Contents List

Contents 5.2 Monitoring

Monitoring on Program Events

There are several circumstances under which the machine, when obeying an instruction will interrupt (and hence cause an entry into OMP.) This instruction is not completed and no results are written away. Interruptions of this sort are called program events; the term program failure (prog fail) is used to distinguish unanticipated program events.

Following the interruption, the OMP Kernel makes a partial analysis of the circumstances, taking notice of 3-address unreplaced 150 instructions and implementing *OWN monitoring. All further analysis and action is then left to an OMP job called the Interpreter, thus all monitoring actions except *OWN require at least one drum reference. For *OWN (see 5.4)

5.2.0.1 Program Failures

Unanticipated program events (program failures) which may occur are.

	Event	Message
(i)	Impermissible operand	IMP.OPER
(ii)	Illegal instruction	ILL.INST
(iii)	Writing with OVR set	W.W.OVR
(iv)	Reservation violation	RES.VIOL
(v)	Peripheral violation	PER.VIOL

(vi) and (vii). If monitoring signals in Style 1 is on then jumping to a signal or obeying a signal is a program failure (if these were anticipated then monitoring signals in another style would be on) the messages being JUMP.SIG and HLTD.SIG respectively.

(viii) and (ix). If monitoring floating point overflow in Style 1 or monitoring fixed point overflow in Style 1 is on and an appropriate instruction overflows then this program event is a program failure, the messages being FLT.OVR and FXD.OVR respectively.

When an instruction fails because of one of the above program events then OMP's default action in cases (i) to (iv) is to suspend, and in cases (v) to (ix) is to halt the job. Information is then printed; on the Flexowriter appears the following

```
            jobname     Message
```

and if the job has no monitoring peripheral then also on the Flexowriter is printed on the next line the instruction (or string) causing the failure, but if the job has a monitoring peripheral then on it is printed the same message and the instruction (or string) together with the values of all replacers and modifiers.

If *IMP or *PFP or *PFM has been set then OMP enters the appropriate restart (see 5.2.9 and 5.2.10, 5.2.20 and 5.2.23)

Examples of printout (replacers and modifiers on mon per only)

(a)

```
PER.VIOL
A617    116      0     (A1)             (A1) = 100
A618    141.1Y   0       2     A2       (A2) = 420
A619    142     (A4)   (A5)             (A4) = A530 (A5) = 128
```

This indicates that the drum region 522 to 649 extends beyond the reserved region.

(b)

```
ILL.INST
A401    130     0       0       0
```

(c)

```
RES.VIOL
A239    04Y     (A27)   36     A28      (A27) = 4 (A28) = A300
```

Note that addresses outside reservations are printed as integers: address within reservations as integers relative to the datum point with an A in front.

(d)

```
RES.VIOL
A247    04      (-1)   A36              (32767)
```

If the address being replaced is outside reservations it is indicated as above.

(e)

```
RES.VIOL
5104
```

The control number is outside reservations.

5.2.0.2 Optional Events

		Section	Style name
(i)	Signals and jumps to signals	5.2.1	*SIG
(ii)	Jumps	5.2.2	*JUM
(iii)	Floating point overflow	5.2.3.0	*FOV
(iv)	Fixed point overflow	5.2.3.1	*OVR
(v)	Drum and peripheral transfers	5.2.6	*DRU or peripheral name
(vi)	Impermissible operand	5.2.9	*IMP
(vii)	Program failure	5.2.10	*PFP or *PFN
(viii)	Timer overflow	5.2.4	*TIM
(ix)	Unrounded floating point	5.2.5	*UNR
(x)	Urgency of program	5.2.8	*URG
(xi)	Weak reservations	5.2.11	*WEA
(xii)	Quick jump	5.2.12	*QUI

Events (i) to (iv) cause interruptions if appropriate staticisers are found to be set when the event occurs (152 instruction see 3.15). Drum transfers are made to interrupt by setting the number of words reserved equal to 1 (the hardware cannot deal with a reservation of zero words and this means that single word transfers to or from drum address zero cannot be monitored.) Transfers to or from a particular peripheral are made to interrupt by effectively removing the peripheral from the program's reservations. Timer overflow is detected by the pre 150 instruction (see 3.15).

The necessary conditions are controlled by the object program (or possible the operator) using either 150/20 instructions or MONITOR directives. For the optional events there is a choice of styles. Style 0 usually means no monitoring or the program is allowed to monitor in “style 7” which means that the action is taken by the object program itself (see 5.2.7 - note that style 7 does not apply to *UNR and *URG.)

The following sections describe the standard styles of monitoring and which style is set when the program is initially entered (i.e. which is the default style).

It is possible that if a program is monitoring several events, the same instruction may call for several monitoring actions. To deal with this, events (i) to (vii) are assigned a hierarchy - the order is that given with event (i) being low. If a program is monitoring signals, jumps and fixed point overflow and a signalled jump instruction overflows the monitoring will (or at least may) take place in order. i.e. signals first, then jumps and then overflow.

In branched programs each branch sets its own monitoring styles, the monitoring applying to that branch only i.e. different branches may monitor the same event in different styles, except for transfers to the same peripheral or drum. All branches are held up (suspended) only while OMP’s action for the monitoring takes place.

Some monitoring information will be output only on the monitoring peripheral. If it is disengaged the message on the Flexowriter asks for it to be engaged, e.g.

```
            jobname     ENGAGE      SPB*
```

If the job has no monitoring peripheral then the job is halted and a message is printed on the Flexowriter,

```
            jobname     NO    MON.PER
```

If the monitoring peripheral fails while OMP is outputting monitoring information, then OMP outputs a message on the Flexowriter informing the operator of the failure e.g.

```
            jobname     LPA   OMP   BUFFER      FAIL
```

In most cases OMP will repeat the transfer and/or carry on outputting the monitoring information. If the failure is of Operator type then OMP disengages the device. The operator should carry out appropriate action (such as reload more paper if paper low) and then engage the device when OMP will repeat the transfer and/or carry on outputting the information.

If the Flexowriter fails ╫ NL is output and the message repeated.

5.2.1 Monitoring on Signals

Cause of interruption: the program has attempted to obey or to jump to a word which has a zero signal bit.

Style 0 - No monitoring. i.e. a signalled instruction would not cause an interruption and would be obeyed as if not signalled.

Style 1 - (Default Style). The program is halted and the failure message HLTD.SIG or JUMP SIG is printed as described in 5.2.0.1; in this case this program event is a program failure. Note that RUN (see 5.7.3.3) will cause the job to continue.

Style 2 - Jumps to signals are ignored (by OMP). Details of each signalled instruction are output on the monitoring peripheral and the program continues. The information is output on one line in the following order:-

*S, Control number, function, effective X-address, effective Y-address, Z-address (3-address instructions only) - and then any addresses written to, followed by their new contents, in octal.

A monitoring peripheral is necessary.

No printing takes place for 140, 141, 142 or 143.

Style 7 - Jumps to signals are ignored (by OMP). For details of style 7 monitoring, see 5.2.7.

5.2.2 Monitoring on Jumps

Cause of interruption: The program has attempted to obey a jump instruction which would have been successful.

Style 0 - (Default style). No monitoring.

Style 1 - This is similar to monitoring signals in style 2, the output being preceded by *J instead.

A monitoring peripheral is necessary.

Style 2 - When monitoring in this style OMP keeps a record of the last 16 distinct jumps made by the program and outputs them on the monitoring peripheral (or some on the Flexowriter) if one of the following events occurs:

(a) Program is stopped because of program failure (see 5.2.0.1)

(b) A style of monitoring on jumps is set.

(c) The program is abolished.

When this style is first set, OMP annexes the last 16 words of the program's core store for the space in which to record the information. These words are not returned to the program when the style is changed; if the program subsequently resets this style however, the same 16 words are used.

The actual information output is the function, the effective destination address and the control number; under the heading J TO FROM. Loops containing a single successful jump instruction will be represented by one entry in the list, with a number printed to the right of the standard information indicating the number of jumps that occurred. If this number is greater than 511, the number 511 will be printed. This Style slows the program down by a factor of 10.

Style 3 - This is the same as style 1, except that printing takes place on 86 instructions only.

Style 7 - For details of Style 7 monitoring, see 5.2.7. Monitoring on jumps is switched off on entry to the routine and restored by the 150/23 instruction - provided that the 150/23 would cause monitoring on jumps to be ignored.

5.2.3 Monitoring on overflow

5.2.3.0 Monitoring on floating point overflow

Cause of interruption: The program has attempted to obey an instruction with one of the functions 90-95 or 102 and which would have overflowed.

Style 0 - No monitoring.

Style 1 - (Default Style). The program is halted and the failure message FLT.OVR is printed as described in 5.2.0.1, in this case this program event is a program failure. Note that RUN (see 5.7.3.3) will cause the job to continue.

Style 2 - This is similar to monitoring signals style 2, the output being preceded by *O. A monitoring peripheral is necessary.

Style 7 - In addition to normal style 7 action (5.2.7) the address to which the instruction was about to write is stored in the X—address position of the word containing the link. OVR is not set on entry.

5.2.3.1 Monitoring on fixed point overflow.

Cause of interruption: The program has attempted to obey an instruction which would have overflowed.

Style 0 - (Default style). No monitoring.

Style 1 - The program is halted and the failure message FXD.OVR is printed as described in 5.2.0.1, in this case this program event is a program failure. Note that RUN will cause the job to continue.

Style 2 - This is similar to monitoring signals style 2, the output being preceded by *O. A monitoring peripheral is necessary.

Style 7 - In addition to normal style 7 action (5.2.7) the address to which the instruction was about to write (or in the case of instructions with double-length write-back the first address) is stored in the X-address position of the word containing the link. OVR is not set on entry.

5.2.3.2 Interaction between fixed and floating point OVR

There is no special floating point overflow register. Floating point operations which overflow, set the ordinary overflow register like other instructions. Floating-point overflow is only distinguishable from fixed-point overflow when it is being monitored. In the case where both are being monitored it should be noted that both monitoring actions take place for each floating point operation which overflows - the floating point monitoring first. If, for example, both are being Monitored in style 7 and the exit from the floating-point routine is a 150/23 instruction (q.v.) with *FOV in the X-address then the fixed point routine will be entered. If *OVR is used the program will continue - see also 5.2.7

5.2.4 Timer overflow

Cause of interruption: Any interruption

Style 0 - (Default style). The job is halted and the timer is set to a minute and the message TIME UP is printed on the Flexowriter only. If RUN directive is given then the program continues.

Style 7 - The job is given another minute before the style 7 routine is entered. The return to the program must be with an 87-instruction - and never with a 150/23 which would cause impermissible operand action.

Timer overflow event is not a real program failure and, for example *OWN (Style 7) or *PFP (Style 7) etc. would not be entered.

5.2.5 Unrounded Floating-point operations

There is no interruption.

Style 0 - (Default style). From now on all floating-point operations are rounded.

Style 1 - From now on all floating-point operations are unrounded.

This facility is intended to be used to measure rounding errors. Programs may be run with floating-point operations rounded and again unrounded. Comparison of results then indicates the number of reliable significant figures.

