer empty, and not reserved, 0 if in use for current program.
Remaining digits not used or altered by R722

(13) = "Reader control table"
4 alternate half words in main store, entry n for reader n
Digit 0 = 1 if temporary stop, 0 if end of document or failure
Digit 1 = 1 if engaged, awaiting start, 0 if no engaged
Digit 23= 1 if buffer in use by program or awaiting use by program
0 if buffer empty or filling from peripheral
Remaining digits not used

These may occupy the same block of store as R720, 721, 722 or may occupy any supervisor block in main store.

Parameters used: (1) to (26), (60), (64), (66)

(4) = Length of run out. Number of UC characters to space output tape, digits 23-3
(7) = V-store address of TR5's = *600416
(17) = Location in main store of message "DESK n END" (see note 4)
(20) = Number of "NL" on output punch to space output, digits 23-3
(8) = Subsidiary store address of TR5's.

Cross References:

(5) = (1/201) Enter supervisor
(9) = 7(3/230) Enter SER to slow queue
(10) = (2/724) Temporary Start punch SER
(11) = (5/724) Temporary Start reader SER
(14) = (5/201) SER Re-entry address
(15) = (1/220) Reserve operation output
(19) = (1/240) SER Print string on operators output

R722 continued

(18) = -3(54/724) Disengage and free reader
(21) = (6/229) One second clock
(22) = (12/229) Excess clock counter
(23) = (2/240) SER Print one character on operators output
(24) = (3/240) SER Print layout
(25) = (2/220) Free operator output
(60) = (60/720) Select output
(64) = (64/720) Punch one character
(66) = (66/720) Punch layout
(26) = (1/723) Temporary clear program.

Connections with other routines

Entry at (1) from End Program via R720 in extracode control
Entry at (2) from monitor via R721 in extracode control

Exit to "Temporary clear program and job scan" via 2/220 with
B91 = current control number

Alternative entries, in supervisor,

At (16) from "Temporary start job" to print DESK n STARTED, time with B95 = Return address to Supervisor.
B94 = Start of message (in character 3, half word S K (space n blank)
B108= Current control number, digits 4-3
Exit to return address with B93, 94, B105-110 altered

At (3) to print time with operators output 0 reserved
B95 = Return address in supervisor
Return after printing time, NL, with output channel freed,
B93, 94, 105-110 altered

Subroutines:

a) Temporary extracode: R720 entered at (60), (64) (66) to replace extracode

(60) : Select output 0 for internal code and return to address B99 in extracode.
(64) : Print character from B98, digits 5-0, and return to address B99 extracode.
(66) : End record, B98 = Carriage control, and return to address B99 extracode.

b) "Enter supervisor" Entry at 1/201 with interrupts inhibited,
B96 = Address of entry to supervisor

c) "Enter SER to queue" Entry at 2/206 via 7(3/230)
B107 = 0 (slow queue)
B108 = Current control number
B109 = Start of SER
B110 = Return address
Return with B106, 107 altered.

d) Start punch, start reader SER's entered via 2/206 with B100 = Private store of Reader

e) Reserve and free operators output; Entered at 1/220 with
B100 = 0
B110 = Return address

R722 continued

f) "Operators output print", R240, entered at
(2/240) : Print 1 character, internal code, from B109, digit 5-0
(1/240) : Print storing, internal code, from store location in B109, digits 23-0. B108 = no. of characters.
(3/240) : Print layout. B109 = carriage control information as for extracode 1066. B110 = Link in all cases.
(28/240): Break Output.

Temporary working space: B90-99, B100, B107-110

Notes:

1. A succession of NL is punched, followed by a succession of UC

2. An SER is entered to the slow queue to start the output punch. If not engaged, the request waits until the punch is re-engaged, when output is continued.

3. The input well is recorded as empty. If the reader is engaged to read in more of the current program, it is disengaged. If it is engaged waiting to read another program on SER is entered to the slow queue to start reading.

4. The message

DESK (n) END (Time) (NL)

is printed on the operators output. Time is printed as

(Hours) . (Mins) . (Seconds) . (Tenths of seconds)

R723 Temporary clear program and job scan

Purpose: A temporary supervisor routine for the intermediate supervisor, entered from End Program (R722) to lose all store blocks of the current program, and to scan for the next available program. If no control desk is ready, the idling program is entered.

Registers of Main Store: 31

Instructions obeyed: Maximum 32 + 14 (no. of blocks).

Parameters used: (1) to (18)

(4) = Lowest reserved block label, digits 22-3. Rest zero
(7) = V store address of teletype punch = *60042
(18) = Current control desk: One half word in main store holding desk in current control (digit 4-3). Rest zero.