5.2.6 Monitoring on External Transfers

5.2.6.0 Monitoring on drum transfers

Cause of interruption: The program has attempted to obey a 141, 142 pair or a 150/50 instruction.

Style 0 - (Default action). No monitoring.

Style 1 - Details of each drum transfer or 150/50 instruction are printed on the monitoring peripheral and the program continues. The information printed is:

*D drum address, control number, mode, core store address, length of transfer, and the jump address if a 150/50. Mode 2 means 150/50.

A monitoring peripheral is necessary.

Style 2 - Monitoring is on 150/50 only. Ordinary drum transfers are not slowed down.

Style 7 - This is permitted. See 5.2.7

Note that single word transfers referring to drum address zero cannot be monitored.

5.2.6.1 Monitoring on peripheral transfers

Cause of interruption: The program has attempted to obey a 140, 142 pair referring to the particular peripheral.

Style 0 - (Default style). No monitoring.

Style 1 - Details of each transfer (rewind is included) are printed on the monitoring peripheral and the program continues, the information printed is: programmers name of peripheral, control number, mode, core store address, length of transfer. A monitoring peripheral is necessary. If the monitoring peripheral is the peripheral being monitored, then the monitoring information will appear before the actual transfer.

Style 7 - This is permitted. See 5.2.7

In branched programs only one branch may monitor on external transfers to a particular peripheral or drum at any one time (see 5.3.20).

5.2.7 Monitoring in Style 7

This is set up with a 150/20 instruction (see 5.3.20); the X-address specifying the event and the Y-address the entry-point of the routine so that the program itself can deal with the event. When the event occurs, OMP’s action is to set the program’s control number to Y and to store the link information in Y-1 (i.e. the modifier half contains the current (old) control number and the sign bit is the state of OVR). OVR is then cleared and OMP’s action finishes by entering the routine at Y. If the program is branched all branches are held up only while OMP is storing the link information etc.

Return to the main program is then usually effected by using a 150/23 instruction, specifying the link address in the Y-address and the event in the X-address. The instruction

```
               150      *SIG     A100     23
```

has the same effect as

```
               87       0        A100
```

except that the 150 instruction causes monitoring on signals and lower events to be ignored while obeying the instruction in $[A100]_m$. Higher levels of monitoring remain active. Thus if the program is monitoring OVR is Style 1 it would be halted if the instruction overflowed.

It is quite admissible to write

```
               150      *DRU     A100     23

         or    150      *PFP     A100     23
```

both meaning obey the instruction in $[A100]_m$ with signals, jumps, overflow, peripheral transfers all monitored Style 0.

If the event being dealt with occurs in the Style 7 routine (e.g. a signal in a routine to deal with signals) the link back to the main program will be lost and chaos is likely to ensue. The exception is that jumps are allowed in a Style 7 routine for monitoring on jumps. (see 5.2.2). It is insanitary for two branches to monitor the same event in Style 7 with the same entry-point.

5.2.8 Urgency of Program

The normal time-sharing system on Orion ensures that all jobs within the machine have their fair share of time - in general arranging peripheral-limited jobs at the top of priority list and mill-limited jobs at the bottom. For certain applications, however, it is desirable for this to be overridden either by a 150/20 instruction or a Basic Monitor sequence in the program or by the operator typing a Monitor directive (event *URG) on the Flexowriter. The styles are:

Style 0 (Default style). Program reverts to its natural place in the time sharing system.

Style 1 The program is treated as baseload and goes to the bottom of the priority list.

Style 2 The program is treated as urgent and goes to the top of the priority list.

If there is more than one program of urgency 1 or 2 they will be time-shared among themselves in the normal way. Different branches of the same program may have different styles of urgency but a new branch is given the style of branch 1, not style 0.

A message job—name URG 0 1 or 2 is printed whenever the style is changed.

5.2.9 Monitoring Impermissible Operand.

Cause of interruption: The program has attempted to obey an instruction containing an impermissible operand.

Style 0 - (Default style). The program is suspended and the failure message is printed as described in 5.2.0.1.

Style 7 - In addition to the normal Style 7 action (see 5.2.7} the write back address is stored in the X-address field (not for failing 150's) and the string count, modulo 128 (this is 1 unless a compound instruction) in the Function position of the link word. If the program returns with 150/23 with X=*IMP then the program gets suspended as usual, unless the program is monitoring on program failure, in which case that routine gets entered.

The following function cause impermissible operand action.

(a) 40-45, 95, 101, division by zero.
90-95, 97, 103-non standard floating point operand.
100 - incompatible data and radices.
125 - x* < 0
142 following 140 or 141 if Y=0 or > 32767
150/2 - being in closed down state.
150/3 - Y > 1
150/4 - X > 5 or Y not correct
150/10 - Y > 6
150/13 - Y > 3
150/15 - Y < -1 or Y > 8+n
150/16 - impermissible style or Y=0 or Y > 20
150/17 X > 1
150/20 X not recognised event or not 0.
150/21 - Impermissible code number.
150/22 X not for an output device
150/23 X not a recognised event
150/24 X=0 or X > 7
150/30 X not a meaningful programmers name.
150/31 Y > 3 or X for reserved tape deck or output device
150/33, 34 or 35. Impermissible document name or X not meaningful.
150/36 Peripheral not the same
150/40, 42 X not for a tape deck
150/41 or 44. X not for a tape deck or impermissible doc. name
150/43 Y > 1 or X not for a tape deck.
150/50 Transferring zero words
150/51 X=0 and y=0, or referring to SBIP not associated with program

5.2.10 Monitoring on Program Failures

Cause of interruptions: The program has attempted to obey an instruction which would normally prevail and cause suspension or halting as described in 5.2.0.1.

Style 0 - (Default Style). The program is suspended or halted and the failure message is output as described in 5.2.0.1

Style 7 - In addition to the normal Style 7 action the write back address (not for failing 150's) and the string count, modulo 128 (this is usually 1 unless it is a compound instruction) is stored in the X-address and Function fields respectively of the link word.

Monitoring on program failures is of two forms *PFP or *PFN. If the job has a monitoring peripheral, then in the first case then the failure message is output to the monitoring peripheral before entering the Style 7 routine whereas in the second case no printing takes place before entering. No message appears on the Flexowriter.

If the routine exits with a 150/23 with X=*PFP or *PFN, then if the reason for the progfail was one of the cases (i) to (v) see 5.2.0.1 then the normal printing will take place (again in the case of *PFP with a monitoring peripheral) and the program will be suspended (halted if peripheral violation). Since 150/23 with X=*PFP or *PFN says ignore all monitoring while obeying the instruction, the program may continue with no printing taking place if the reason for the progfail was one of the cases (vi) to(ix).

Switching on or off of either of *PFP or *PFN switches off the other.

Note that because monitoring on program failures may now be set up, the manual in some places notably Sections 2 and 3, is not strictly accurate. Where it states that the program will be suspended due to reservation violation, impermissible operand etc. it is now true to say that reservation violation action etc will occur i.e. if these failures are being monitored the Style 7 routine will be entered, otherwise the standard action of suspension etc will occur.

5.2.11 Weak reservations *WEA (see 5.3.4)

A program (job) can switch on this condition with

```
      150      *WEA     1        20
```

and then can read (but not write) outside its reservations.

The condition is switched off by

```
      150      *WEA     0        20
```

The general arrangement is that one program declaring itself the Master should have fixed datum point and other programs declare themselves subsidiaries. In general each will have two or more buffers and communication between the Master and subsidiaries is via 150/4 instructions and the master's interrupt routine. Subsidiaries know where to look for markers as to whether there is a buffer with information for them. The master can scan the subsidiaries to see if they have information to deal with, the markers in this case being in fixed places relative to their datum point.

The master can be interrupted as described for 150/4 and also if a subsidiary is abolished or if a subsidiary restores monitoring conditions to standard.

5.2.12 Quick jump instructions *QUI

This facility is used to speed up programs by not checking the destination addresses of successful jump instructions. It is available only on installations with a suitable hardware modification.

Style 0 - (Default style) Destination addresses are checked for being within reservations.

Style 1 - The program will not be checked on its jump address in a jump instruction. Thus a program jumping outside reservations will have a RES.VIOL at the location it has jumped to, rather than the jump instruction itself.

Contents 5.3 - 150 Instructions

5.3 Special 150-instructions in Programmer's Mode

Only 3-address form is allowed and may be described as:-

"Call in Orion Monitor Program (OMP) to perform action Z on operands X and Y"

For legal values of Z, the permissible values of X and Y vary for each Z and are described for each 150-instruction.

The 150-instructions involve OMP reading from the drum (except for Z = 1 or 2 or 3 or 4)

The X-address and Y-address fields of legal 150-instructions may be replaced in the normal way and may be preceded by 116 and 117 instructions unless the contrary is explicitly stated. Replacement and pre-modification involve an additional drum reference.

2-address forms of 150-instructions and illegal values of Z will cause illegal instruction action.

Erroneous values of X and Y will cause reservation violation, peripheral violation or impermissible operand action as appropriate.

If a 150-instruction is obeyed in a branched program then all branches are held up (suspended) until OMP has completed the 150-instruction.

Note that if the instruction format is used e.g. for the second word of a 150/16 or 21 or 50 instruction, then the 3-address instruction format should be used; the 2-address unmodified instruction format causes D8 (TX) bit to be set.

The 150/1 Instruction - Timing Flag

```
e.g.  150   5     0     1
```

X is an integer

Y-address field must be zero.

This instruction sets the job's timer to X seconds. It does not involve a drum transfer.

Each timing flag obliterates all preceding ones; i.e. if a job requests 5 seconds and then 3 seconds later, requests a further 5 seconds, its timer will be set to 5 seconds, not 7.

When a job is accepted, its timer is set to one minute.

Mill time used by the job is subtracted from this timer. If the requested time is used up then timer overflow action will take place.

The 150/2 Instruction - Branch Interlock (see 10.1.4)

(i) For $Y \neq 0$

```
e.g.     150      3       A107     2
```

This instruction switches on branch X if X is waiting for this current branch (if X is waiting for a third branch then X is switched on but 1 is subtracted from its control number, this normally makes X repeat the 150/2 instruction which switched it off). If $y < 0$ then this instruction also switches off this (current) branch and records it as waiting for branch X.

If X is for the current branch and $y < 0$ then this branch is switched off awaiting for itself - this can be used to shut down a branch.

(ii) For $Y = 0$

```
e.g.     150      4       0        2
```

In this case branch X is not switched on, but if X is already switched on or is switched off an waiting for a branch other than the current one, then the current branch is switched off and recorded as waiting for X (i.e. unless X is switched off waiting for this branch switch off this branch waiting for X)

If X is for the current branch, then this branch is shut down waiting for itself.

Only the least significant 3 bits of X are used. If X is for a non-existent branch or zero then illegal instruction action will occur. If X is in a closed down state because it is temporarily unbranched, then impermissible operand action will occur.

The Y-address (when $Y \neq 0$) is not normally checked for either lockouts or reservations, this is to enable the instruction to be made fast. Care must be taken to ensure that the marker in Y is not involved in a peripheral transfer since if it is, the wrong marker may be examined. The effect of having Y which is either locked out or outside reservations will be similar to any other wrong Y address.