Cross references:

(3)	=	(1/727)	Temporary "Start Job"
(5)	=	(5/203)	Current program number
(6)	=	(9/207)	Leading program number relative to program status directory.
(8)	=	(16/222)	Entry to timed idle
(9)	=	(3/203)	Block timers
(10)	=	(6/202)	Program change marker
(11)	=	(14/314)	Alternative entry to Lose Block
(12)	=	(12/722)	Punch control table
(13)	=	(13/722)	Reader control table
(14)	=	(4/203)	Program store directory
(15)	=	(9/204)	Program directory

Subroutine

"Lose block" entered at 1(14/314) to halt if block locked out and lose block otherwise

B101 = Block label (digits 22-12), rest
B100 = Return address
B105 = Block directory location
B107 = Lock out status (digits 2-0 of Block Timers)
B102 = Program number

Return to return address with B105 unaltered.

Connections with other routines:

Entry at (1) in supervisor from R722 with B91 = current control number

Exit: (a) To start program with B91 = new control number, digit 4-3

(b) To idle via (16/222) if no program ready, which enters program scan on conclusion.

Temporary working space: B90-94, B100-107

Notes

1. Only blocks with labels less than a present limit are lost from the current program area. These entries are set to zero, and the block timers remain unaltered.

2. The control desks are scanned cyclically from the current number to find one for which the reader is stopped and waiting and the output buffer is empty. If one is found, exit is to start program. Otherwise exit is to the idling program, with the program change marker set and leading program number zero. Subsequently re-entry to program scan causes cycling of the idle program until a new job is ready. The current program is left unaltered.

R801 Octal Input

Purpose: Simple input for paper tape for use during development or for engineers' programs. It will operate only during Engineers' Interrupt and uses the engineers tape reader TR5 number 0. It is intended to read 5 channel tape but will also read the 5 least significant digits of 7 channel tape.

Registers of fixed store: 64

Temporary working space: The only working space used is the flip flop B-registers:

B120, B121, B123, B124, B125, B126, B127, Bt

Interrupt control is used and all other interrupts are inhibited for the whole of the time that the tape is being read.

Functions used:

101	107	113	121	124
147	163	164	165	172
210	214	216	217	225

Entry: (1) By manual jump on interrupt control to the first address, =*4005 5300

(2) Set h.s. = 001 000 00 Press "Engineers Interrupt" Press "Start". This tests all the instructions used by octal input, the B-core store, the B-adder and the subsidiary store before entering octal input with all 24 display lamps lit (B120=*7777 7777) if tests are successful.

Exit: In interrupt control, to the address specified by the warning sequence on the tape.

Tape Preparation:

The routine is intended for use with 5 channel tape. 7 channel tape can also be read but the two m.s. digits are ignored, and the remaining five treated exactly as for 5 channel tapes.

The tape must be punched in Mercury/Pegasus code using only the following characters:

Octal digits Sp LF CR Er FS * →

The octal digits are packed into successive half words of the store. The first half of a word must be terminated by Sp and the second half by LF. Both halves of each word must always appear. Left hand zeros in any half word may be replaced by Sp, or left out altogether, provided that at least one octal digit remains.

Sp (apart from the terminating space) will be ignored between half words.

LF (apart from the terminating line feed), Sp and FS, will be ignored before full words.

CR and Er are ignored everywhere.

Warning Sequences:

*followed by an address of up to 8 octal digits is used to indicate that subsequent words are to be read into the store starting at this address.

→followed by an address of up to 8 octal digits stops the reader and causes a jump to be made, in interrupt control to this address.

The warning sequences must end with LF.

Note that unless the page address registers have been set appropriately before entry to this routine it will not be possible to read into the main store.

Punching Errors:

If a punching error is found on the tape, the reader is stopped and the computer comes to a loop stop. At the same time the lamps in B120 are lit to indicate the type of error according to the following code.

*7770 0000	A character which is not in the permitted list had been read from the tape.
*7770 0001	No warning character at the beginning of the tape.
*7770 0002	More than 8 octal digits in a half word.
*7770 0003	First half of a word terminated by LF.
*7770 0004	Second half of a word terminated by Sp.
*7770 0005	FS or warning character in the first half of a word.
*7770 0006	FS or warning character in the second half of a word.

Method of Operation:

Before entering the routine the tape must be loaded on to the engineers' reader, TR5 number 0, and the Engage button should be pressed.

The operator may halt the reader at any time by pressing the Disengage button on the reader. While disengaged in this way the computer will be executing a closed loop within the routine, repeatedly testing the Disengaged digit. To resume reading the tape the operator should engage the reader again.

R980. Set Up Compiler Directory.

Purpose: To examine all active list jobs for the compiler they require, set up the compiler directory and enter the compiler assembly routine when necessary.

Connections: Entered at (1) from R660 and exits to the same routine.

Subroutines:

1. ''find compiler name'' entered at (1) with
B104 (23-2) = document address.
B110 = link.
and this returns to the link with compiler name in B102 & B103.