It is illegal for the 150/2 instruction to be pre-modified, this is because it would cause trouble in the case where subtracting one from the control number is expected to restore the instruction which switched the branch off. Both X-and Y-addresses may be replaced but this will be slow and in this case this Y-address is checked for both reservations and lockouts.

The 150/3 Instruction - End of HPD interrupt

This special fast 150 instruction is available only on machines fitted with a Hough Powell device - on other machines its attempted use will cause suspension due to illegal instruction.

The X-address must be that of a Hough Powell device reserved by the program - otherwise the program will be suspended due to reservation violation. The Y-address must be 0 or 1 - otherwise the program will be suspended due to impermissible operand.

When the program's restart on the HPD is entered, further interruptions are ignored until the program obeys a 150/3 instruction with Y=0 which causes OMP to start listening to interruptions again.

The case Y=1 says abandon the current transfer on HPD X. This is intended to be used in a routine emptying the 128 word buffer store after an end of frame interrupt.

The 150/4 Instruction - Weak Reservations

The cost in space is 64 words of core store and less than 200 words of drum. It will be possible for any installation to have this package built into its OMP but it is not desirable for an installation to have such an OMP in addition to its normal OMP. On Orion 1 if it is intended to run program which contain 142 instructions which refer to locations outside the reserved region then a hardware modification is also necessary - otherwise such instruction will be interpreted by OMP and will therefore be very slow.

The weak reservation package provides the following additional facilities.

1. A program may obey the instruction

```
150      *WEA     1        20
```

which means “from here on allow me to read from any address in the core store”. The condition is switched off with

```
150      *WEA     0        20
```

2. It will be possible for one program in the machine to declare itself a master and for other programs to be subsidiaries. Some means of communication between the master and its subsidiaries is provided. In particular it will be possible for the master to keep a list of which subsidiaries are loaded and. to know where they are in the machine. The idea is that one program, the master, may read in diverse data which can then be processed by several other programs. The master and any of the subsidiaries may be branched but the overall limit of not more than 15 branches in the machine remains.

The 150/4 instruction has the following specifications:

(a) X = 0, Y = N (0 < N < 16)

This says switch off this subsidiary in Style N.

Errors:	(i)	Y out of range.	IMP.OP
	(ii)	This program not a subsidiary	IMP.OP

(b) X = 1, Y = N (0 < N < 16)

This says switch on all subsidiaries which are switched off in Style N.

Errors:	(i)	Y out of range	IMP.OP
	(ii)	This program not the master	IMP.OP

Note that X = 0 and X = 1 are fast if there is no replacement or premodification.

(c) X = 2, Y = IP

This says I am the master and IP is my interrupt point.

Errors:	(i)	There is already a master	IMP.OP
	(ii)	IP or IP-2 not in reservations	RES.VIOL

(d) X = 3, Y = M (0 < M < 16)

This says I am subsidiary M. Interrupt the master, storing the master's link in IP-1, and store M in the m.s. half and my datum point in the l.s. half of IP-2.

Errors:	There is already a subsidiary M	IMP.OP
	M out of range	IMP.OP

If there is no master, the subsidiary is left halted.

(e) X = 4, Y = 0

This says remove this subsidiary. Interrupt the master as in the case X = 3, but in addition set the sign-bit of IP-2 equal to 1.

Errors:	(i)	This is not a subsidiary	IMP.OP
	(ii)	Y ≠ 0	IMP.OP

(f) X = 5, Y = IP-1

This says, exit from master's interrupt routine. Obey the link in Y as in the instruction

```
150      IP-1     0        25
```

Errors:	(i)	Y or $[Y]_m$ not in reservations	RES.VIOL
	(ii)	This is not the master.	IMP.OP

When the master is in its interrupt routine (i.e. following a 150/4 with X = 3 or 4 in a subsidiary), it will not be interrupted again until after it has obeyed a 150/4 with X = 5. If the master is branched, then the control path of Branch 1 is used for the interrupt routine and the other branches are temporarily closed down as in the case of peripheral incident routines or ENTER with links. If the subsidiary attempts to interrupt the master when the master is in its interrupt routine, it will be suspended until the master obeys 150/4 with X = 5.

Further error X > 5 causes IMP.OP.

The 150/10 Instruction - Stop

```
e.g.     150      8        0        10
```

X is an integer.

Y is an integer.

This instruction stops the job in mode Y and prints X (if X ≠ 0)

(a) Y = 0 This mode causes the job to be halted. A message of the following form is printed on the Flexowriter.

```
jobname  HALTED   Value of X (if ≠ 0)
```

(b) Y = 1 This mode causes the job to be halted until a document is loaded anywhere; when the job will continue by obeying the instruction after the 150/10 (this mode is for use by Basic Input). There is no printing on the Flexowriter and the X-address is ignored.

(c) Y = 2 This mode is used when the job is to be rerun (see 5.7.2.2)

If a rerun-tape has not been loaded when this instruction is obeyed then the job is suspended awaiting rerun. A message of the following form is printed on the Flexowriter.

```
jobname  SUSPD.RERUN      Value of X (if ≠ 0).
```

When the rerun-tape is loaded the job is rerun.

If a rerun-tape has been loaded when this instruction is obeyed then the job is rerun.

When the instruction has been read the job will be re run so long as the job's core-store reservation is at least 1008 words, otherwise reservation violation on the 150/10 instruction will take place. A branched program is temporarily unbranched when re-run. 150/25 may be used to restore the branch conditions.

(d) Y = 3 This mode causes the job to be suspended. A message of the following form is printed on the Flexowriter.

```
jobname  SUSPD   Value of X (if ≠ 0)
```

(e) Y=4,5 or 6 These modes are intended to be used after an unsuccessful request for more core-store, drum-store or peripheral. These modes cause the job to be halted until some core-store or drum-store or peripheral becomes available when the job will continue by obeying the instruction after the 150/10.

The message for Y=4 is of the following form:-

jobname NO SPACE Value of X (if ≠ 0)

The message for Y=5 is of the following form:-

jobname NO CORE Value of X (if ≠ 0)

The message for Y=6 is of the following form:-

jobname NO DRUM Value of X (if ≠ 0)

For jobs which are halted with Y=0 or 1 or 4 or 5 or 6, RUN directive (see 5.7.3.3) will cause the job to continue by obeying the instruction after the 150/10.

For jobs which are suspended with Y=2 or Y=3, ENTER directive (see 5.7.3.4) will cause the job to continue by loading and entering the chapter.

If the job is branched then all the other branches will be stopped in the some mode, except for Y=2 when OMP will temporarily unbranch the job.

(f) Y=7 Program will be HALTED AWAITING SPACE but no message will be printed on the Flexowriter.

If Y>7 then impermissible operand action will take place

The 150/11 Instruction - Abolish.

```
e.g.     150      1       0       11
```

X is an integer.

Y-address field must be zero.

This instruction terminates the job. It causes the peripherals reserved for the job to be relinquished and after appropriate terminating action, disengages the devices (see 5.3.31).

The core-store, drum-store and devices allocated to the job are thus made free.

After outputting the relinquish message for each peripheral reserved for the job, two lines of the following form are output on the Flexowriter.

```
ABOLD    Mill     Peripheral        Value of X (if ≠ 0)
         time        time

Local Civil Time
```

All times are in hours, minutes and seconds.

Mill time is the mill time used by this job.

Peripheral time is the total mill time used by higher priority jobs while this job was in the machine.

Local Civil Time is the time at which the job was abolished.

For example, the instruction given for a job whose name is BLOGGS may produce the following printing

```
BLOGGS   RLQD     *SR1-TRB

         RLQD     *SP1-SPA

         RLQD     *MT1-MTD

         ABOLD    0.05.21  0.10.56  1

         15.18.37
```

150/12 Instruction - Date and Time

```
      e.g.     150      A100     A200     12
```

This instruction asks OMP to give the date and time.

Into X is stored the date in character form; the characters are in the standard internal code (see 5.6.1)

the m.s. character C0 is SP

C1 and C2 give the date (SP1 to 31)

C3, C4 and C5 give the month (JAN to DEC)

C6 and C7 give the last two digits of the year

e.g.   SP21FEB65

Into Y is stored the date (in a mixed radix form) and the time

Bit D0 is zero

Date in D1 to D23

Bits D1 to D11 give the year (value 1965 for year 1965)

Bits D12 to D15 give the month (value 1 for Jan)

Bits D16 to D20 give the date (value 1 for the 1st of the month)

Bits D21 to D23 give the day of the week (value 1 for Sunday)

Time in D24 to D47

This field, a binary integer, is the number of seconds since midnight.

If X=Y then the information obtained is the date in the mixed radix form and the time.

If X or Y is outside reservations then reservation violation action will occur.

150/13 Instruction - Message to Flexowriter or Monitoring Peripheral

```
e.g.     150      A100     0        13
```

X is the address of a core-store register.

Y is an integer.

This instruction asks OMP to output one line of information (i.e. the message) on the Flexowriter or the job's monitoring peripheral.

The message in character form, is stored in the core-store starting at the m.s. end of the register whose address is X. The message should be terminated by NL character.

The characters and maximum number allowed in the message, depend on the type of device being used for output of the line containing the message.

Y = 0

A line containing the message is output to the Flexowriter, OMP precedes the message by the jobname and appends full stop character at the end of the message.

The characters allowed are those of the standard internal code (see 5.6.l) Character-value 7 and character-value 63 are not output. The error character (character-value l4) is output as underline character.

The maximum number of characters allowed in the message is 80. If no NL character is present within the ten words X to X+9, then the first 80 characters are taken as the message.

Y = 1

A line containing the message is output to the monitoring peripheral. OMP does not output the jobname nor full stop character.

The characters allowed are those available on the particular type of device being used as monitoring peripheral (see 5.6). Character-values not in the particular code are output as the appropriate error character (character-value 14) for that particular code.

The maximum number of characters on

a 7-track paper tape punch is	104
a 5-track paper tape punch is	69
a line printer is	118
a card punch is	80

If no NL character is present then the maximum number is taken as the message.

If the job has a monitoring peripheral but it is disengaged then OMP asks the operator to engage it; a message of the following form is output on the Flexowriter

```
jobname  ENGAGE   SPB*
```

When this is done the line containing the message is output

If the job has no monitoring peripheral, then OMP outputs the message to the Flexowriter, treating it as the case Y=0.

Y = 2

This differs from the case Y = 1 only if the job has no monitoring peripheral, in which case, the job is halted on this 150/13 instruction. OMP outputs a message on the Flexowriter to inform the operator, e.g.

```
jobname    NO MON.PER
```

The operator may use OUTPUTON (see 5.7.3.7) and RUN (see 5.7.3.3) directives to cause the job to continue by obeying this 150/13 instruction again.

Y = 3

This differs from the case Y = 1 only if the job has no monitoring peripheral, in which case OVR is set and the job continues by obeying the instruction after the 150/13 instruction.

Core-store region (X onwards) containing the message must be within reservations, otherwise reservation violation action will occur.

Y > 3 causes impermissible operand action.

If OVR is set then writing with overflow set action will occur.

150/14 Instruction - Question to Flexowriter.

```
      e.g.     150     A100     A200     14
```

X is the address of a core-store register.

Y is the address of a core-store register.

OMP outputs a line containing the question stored in X onwards on the Flexowriter, and then reads a line containing the answer into Y onwards.

The question in character form is stored in the core-store starting at the m.s. end of the register whose address is X. The question should be terminated by NL character. The characters allowed are those of the standard internal code (see 5.6.1). The maximum number of characters allowed is 80.

If no NL character is present within the ten words in X to X+9 then the first 80 characters are taken as the question.

OMP outputs two lines to the Flexowriter. The first line contains the question preceded by the jobname and is terminated by a question mark. The second line contains the characters

```
               TB       ANS      SP
```

and the Flexowriter is then left in the select state waiting to receive the answer which, if known will be typed on this second line by the operator. The answer is terminated by NL character.

If the answer is not known the operator must free the Flexowriter by typing NL character. This answer will be taken to mean that the true answer will be given later and so the job is then suspended awaiting answer. The answer when known is given using ANSWER directive (see 5.7.3.5). When the answer is given either on the Flexowriter or via a paper or card reader, the answer will be stored and the job allowed to continue.

The answer must be less than 80 characters long and must be terminated by end of line character (NL character on 7-track, CR LF characters on 5-track, on cards one card is one line). The answer will be stored in the core-store: the first character is stored at the m.s. end of the register whose address is Y. The NL character is stored and terminates the answer. The characters after the NL character are SP characters, in the word containing the NL character.

The answer punched on the Flexowriter or 7-track or 5-track paper tape or cards may contain ER or BS characters or VS. OMP forms a "clean line" (i.e. ER and BS characters are removed accordingly, ER characters are removed) and it replaces VS by a single SP character. The first character stored is the first correct printing character.

When the 150/14 instruction is obeyed the core-store region, X onwards (containing the question and NL character) and Y must be within reservations otherwise reservation violation action will occur for the 150/14 instruction.

When the answer comes to be stored then the registers (up to a maximum of ten) actually used for the answer are checked for being within reservations. If all the answer has not yet been stored and reservations are violated, then an error is reported for the line containing the ANSWER directive, this error being for reservation violation. The job is suspended awaiting answer.

If the answer is more than 79 characters then an error will be reported for the line containing the ANSWER directive, the error being for impermissible operand. The job is suspended awaiting answer.

If OVR is set then writing overflow set action will take place.

If the job is suspended awaiting answer and ENTER with a link is entered, then this enter sequence should not obey a second 150/14 instruction.

The 150/15 instruction - Read directory

This instruction has the effect "copy word Y of my directory to X". It enables the program to find out information set by the job tape as stored by the monitor program in the directory - e.g. what peripherals are reserved for the job. X must be a working store address within reservations and Y must lie in the range $-1 \leq Y \leq 8+n$ where n is the number of peripherals reserved for the job. Violation of these restrictions causes suspension due to reservation violation or impermissible operand respectively.

Y = -1 (or 32767) Drum Reservations. x'_m = drum datum point, $x'_u = -N$ where N is the number of words on the drum reserved for the job. If N=1 this means that the job is monitoring on drum transfers; it is then only possible to find out the drum reservations by switching this monitoring off.

Y = 0 produces no useful information.

Y = 1 Time available $x'=-t$ where t is the amount of time (in units of 16 microsecs) reserved for the job till the next timing flag.

Y = 2 Link x'_m = address of the 150/15 instruction (or the beginning of the string leading to it). x'_s = the state of overflow before the instruction was obeyed.

Y = 3 Job Name x' = the job name in character form, left justified.

Y = 4 produces no useful information.

Y = 5 Reservations and Monitoring The X address position gives the datum point and the Y address the upper reservation. D0, 1, 24, 25 are 1 if the job is monitoring on signals, overflow, jumps and floating point overflow respectively. D26 is 1 if the job is doing unrounded floating point operations.

Y = 6 Time used x' = T where the time used by the job (in units of 16 microsecs) is T-t. To find the time used add the contributions for Y = 1 and Y = 6.

Y = 7 Miscellaneous Information D0 is a 1 if the program is monitoring on *OWN - in which case the Y-address position is one less than the address jumped to on program failures. D24-26 are zero if the program is not branched, otherwise they give the branch number. D28-32=m, where m is zero if the job has no monitoring peripheral other than the Flexowriter, otherwise a 150/15 instruction with Y-address m will yield the peripheral word corresponding to the monitoring peripheral.

Y = 8 First Peripheral Word x'_u = programmers name for the first peripheral reserved for the job. D24 is 1 if the job is monitoring transfers on this peripheral. The integer contained in D40-47 has the value 253 if the peripheral is floating.

Y = 9,10,..7+n give the corresponding information for the second, third...nth peripheral reserved for the program.

Y = 8+n x'=0. To find the value of n obey 150/15 instructions with Y = 8, 9, etc. until a zero word is obtained.

The 150/16 Instruction - Printout region of Job's store

```
      e.g.     150      *APE     *IFG     16
               00S      A125      20      1
```

The instruction and the word in the register after it provide information which cause OMP to printout up to 20 words of either core or drum store in various styles onto the job's monitoring peripheral.

The first word

```
      e.g.     150      *AI      0        16
```

The X and Y address fields specify the styles of printing required. Permissible styles are APEIFG; impermissible styles or no style specified will cause impermissible operand action.

The second word specifies drum or core, the starting address of the region, the length and the condition Z (i.e. whether the printing of clear words is to be suppressed).

Drum or core

If Z = 2 bit (i.e. D30) of this word is 0 then the core is specified, if this bit is 1 then the drum is.

Starting address

If RX bit (D24) of this word is 0 then the upper half (D0 to D23) of this word gives the starting address of the region to be printed.

If RX bit (D24) of this word is 1 then the X-address field (D9 to D23) of this word is the address of the core-store register containing in its l.s. half, the starting address of the region.

The region specified must be within reservations otherwise reservation violation action will occur.

Length

If RY bit (D25) of this word is 0, then the Y-address field (D33 to D47) gives the length of the region.

If RY bit (D25) of this word is 1 then the Y-address field is the address of the core-store register containing in its l.s. half, the length.

The length specified must be non-zero and not greater than 20, otherwise impermissible operand action will occur.

Condition Z

If Z = 1 bit (i.e. D31) is 1, then the printing of clear words is suppressed otherwise not.

Styles and their printout

(i) A - prints the address of the register referred to (8 characters)

e.g. A5 for core-store register A5, D106 for drum-store register 106.

(ii) P - prints the content as program (29 characters)

```
e.g.      80S      0        83       0
```

The function is output followed by X,Y or S if necessary; 140 and 141 functions with the mode preceded by full stop.
The X and Y address fields are printed as core-store addresses if within reservations e.g. A51; if not these are printed as integers; they are printed as negative integers if greater than 32192; Y-address field for 140 and 141 is printed in the appropriate form e.g. *MT3. Replaced addresses will be bracketed. The Z-address field is printed unless TX = 1 and TY = 0 and Z = 0 (i.e. the instruction is 2-address unmodified). Otherwise the Z-address field is printed and unless it is zero or unless the instruction is a 150, it is printed as a core-store address e.g. A5.

(iii) E - prints the contents in octal 16 characters)
e.g. 2000000000000123

(iv) I - prints the contents as a signed integer (16 characters)
e.g. +70368744177747

(v) F - prints the contents as a signed fraction (16 characters)
e.g. +0.5000000000005

(vi) G - prints the contents in floating - point form (16 characters)
Non-standard numbers are printed as *
e.g. +1421085471 -14

For each register, the required styles are printed on one line with two spaces between each style.

Printout is to the monitoring peripheral. If the job has no monitoring peripheral and more than one word or one word with more than 4 styles specified is to be output, then the job is halted and the message NO MON.PER is printed on the Flexowriter. If only one word with fewer than 5 styles is specified is to be output and the job has no monitoring peripheral then the printout of the word will be to the Flexowriter.

150/17 Instruction - Output Directory Information

```
      e.g.     150      0        0        17
```

X is 0 or 1

Y-address field must be zero

This will output information about the job's directory. If X = 0 and the job has a monitoring peripheral the information is output on it, otherwise on the Flexowriter. If X = 1 the information is only to the Flexowriter even if the job has a monitoring peripheral.

The information given is

(i) the amount of core, the datum point and the amount of drum reserved,

```
e.g.     CORE     2736     DATUM    4096     DRUM     8704
```

(ii) the control number, state of OVR and state of the job,

```
e.g.     CN       A126     OVR      0        HALTED
```

(iii) the mill time used and the time left,

```
e.g.     USED     0.01.30  LEFT     0.00.57
```

where time used includes the time used by OMP for the job, as well as the time used by the job itself; time left is time asked for minus time used by the job itself;

(iv) the events being monitored in a style other than 0,

```
e.g.     MON      S J F
```

(v) the peripherals reserved and devices allocated to the job,

```
e.g.     *SP1     SPB      *MT1     MTB      *MT3
```

The state of the job given in line (ii) may be HALTED, WT.DOC, WT.RERUN, SUSPEND, WT.ANS, WT.SBP, WT.SPACE, WT.MASTR (for weak reservations), otherwise the job is going.

The letters in line (iv) may be S (signal), J (jumps), O (overflow), F (floating-point overflow), D (Drum), I (impermissible operand), R (program failure), T (timer), U (unrounded), P (priority), B (baseload), X (strong reservations - weak reservation machines), H (high density job - high density installations), Q (quick jumps), OWN followed by the address.

For branched programs (ii), (iii) and (iv) are given for each branch, together with the branch number and the number of the branch, if any for which it is waiting. If output is to the monitoring peripheral then information about all the branches is output whereas if to the Flexowriter information is about that branch only.

Line (ii) e.g. .2 ON A313 OVR 0 AWTS .3

X must be 0 or 1 otherwise impermissible operand action occurs.

If the monitoring peripheral is disengaged, then the ENGAGE message is output on the Flexowriter and output continues when this is done.

The 150/20 Instruction - Set Monitoring Style

e.g.   150   *SIG   2   20

X is an event
Y is a style

This instruction has the effect "set monitoring on event X in style Y".

Events, this is one of

Own monitoring (see 5.4)	*OWN
Signals	*SIG
Jumps	*JUM
Floating point overflow	*FOV
Overflow	*OVR
External transfers	*DRU or peripheral name
Impermissible operand	*IMP
Program failure,	
with printing	*PFP
without printing	*PFN
Quick jumps	*QUI
Timer overflow	*TIM
Rounding floating point	*UNR
Urgency	*URG
Weak reservations (see 5.3.4)	*WEA

If X is zero, then the effect is to restore monitoring on all events to the default styles.

Style, this is an integer between 0 and 5, or a core-store address, known as Style 7 (see 5.2.7). For a list of styles on each event and default styles (see 5.2.)

If X is neither zero nor a recognised event then impermissible operand action will occur.