2. ''locate compiler entry'' entered at (1) with
B101 (22-13) = compiler number.
if compiler number unknown set B101 = 0 and
B102 & B103 = compiler title.
B110 = link.
Exit with
B109 = compiler entry position in CD
(compiler directory)
if B109 = 0, compiler not listed in CD.
Uses B104 - 110 as working space.

3. ''assemble compiler'', entered at (1) with entry conditions as in 2.
Exit with B109 as 2 but also
B109 (0) = 0/1, assembly complete / otherwise.

Notes:

1. This routine is entered when.....

a) a job has been assembled in main store.
b) a compiler is assembled in main store.

2. The assemble compiler routine assembles the compiler specified in the list in main store and also sets up the compiler directory entry if this is not already done. After queuing the requests return is made to the link address but it is not until the assembly of some compiler is complete that a new entry is made to R980 via the scheduler line.

3. The routine also sets the compiler space required (ax+b) in each document entry and the maximum one for the job in the active list entry.

30.7.63

R982. Find or free halted program.

Purpose: To find if there is a current job on the execution list halted for a particular reason and if necessary free it.

Connections with other routines:

Entered at (1) with
B106 = return address, even if find program
odd if free program.
B109 = reason for halt.

Exit if B106 even with
B107 = 0 if there is a program halted for this reason
≠ 0 if there is no program halted for this reason

Subroutines:

30.7.63

R983. Tape assembly for execution jobs.

Purpose: To arrange for the mounting of a tape which is called for by a job during it's execution.

Connections with other routines:

Entered at (1) from R660 and exits to
(1/660) if no decks available or
(1/983) if no job halted for tapes

Subroutines:

a) Enters (1/982) to find or free or halted job with
B106 = link, even to find program & odd to free program.
B109 = reason for halt.
return of B106 even with B107 = 0 if program halted.

Notes:

1. When a job calls for a tape if there is a deck free the tape is immediately mounted other wise the job is halted & the scheduler is entered.

30.7.63

R984. Tape deck allocation for active list jobs.

Purpose: To allocate tape decks and initiate mounting of all tapes required by jobs on the active list.

Connections with other routines:

Entered at (1) from R983 and exits to R660 directly but there is a secondary exit after tapes are mounted.

Subroutines:

a) Enter (1/985) to find state of decks with
B110 = link & add.
Exit with B103 = number of decks available.

b) Enter (2/985) to obtain free deck. with
B110 = link.
Exit with B108 = deck number (zero if no decks available).

c) Enter (6/498) to mount tape on deck d with
B107 = address of title (zero title if free tape).
B108 = d
B109 = return after tape mounted and checked.
B110 = immediate return.
Exit to B110 with
B126 = odd if tape already mounted, even otherwise
Uses B105 - 107, & 109, B108 unaltered.

d) Enter -n(2/667) to shift B105 n places to right with
B110 = link.
Exit after shift to link.

Notes:

1. When an input document is required from a private tape the archive tape needed bit is set and the tape mounted in exactly the same way as an archive tape.

2. The routine does two passes of all active list jobs, on the first pass only archive tape assembly is considered and on the second private tape assembly. The only exception to this is that private tapes associated with top priority jobs are assembled before other archive tapes.

3. Private job tape assembly is only iniated when all decks required by that job became available.

4. When private tape assembly begins the tape needed bit is reset and a tape assembly bit set till all the tapes are mounted and checked. The archive tape needed bit is not reset till all the archive tapes needed by this job have been mounted.

R985. Obtain a free deck.

Purpose: To ascertain the number of decks which could be made available and where necessary obtain a free deck.

Connections with other routines:

a) Entered at (1) with
B110 = return address.

Exit with
B103 = number of available decks if B110 odd.
or
B108 = deck number if B110 even.
B103 = 0 if none available.

Subroutines:

a) Enter (4/498) to find deck of given property with
B107 = mask (lists of interest in deck ownership directory)
B109 = value to be found after masking.
B110 = link.
Exits with
B106 = 0 if one found ≠ 0 otherwise.
B108 = deck number if one found, 0 otherwise.
Find others by re-entry at 5(4/498) with B108 preserved.
Uses B106 & B108 only.

b) Enter (5/498) to find deck of given property & title with
B107, 109 & 110 as for (a) and
B101 = address of title.
Exit with
B106 = 0 if no agreement, ≠ 0 if agreement
B108 = deck number.
Uses B105, 106 & 108.

c) Enter (7/495) to dismount tape with
B100 = deck number.
B110 = link.
Returns to link after disengage given & uses all but B100.

Notes:

1. Decks are allocated in the following order, free decks, decks with free tapes, private tapes & archive tapes, with the lowest numbered deck of each class being made available first.