In a branched program only one branch may monitor on transfers to a particular peripheral (drum included) at any one time; if another branch attempts to set up monitoring to the same peripheral then impermissible operand action occurs, unless the first branch has previously set its monitoring style on this peripheral to Style 0.

If Y is not a permitted style or Y and Y-1 are not within reservations then reservation violation action will occur.

If X=*URG a message is printed on the Flexowriter informing the operator of the change of urgency e.g.

```
jobname  *URG     2
```

0 is normal, 1 is baseload and 2 is top priority.

The 150/21 Instruction - Set Peripheral Incident

(see also 5.5 and 5.3.25)

```
      e.g.     150      A100     A200     21
```

X is the address of a core-store register.

Y is the address of a core-store register or 0.

This informs OMP that should a specified incident (event or failure) occur on a specified peripheral then because of this interruption OMP is to enter the job's peripheral incident routine at Y with OVR clear. OMP, if the incident occurs also records the job as being in a "pushed down" state; a 150/25 in the routine should be obeyed to "pull up". Exit from the restart in general will be by a 150/25.

Storing of the link information, the failure information and output of the message are optional.

Possible incidents that may occur on a particular type of peripheral device have been given code numbers (see 5.5).

The Specified Incident and Peripheral Device

OMP obtains these from the word in X as follows:

Specified Incident

If RX bit (D24) is 0 then the X-address field. (D9 to D23) is the code number of the incident being specified. If RX is 1 then the X-address field is the address of the register containing at the l.s. end, the code number of the incident.

Specified Peripheral Device

If RY (D25) is 0 then the Y-address field (D33 to D47) is the programmer's peripheral name of the peripheral being specified. If RY is 1 then the Y-address field is the address of the register containing at the 1.s. end, the programmer's name of the peripheral.

Link Information, Failure Information and Output of a Message

The Z-address field in X is looked at by OMP to see if the programmer wants these, should the incident occur.

Link Information

This is stored in the Link address i.e. in the register whose address is Y-1. The sign bit records the state of OVR and the modifier half is the control number of the job at the time of the interruption. Note that if incidents occur simultaneously the restarts are chained.

Failure Information

This is stored in the register whose address is Y-2. Certain bits record the state of the device and addresses concerned with the transfer (see 5.5)

Output of Message

This is on the Flexowriter and is of the form

jobname	Geographical Name of Device	Type of Incident

If D31 (i.e. Z = 1 bit) of the word in X is 0, then the message will be output; if 1 then not.

If D30 (i.e. Z = 2 bit) is 0, then both link and failure information will be stored; if 1 then neither.

If D29 (i.e. Z = 4 bit) is 1 then the device (except for magnetic tape decks) will be disengaged. Magnetic tape decks are left engaged except in the case of deck fail when the deck is disengaged.

Special cases

If the code number obtained is zero, then all possible incidents on the specified device will be set, so any incident will cause a jump to Y etc.

If Y = 0 (in the 150/21 instruction) then the standard (default) action on the specified incident will be set. The standard action is to halt or suspend the job and output a message (see 5.5 and 5.8.4).

Branched Programs - see also 5.3.25 and 10.1. In this case if the incident occurs the program is also temporarily unbranched (i.e. OMP remembers the branch interlocks and switches off each branch as waiting for Branch 1) and if the link is required OMP uses Branch 1's control number as the link for the program.

The following checking takes place for the 150/21 instruction; X and Y and other registers that may be referred to must be within reservations otherwise reservation violation action will take place. The peripheral name obtained must be meaningful otherwise impermissible operand action will occur. The code number must be one permitted (see 5.5) otherwise impermissible operand action will occur.

If X is the register after the one with the 150/21 then return is to X + 1 i.e. with X = V2 + 1, return is to V2 + 2.

Note that this incident restart information will be lost by OMP if the peripheral (e.g. *MT2) is relinquished or wronged. This incident restart information is not lost if the peripheral is subsequently floated or its name is changed.

The 150/22 Instruction - Set Monitoring Peripheral

```
      e.g.     150      *SP1     0        22
```

X is a programmer's peripheral name or 0

Y-address field must be zero.

This instruction sets X as the monitoring peripheral for the job.

X must be the programmer's peripheral name of a slow output device already reserved for the job. This instruction makes sure that the allocated device is in fact engaged (by asking the operator to engage it if not). It also refills the code buffer with the standard code table for line printers and punches.

Monitoring information will come out on this output device from then onwards. If this peripheral is subsequently relinquished, then monitoring information will come out on the Flexowriter.

If X = 0 then this instruction sets the Flexowriter as the monitoring peripheral. This is the default action. Only a limited amount of monitoring information is allowed on the Flexowriter.

If X is a programmer's peripheral name for an output device which is not reserved for the job then reservation violation action will take place.

If X is a programmer’s peripheral name for an input device or a magnetic tape deck or is not a meaningful programmer's peripheral name, then impermissible operand action will take place.

150/23 Instruction - Return from Private Monitoring

```
      e.g.     150      *SIG     A197     23
```

This is usually used after monitoring some program event in Style 7. The effect is to obey the instruction whose address is y_m and set OVR according to y_s and the current state of overflow, as though monitoring on X were switched off (suppressed or ignored). It enables, for example signalled instructions to be obeyed although the program is monitoring on signals. Y is normally one less than the Y of the 150/20 instruction which set up the monitoring style.

X maybe one of the following events; these are arranged in a hierarchy (*OWN is low) such that if monitoring on one event is suppressed then so is monitoring on lower events, monitoring on higher events remaining active. The hierarchy is

*OWN

*SIG

*JUM

*FOV

*OVR

*DRU or peripheral name

*IMP

*PFP or *PFN

Thus if, for example, an instruction which is both a signal and sets OVR (where the program is monitoring OB both *SIG and *OVR in Style 7) will first jump to the *SIG routine which will end with a 150/23 with X as *SIG, and then the *OVR routine will be entered, this will end with a 150/23 with X as *OVR, so that the instruction will now be obeyed and the program will continue.

X must be one of the above events otherwise impermissible operand action will occur, in particular X=*TIM, *UNR or *URG will cause this action.

Y and y_m must be within reservations, otherwise reservation violation action will occur.

If the program is monitoring overflow, *OVR in Style 7 then a 150/23 instruction which overflows will not cause entry to the *OVR routine (if this were done the program would get into a loop).

150/24 Instruction - Start New Branch (also see 10.1)

This instruction has two main uses

```
   (a) to set up a new branch of a program
or (b) to change the control address of an existing branch.
```

Case (a)

```
      e.g.     150      2        A200     24
```

This asks OMP to set up a new branch of the program and to give this new branch the branch number X and to set its entry-point as Y, there previously not being a branch numbered X. X must be an integer in the range 2 to 7.

If the program has not been previously branched, then the current branch is set up as branch 1.

The new branch is recorded as switched off waiting for the current branch (it can be switched on by a branch interlock instruction when required)

When a new branch is set up its directory-entry is put in the high-numbered registers of the program's reserved region, the program's reserved region is thus reduced. The monitoring conditions are set to default and the branch's timer is set to 1 minute. Its jobname is composed of the original jobname together with the branch number, e.g. BLOGGS2, note that the jobname of Branch 1 is BLOGGS and that BLOGGS1 is illegal.

If there is already a total of 15 branches in the machine, then a message of the following form is output to the Flexowriter,

```
      jobname  CANNOT   BRANCH
```

and the job is halted.

Case (b)

In this case X is the number of an existing branch (X must lie between 1 and 7) and must not be the current branch. This asks OMP to set X's control address as Y and switch X off waiting for the current branch.

The instruction,

```
      150.1    X        Y        24
```

has the specification switch branch X off waiting for the current branch and put branch X's control address into Y. If branch X does not exist Y is set to zero.

If X does not lie between 1 and 7, *or* if X is the current branch then impermissible operand action will occur.

Y must be within reservations, otherwise reservation violation action will occur.

150/25 Instruction - Return, restoring Conditions

```
      e.g.     150      A199     0        25
```

X is the address of a core-store register.

Y - address field must be zero.

This instruction is associated with peripheral incident routines and with Primary Input ENTER (with a link) routine, to cause “pull up” action and in the general case to return (cf 87 instruction), restoring the program to the state it was in before the incident occurred or ENTER (with a link) was read.

If the incident occurs or ENTER (with a link) is read OMP before entering the routine, if asked to will store link information in the link address specified. Besides storing the link in the modifier half and the state of OVR in the sign bit OMP also stores other information about the state of the job in the upper half. OMP also records the job as being in a “pushed down” state. The routine is then entered.

A 150/25 causes one level of “pulling up” action. X (if not A0) is the address of the register containing the link information; OMP sets x_m as the control number and set OVR according to x_s and the current state of OVR, and then restores the program to the state it was in before the incident occurred or ENTER (with a link) was read.

The instruction

```
               150      A0       0        25
```

causes one level of “pulling up” action to take place, i.e. no jumping takes place. The instruction after this 150/25 is the next instruction to be obeyed.

If a job is already in a “pushed down” state when an incident occurs or ENTER (with a link) is read, then there is a further level of “pushing down” action. The 150/25 of the second routine will pull up one level and, if not A0, returns to the first routine etc.

____

Non branched Programs

If a peripheral incident routine or ENTER (with a link) routine (also see 5.7.3.4) is entered OMP records the program as being in a "pushed down" state, and any previous pushing down is remembered. ENTER (without a link) does not cause “push down” action and causes any previous “pushing down” to be forgotten.

____

Branched Programs (see also 10.1.9)

If a peripheral incident routine or ENTER (with a link) routine is entered OMP records the program as being in a “pushed down” state and in this case OMP temporarily unbranches the program (i.e. the branch interlocks are remembered and all branches are switched off awaiting Branch 1). If already "pushed down" OMP remembers this.

If the link for the program is required then Branch 1's control number is used. The routine is considered as being Branch 1. The 150/25 of the first routine will restore the branch interlocks and return to the main program. The control number for the current branch (150/25 can appear in any branch but it is usually Branch 1) is set to x_m and state of OVR according to x_s and the current state of OVR. ENTER (without a link) does not cause "pushing down" action and causes OMP to switch off all branches awaiting Branch 1; any temporarily unbranched state is forgotten as is any "pushing down".

Rerun 150/10 with Y = 2 causes the program to be temporarily unbranched (i.e. the branch interlocks are remembered and all branches are switched off awaiting Branch 1).

A use of the dummy 150/25 i.e. with X = A0 is in restarts within restarts for example.

Chained Peripheral Incidents (see also 5.5)

If separate peripheral incident routines (with links) for several incidents have been specified and more than one of these incidents occur simultaneously then the routines will be chained, so that the 150/25 of one will cause entry to the next restart and so on.

X and x_m must be within reservations otherwise reservation violation action will occur.

150/30 Instruction - Reserve Peripheral.

```
      e.g.     150      *MT1     0        30
```

X is a programmer's peripheral name

Y-address field must be zero

This asks OMP to reserve the peripheral, (if not already reserved for the job) and to add this request to the list in the directory-entry, of peripherals reserved for the job.

Slow Input Peripherals and Magnetic Tape Decks

X not already reserved

If there is a device of the required type available (if not, OVR is set and the job continues by obeying the next instruction), OMP reserves the peripheral for the job as a "floating" peripheral i.e. *a* specific peripheral device is not allocated but OMP makes sure that an idle device of that type will be available when the job later requests a specific device to be allocated. A reservation message is printed on the Flexowriter

```
jobname    RESD   *MT1
```

This informs the operator of the reservation.

X already reserved

The instruction is a dummy

Slow Output Peripherals

X not already reserved

If there is a device of the required type free (if not, OVR is set and the job continues) and there is one engaged, OMP reserves peripheral for the job and allocates the specific device, and in the case of

card punches, OMP fills the code buffer with the standard code table (see 5.6.2)

line printers, OMP fills the code buffer with standard code table (see 5.6.3) and outputs a line containing 1------1 repeated 15 times.

The reservation and allocation message is output on the Flexowriter.

```
jobname  RESD     *SP1-SPB
```

This informs the operator of the reservation and allocation.

If none of the free devices of the required type are engaged OMP asks the operator to engage one; the engage message is printed on the Flexowriter.

```
jobname  ENGAGE   LPB*
```

When this is done, OMP will carry out the reservation and allocation as already described. It is possible for the operator to engage another device, type RUN and then OMP will complete the reservation and allocation

X already reserved with a specific device allocated.

OMP ensures that the allocated device is engaged, by asking the operator to engage it, if not. It also refills the code buffer with the standard table for card punches and line printers (The line with 1------1 is not output since X is already reserved.)

If X is not a meaningful programmer's peripheral name, then impermissible operand action will occur.

150/31 Instruction - Relinquish Peripheral

```
      e.g.     150      *SR1     0        31
```

X is a programmer's peripheral name

Y is an integer

This instruction asks OMP to carry out some form of relinquishing action on peripheral X which is reserved for the job.

If X is a programmer's peripheral name for a peripheral not reserved for the job or is not a programmer's peripheral name then the instruction is a dummy.

(i)   Y = 0 Relinquish (all devices)

This asks OMP to relinquish X (i.e. the record of its reservation in the directory-entry is removed and if necessary, the record of other peripherals reserved for the job moved up in the directory-entry.)

Slow Input Devices and Magnetic Tape Decks

X reserved as a “floating” peripheral

X is removed from the job's reservation. A relinquish message of the following form is printed on the Flexowriter.

```
jobname      RLQD      *SR1
```

X reserved with a specific device allocated

X is removed from the job's reservations. The specific device becomes free. The relinquish message of the following form is printed on the Flexowriter.

```
jobname      RLQD      *SR1-TRC
```

This informs the operator that TRC is now free.

OMP carries out terminating action on the specific device.

In the case of readers, OMP disengages the device.

In the case of tape decks, OMP rewinds the tape, rewrites Block 0 if necessary, disengages the deck and asks the operator to unload the deck. The unload message is of the following form.

```
ORION      MTD      UNLOAD serial-number
```

If the number (n) of errors that occurred since the tape was loaded is non-zero then ERn follows the serial-number. Also on this line may be printed | followed by various digits, whether these are printed depend on the ENGINEER directive setting (see 5.7.4.9)

Slow Output Devices

X reserved with a specific device allocated

X is removed from the job's reservations. The relinquish message informing the operator that the specific device is free, is printed on the Flexowriter.

OMP's terminating action, if the device is engaged, in the case of

7-track paper tape punches is to output Runout, 6 ER, ST, NL and Runout characters.

5-track punches is to output Runout 8 ER and Runout characters.

card punches is to output blank cards.

line printers is to output one PT, then print a line containing 1------1 repeated 15 times and then PT's

The number of blank cards and PT's output is a parameter of OMP.

Then the device is disengaged (Y=0)

(ii) Y=1 Float and Select (readers only)

In this case OMP re-reserves X as a “floating” peripheral. OMP then reads from the reader that had been allocated, as though the select button had been pressed. (see 5.7). The floated message of the following form is printed on the Flexowriter

```
jobname      FLTD      *SR1-TRB
```

This informs the operator that TRB is now idle

Before OMP can read from the device, it must be engaged; if not a message of the following form printed on the Flexowriter asks the operator to engage it,

```
jobname      ENGAGE      TRB*
```

The engage message would precede the floated message.

If X is for a magnetic tape deck or an output device reserved for the job then impermissible operand action will occur.

If X is for a slow input device reserved as a “floating” peripheral then peripheral violation action will occur.

(iii) Y=2 Relinquish and select (all devices)

Slow Input Devices

X is removed from the job's reservations. The relinquish message informing the operator which device is free is output. OMP then reads from the reader as though the select button had been pressed (see 5.7); the engage message is output if the reader is disengaged.

Magnetic Tape Decks

X is removed from the job's reservations.

The relinquish message is output. OMP then rewinds the tape on the deck and reads Block 0 (as though the engage button had been pressed)

Slow Output Devices

X is removed from the job's reservations. The relinquish message is output, on the Flexowriter.

If the device that had been allocated is engaged then OMP's terminating action as described for Y=0 does not take place and in this case (Y=2) the device is left engaged. If the device is disengaged it is left disengaged.

If X is for a reader reserved as a "floating" peripheral then peripheral violation action will occur.

(iv) Y=3 Abolish and select (readers only)

This asks OMP to abolish the job, and then to read from the reader that had been allocated, as though the select button had been pressed.

The core-store, drum-store and all peripherals reserved for the job are relinquished in the normal way (i.e. as for Y=0) The relinquish message for each and the abolished information (see 5.3.11) are output on the Flexowriter.

This instruction is intended to be used when it is required to have a number of jobs (to be run consecutively) on one paper tape reader.

If X is for a magnetic tape deck or an output device reserved for the job then impermissible operand action will occur.

If X is for a slow input device reserved as a "floating" peripheral then peripheral then peripheral violation action will occur.

Y > 3 causes impermissible operand action.

On Orion 2

Y=2 (Relinquish and Select) and Y=1 (Float and Select) lose the extracode buffer

Y=3 (Abolish and Select) does not lose the extracode buffer.

The 150/32 Instruction - Get Geographical Name

```
      e.g.     150      *SR1     A10      32
```

X is a programmers peripheral name
Y is the address of a core-store register.

This instruction puts into Y the geographical name of the device whose programmer’s name is X.

The geographical name of a device consists of three letters - the first two identify the type and the third the particular device e.g. a tape deck may have geographical name MTG. Any message to the operator should refer to devices by geographical names as the operator will not know in general which devices are called by which programmers' names.

The k-bits of a device are an 8-bit quantity which is recognised as the address of the device by the logic of the machine - the 140 on Orion 1 (extracode on Orion 2) does a table search to find the k-bits corresponding to a particular programmer’s name. On Orion 1 the k-bits are checked for odd parity. On Orion 2 the k-bits are not and are sequential.

If X is the name for a peripheral allocated to the program, then 3 characters of the geographical name (in standard 6-bit code) are put into the l.s. 18 bits of Y and the k-bit are put into the last 8 bits of the upper half of Y (i.e. into D16 to D23) - the rest of Y is left clear.

If X is a floating peripheral then y’ = 0

If X does not belong to the program then y’ = - 1.0

Y must be within reservations otherwise reservation violation occurs.

150/33 Instruction - Load Document

see also 5.7.2.3

```
e.g.     150      *SR2     A100     33
```

X is a programmer’s peripheral name or 1 or 2
Y is the address of a core-store register or 0 ≤ Y ≤ 63

This instruction ensures that a peripheral is reserved for the job; the device that is allocated being loaded with the specified document.

Y as the address of a core-store register.

If Y is the address of a core-store register then the 8 registers Y to Y+7 contain either a document name (when X is for a slow output device) or a document request name (when X is for an input device or a tape deck).

Each register stores a component. The permissible characters of a component are letters, digits and .(point). A null component is allowed which is stored as a clear word. The component + (plus) which must not be the first or second component means that the document is composite (see 6.1). Characters in a word must be right - justified (i.e. SP characters in a word must only be non-significant left-hand characters.)

When Y to Y+7 contain a document request name then a word which must not be the first or second, containing the character - minus (the other 7 being non-significant left-hand SP characters) is taken to mean “don't care what this component is.”

Impermissible characters in Y to Y+7 will cause impermissible operand action. Y to Y+7 (or Y to Y+8 when X=1 or 2) not within reservations will cause reservation violation action. X not a meaningful programmer’s peripheral name or not 1 or 2 will cause impermissible operand action.

a) Slow Input Devices (i.e. X is a programmer’s peripheral name for a paper tape reader or card reader)

Y is the address of a core-store register

The 8 registers Y to Y+7, contain a document request name. If there is a suitable device, OMP searches idle devices for this document (i.e. OMP looks for a document whose document name “fits” the document request name (see flow chart on page 8)). When the document is found then the device on which the document is loaded, is allocated to the job.

OMP then copies the name of the document into Y to Y+7, so that words that did contain - (minus) are now filled in.

X not already reserved

```
e.g.     150      *SR1     RQNAM1   33
```

If there is no device of the required type free, then the job is halted awaiting space on this 150/33 instruction. A message of the following form is printed on the Flexowriter:-

```
jobname  NO       SR
```

This informs the operator that a free 7-track reader is not available. When any space becomes free, the job continues by obeying this 150/33 instruction again.

If there is a device free, OMP searches idle devices of the required type for the requested document. If it is not loaded, then the job is halted awaiting document on this 150/33 instruction and a message of the following form is printed on the Flexowriter:-

```
jobname  LOAD     SR       document request name *
```

This asks the operator to load the document on an idle 7-track reader (an idle device is one not allocated). When any document is loaded, the job continues by obeying this 150/33 instruction again.

Having searched devices of the required type and having found the requested document, then OMP reserves the peripheral X for the job. The device is allocated and the document name copied into Y to Y+7 as already described. A message of the following form is printed on the Flexowriter:-

```
jobname  RESD     *SR1-TRC
```

This informs the operator of the reservation and allocation; in this example that, TRC is allocated to the job.

X already reserved as a floating peripheral

```
150      *CR1     RQNAM2   33
```

In this case there will be a device of the required type available and so OMP searches idle devices of this type for the requested document. If the document is not loaded then, the load message will be printed as already described. When the document is found, its name is copied into Y to Y+7 and the device is allocated to the job. A message of the following form is printed on the Flexowriter:-

```
jobname  *CR2-CRA
```

This informs the operator that CRA is now allocated to the job.

X already reserved and a specific device already allocated to the job

```
e.g.     150      *SR1     RQNAM3   33
```

In this case X is temporarily re-reserved as a "floating peripheral" and the specific device which had been allocated becomes an idle device. A message of the following form is printed on the Flexowriter:-

```
jobname  FLTD     *SR1-TRC
```

This informs the operator that TRC is now idle. The instruction now continues as for X already reserved as a floating peripheral which has already been described.

When a device becomes idle, OMP carries out appropriate terminating action and disengages the device.

b) Magnetic Tape Decks (i.e. X is a programmer's peripheral name for a magnetic tape deck)

(i) Y as the address of a core-store register

In this case the tape is being "asked for by name" and the instruction has similar effects as for slow input devices i.e. Y to Y+7 contain a document request name; when the document is found, its name is copied into Y to Y+7 and the deck is allocated to the job.

X not already reserved

If there is no deck free, then the job is halted awaiting space and the message printed. If there is a free device, but the document is not loaded then the job is halted awaiting document and the load message is printed. Having found the requested document, then OMP reserves X for the job and allocates the deck and copies the document name into Y to Y+7. The reserved with allocation message is printed.

X already reserved as a floating peripheral

In this case, there will be a deck available and so OMP searches idle decks for the document. If the document is not loaded, the load message is printed as already described. Having found the document, OMP allocates the deck to the job and copies the document name into Y to Y+7 and prints the allocation message.

X already reserved and a specific deck already allocated to the job

In this case X is temporarily re-reserved as a "floating" peripheral and the specific deck which had been allocated becomes an idle deck. The floated message is printed informing the operator which deck is now idle. The instruction now continues as for X already reserved as a floating peripheral which has been described.

When a deck becomes idle, OMP rewinds the tape, updates Block 0 (see 5.3.40 and 5.3.4l) if necessary, disengages the deck and asks the operator to unload the deck. The unload message is of the following form

```
ORION    MTD      UNLOAD   serial-number
```

If the number (n) of errors that occurred since the tape was loaded is non zero then ERn follows the serial-number. Also on this line may be printed | followed by various digits - whether these are printed depends on the ENGINEER directive’s setting (see 5.7.4.9).

Thus the deck on which the requested document is loaded, will be allocated to the job but in some cases the job will not be allowed to continue unless the deck is isolated. Block 0 (see 5.3.40 and 5.3.4l) contains many items of information, one of which is the document name of the tape and others are a Date and the write permit bit (D0) and the date control bit (D24). In the following cases, the deck must be isolated:-

D0=0 and D24=0 and Date in D1 to D23 not reached

D0=0 and D24=1 and Date in D1 to D23 not reached

In the above two cases, if the deck is not isolated, OMP halts the job and outputs a message on the Flexowriter asking the operator to isolate the deck and then cause the job to continue by typing RUN directive (see 5.7.3.3) The message is of the following form:-

```
jobname  INHIBIT WRITING ON MTA AND RUN
```

The operator must press the write inhibit switch so that it is on, and then type RUN directive.

If the write permit bit (D0) is 1 or the Date is reached then the job is allowed to continue whether the deck is isolated or not.

(ii) Y as an integer, $0 \leq Y \leq 63$

```
e.g.     150      *MT1     0       33
```

In this case, the request is for a deck loaded with a scratch tape and the instruction has the same effect as for the cases when a document request name is given, except that OMP searches idle non-isolated decks for a scratch tape.

A scratch tape is a tape loaded on a non-isolated deck and whose word in Block 0 containing the Date, write permit and date control bits has

D24=0 and Date in D1 to D23 reached.

OMP searches only non-isolated decks for a scratch tape.

If Y =0 then a deck loaded with a scratch tape of any length will be allocated. If 1 ≤ Y ≤ 63 then a deck loaded with a tape of nominal length Y hundred feet will be allocated. Block 0 contains the nominal length of the tape.

The load message for a scratch tape is of the following form

```
jobname  LOAD     MT       SCR *
```

i.e. SCR is printed instead of the document request name.

If Y ≠ 0 then Y is printed between MT and SCR

(c) Slow Output Devices (i.e. X is a programmer's peripheral name for a paper tape punch, line printer or card punch or Flexowriter type (Turitz only))

If OVR is set then writing with overflow set action will take place

Y is the address of a core-store register

The 8 registers Y to Y+7, contain a document name. This document name is output on the engaged allocated device. Each component is separated by solidus though non-significant right-hand null components and corresponding solidi are not output.

On paper tape punches, two lines are output; Runout NL (CR LF on 5-track) DOCUMENT followed by the document name NL (CR LF on 5-track) Runout

On line printers, two lines are output; one is blank and the next one contains DOCUMENT followed by the document name, one line is fed.

On card punches, two cards are output; one is blank and the next one contains DOCUMENT followed by the document name.

X not already reserved.

If there is no device of the required type free, then the job is halted awaiting space and a message of the following form is printed on the Flexowriter:-

```
jobname  NO       LP
```

Otherwise, OMP looks for a free device of the required type. It looks for an engaged one first; if there isn't an engaged one, it looks for a disengaged one and then asks the operator to engage it, by printing a message of the following form on the Flexowriter:-

```
jobname  ENGAGE   SPB*
```

When an engaged device is found, X is reserved for the job and a specific device allocated.

In the case of paper tape punches the two lines, one with the document name are output.

In the case of cord punches, the code buffer is filled with the standard code table (see 5.6.2) and the two cards, one with the document name are output.

In the case of line printers, the code buffer is filled with the standard code table (see 5.6.3) and the two lines, one with the document name are output. These two lines (because X is being reserved for the first time) will follow a line containing the combination 1------1 repeated 15 times - this is to show whether any hammers are failing to fire.

The allocation completed, then a message of the following form is printed on the Flexowriter.

```
jobname  RESD     *CP1-CPA
```

<u>X already reserved and a specific device already allocated.</u>

In this case, OMP asks the operator to engage the specific device, if not already engaged. Then, in the case of paper tape punches, it outputs the two lines already described.

In the case of card punches, it fills the code buffer with the standard code table and outputs the two cards, already described.

In the case of line printers, it fills the code buffer with the standard code table and outputs the two lines, already described. (Since X is already reserved the line with 1------1 is not output).

<u>X is integer 1</u> (For slow input and decks)

```
e.g.     150      1        A100     33
```

Y is the address of a core-store register. The 8 registers, Y to Y+7 contain a document request name. The register, Y+8 contains in the l.s. 5 bits an integer n ($0 \le n \le 31$)

This instruction asks OMP to search all idle devices (except slow output devices) for the requested document, and the effect of the instruction is similar to that already described for slow input devices and magnetic tape decks. The order in which devices are searched may be specified for each installation.

If the document is not loaded then the job is halted awaiting document and a load message of the following form is printed on the Flexowriter.

```
jobname     LOAD  Document Request Name *
```

When the document is found, OMP notes on which type of device the document is loaded and puts into Y+8, in the l.s. 15 bits, a programmer's peripheral name which is appropriate to that type of device; the number in this name being n. (e.g. Y+8 would contain *FRn, if the requested document were found on an idle 5-track reader). This peripheral is reserved, if not already, and the device allocated to the job. The name of the document is copied into Y to Y+7 and the message of this reservation and allocation is printed on the Flexowriter.

X is integer 2 (For slow output)

Y is the address of a core-store register. The 8 registers, Y to Y+7 contain a document name. The register, Y+8, contains, in the l.s. 5 bits an integer n ($0 \leq n \leq 31$).

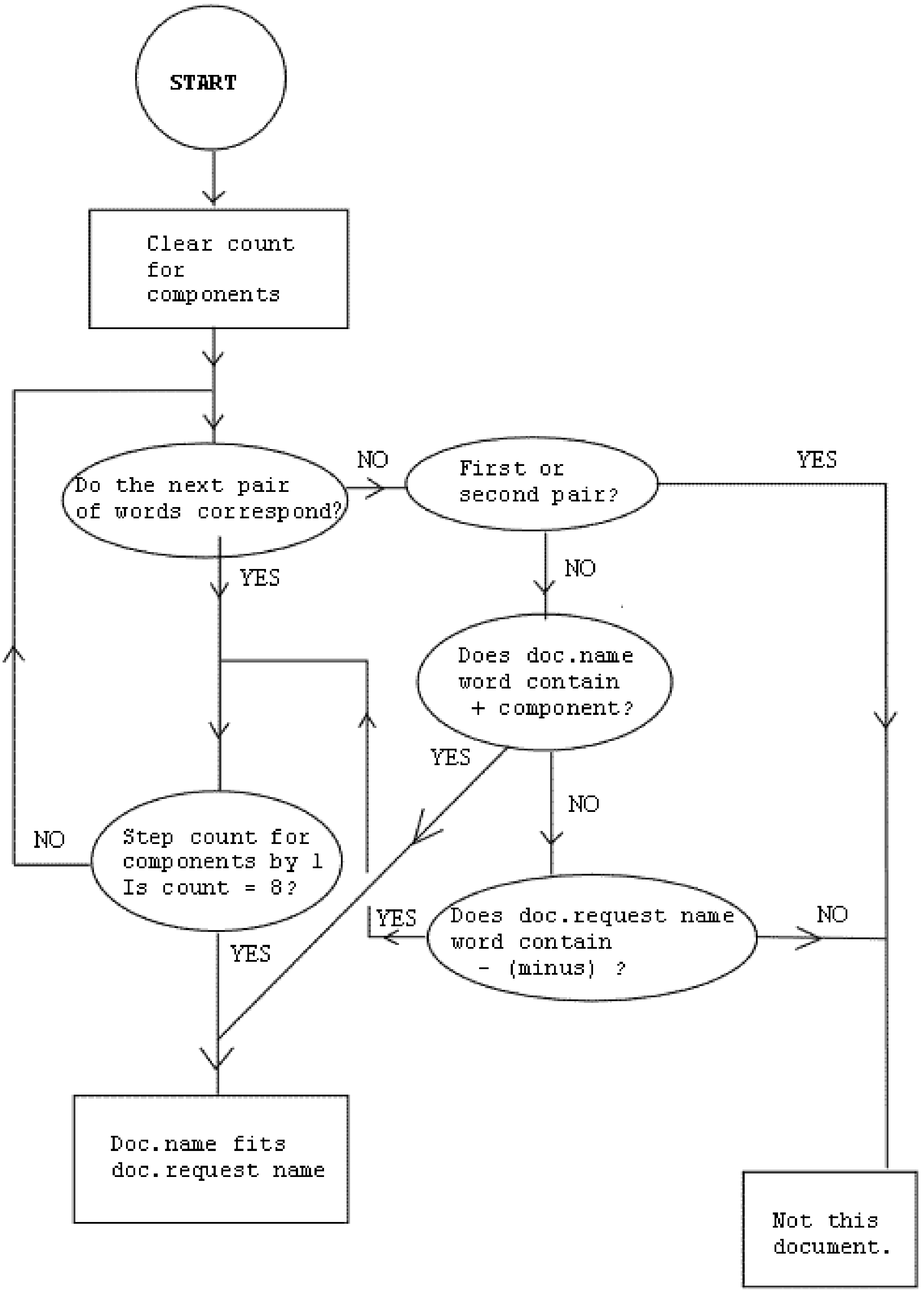
This instruction asks OMP to find a free slow output device. The order of searching may be specified for each installation.

If no free device is available then the job is halted awaiting space and a message of the following form is printed on the Flexowriter.

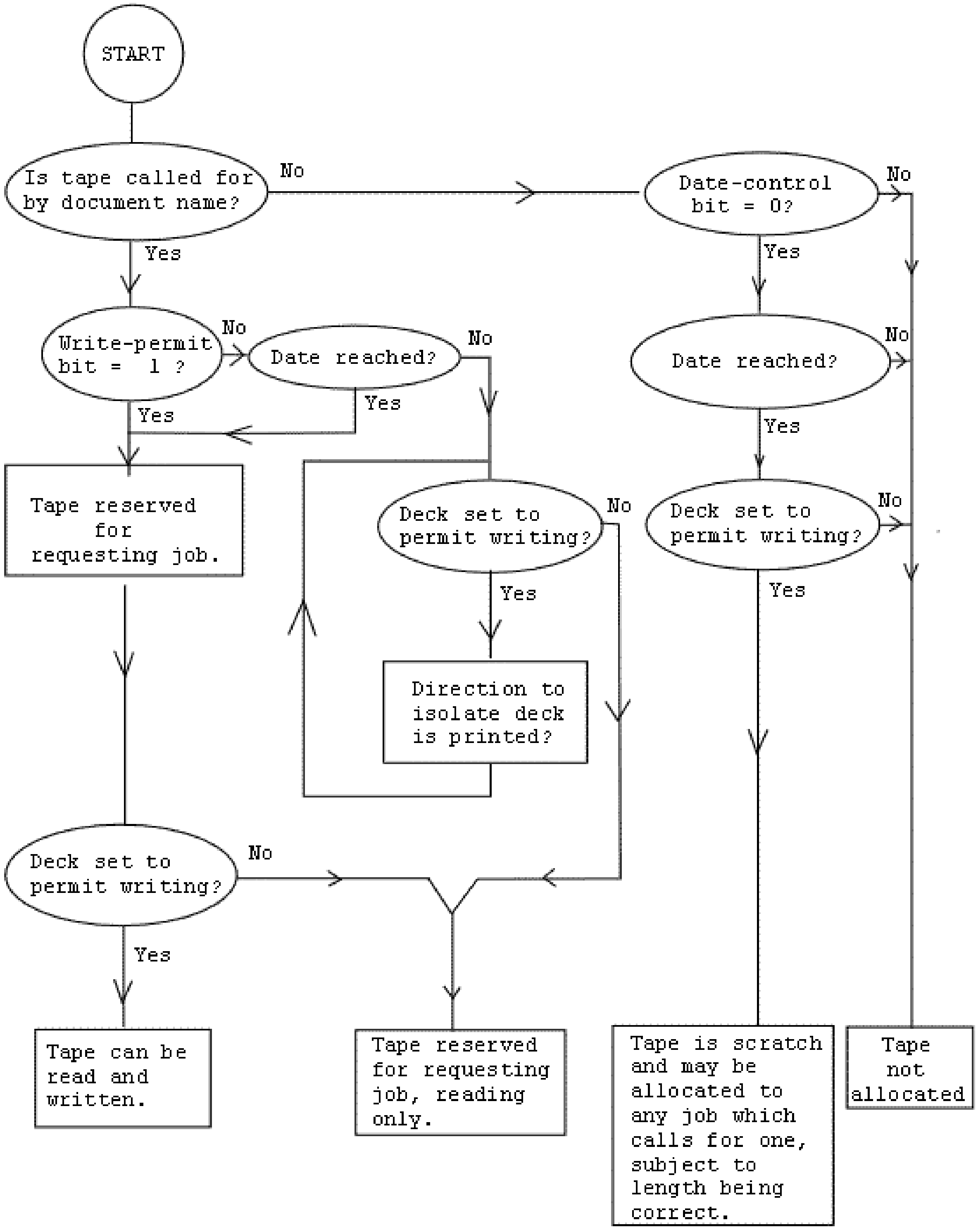
```
jobname     NO    OP
```

OMP finds the first free device that is engaged, if possible, otherwise one that is disengaged and asks the operator to engage it. OMP then puts into Y+8, in the l.s. 15 bits a programmer's peripheral name which is appropriate to that type of device; the number in this name being n. This peripheral is reserved and the specific device allocated. The document name in Y to Y+7 is output on the device. In the case of paper tape punches, the two lines, already described are output. In the case of card punches, the code buffer is filled and the two cards, already described are output. In the case of line printers, the code buffer is filled and the three lines, already described are output. A message is printed on the Flexowriter informing the operator of the reservation and allocation.

Flow Chart for document name "fitting" document request name.


START
Clear count for components
Do the next pair of words correspond?
NO
First or second pair?
YES
YES
NO
Does doc.name word contain + component?
YES
NO
NO
Step count for components by 1 Is count = 8?
YES
Does doc.request name word contain - (minus) ?
NO
YES
Doc.name fits doc.request name
Not this document.

150/33 Reservation of Magnetic Tapes


START
Is tape called for by document name?
No
Yes
Date-control bit = 0?
No
Yes
Write-permit bit = 1 ?
No
Yes
Date reached?
No
Yes
Date reached?
No
Yes
Tape reserved for requesting job.
Deck set to permit writing?
No
Yes
Deck set to permit writing?
No
Yes
Direction to isolate deck is printed?
Deck set to permit writing?
No
Yes
Tape can be read and written.
Tape reserved for requesting job, reading only.
Tape is scratch and may be allocated to any job which calls for one, subject to length being correct.
Tape not allocated

On a low installation, tapes with valuable high density information will not be allocated.

150/34 Instruction - Get Document if Loaded

X is a programmer's peripheral name or 0 or 1 or 2

Y is the address of a core-store register or $0 \leq Y \leq 63$

This instruction, (except for the case X = 0,) is similar to the 150/33 instruction, except that

(i) if there is no device free, for all types of device then the 150/34 merely sets OVR and lets the job continue

or (ii) if the requested document or scratch tape (this applies to input device magnetic tape decks) is not loaded then the 150/34 merely sets OVR and lets the job continue.

Special Case

X = 0

Y is the address of a core-store register.

This instruction checks that the 8 registers Y to Y+7, contain permissible characters for a document request name. If Y to Y+7 contain only legal characters then the instruction is a dummy, otherwise impermissible operand action will take place.

150/35 Instruction - Get or Request Document

X is a programmer's peripheral name or 0 or 1 or 2

Y is the address of a core-store register or $0 \leq Y \leq 63$

This instruction (except for the case X=0) is similar to the 150/33 instruction, except that

if the requested document or scratch tape (this applies to input devices magnetic tape decks) is not loaded then the 150/35 prints the load message, sets OVR and lets the job continue.

Special Case

X=0

Y is the address of a core-store register

This instruction prints the load message, asking the operator to load the requested document (Y to Y+7 containing the document request name). The job is then halted awaiting document on the next instruction.

The 150/36 Instruction - Change Name

```
      150      *SR1     *SR5     36
```

X is a programmer's peripheral name.

Y is a programmer's peripheral name.

This instruction says “the peripheral device which was formerly called X will now be called Y".

X must be a programmer’s peripheral name for a peripheral reserved (floating or allocated) for the job. Y must be a programmer’s peripheral name for a device of the same type as X, i.e. X and Y must be the same except for the last 5 bits. A message of the following form is printed on the Flexowriter.

```
jobname     *SR1 NOW *SR5
```

Styles of monitoring or failures on this device are unaltered by the change of name.

If X and Y are not for the same type of peripheral device then impermissible operand action will take place. Impermissible operand if job already has device called Y.

150/40 Instruction - Get Document Name etc.

also see 5.7.2.3

X is a programmer’s peripheral name.
Y is the address of a core-store register.

Slow Input Devices

```
      e.g.     150      *SR1     A100     40
```

When a document is loaded, OMP stores the document name in its drum working space and notes on which device this document is loaded. If this device is allocated to the job and the job obeys a 150/40 for this device, then OMP gives the document name into the 8 registers Y to Y+7. Characters are right-justified.

If this instruction is obeyed for a device which has been allocated to the job by ALLOCATE directive or by JOB or RERUN followed by RESERVE THIS directive then the 8 words will not be sensible.

Magnetic Tape Decks

```
      e.g.     150      *MT1     A100     40
```

When a tape is loaded OMP stores the information in Block 0 of the tape in its own drum working space. If this tape is allocated to a job, then the job may “re-write” Block 0 (150/41, 150/44, 150/43). This new Block 0 information is stored in this drum working space and if the job obeys a 150/40 then OMP gives the information from the drum (i.e. the information obtained is the information last given to OMP for this tape).

The 11 registers Y to Y+10 will contain,

Y will contain the Date etc

D0 is the write permit bit.

D1 to D23 give the Date.

D1 to D11 give the year (e.g. value 1963 for 1963)
D12 to D15 give month (e.g. value 2 for February)
D16 to D20 give the date (e.g. value 5 for 5th of the month)
D21 to D23 are copied.

D24 is the date control bit.

D25   On a low density machine will be zero whatever the bit on the tape itself. On a high density machine D25 will be the inverse of the bit on the tape itself and D25 will be 0 to mean high density and 1 to mean low density.

D33 to D47 are usually zero unless the tape is pre-addressed in which case this field is an integer. OMP uses this as the block address of the last pre-addressed block on the tape. When a write fail occurs between Block 0 and this block OMP will repeat the writing in situ (see 4.7.6.6.2)

Y+l to Y+8 will contain the document name

The 8 registers contain the components right-justified.

Y+9 will contain non-sequential block information

The modifier half is the “position” of the last written non-sequential block. This is an integer which is one greater than the block address of the block on the tape, before the last written non-sequential block.

Nothing should be assumed about the rest of Y+9.

Y+10 will contain the length and serial-number

D18 to D23 give the nominal length of the tape in hundreds of feet. This is 36 for full length tapes.

The modifier half is an integer which is the serial-number of the tape.

Nothing should be assumed about the rest of Y+10.

X must be for an input device or tape deck otherwise impermissible operand action will occur.

If X is for an input device or tape deck reserved as a “floating” peripheral then peripheral violation action will occur.

Y to Y+7 for slow input devices and Y to Y+10 for tape decks, must be within reservations otherwise reservation violation action will occur.

150/41 Instruction - Write Block 0 Unconditionally

```
e.g.     150     *MT5     A200     41
```

X is a programmer's peripheral name.

Y is the address of a core-store register.

This provides OMP with Block 0 information required on the tape.

Y contains the Date etc, required

D0 is the write permit bit

Dl to D23 give the Date

Dl to Dll give the year. (e.g. value 1966 for year 1966)
D12 to D15 give the month (e.g. value 1 for January)
D16 to D20 give the date (e.g. value 25 for 25th of month)
D2l to D23 are copied.

This Date should be the date on and after which the information on the tape is no longer required.

D24 is the date control bit. If it is 0, then the Date is a void date, if it is 1, then it is a write permit date.

D25. On a low density machine this corresponding bit on the tape itself will always be zero whatever the bit is in the core. On a high density machine, a high density job will set this bit to 1 to inform OMP that the information on this tape is to be written in low density (see 4.7.33 for more details); 0 means write in high. This is CB (core bit).

D26 to D32 should be zero

D33 to D47 should be zero, unless the tape is pre-addressed in which case these bits are the block address of the last pre-addressed block on the tape.

Y+1 to Y+8 contain the document name required (see 6.1)

The 150/41 causes OMP to store this required Block 0 information in its drum working space and asks OMP to re-write Block 0 of this tape as soon as it is convenient for OMP to do so. Thus if a 150/41 is obeyed after the 150/33, 34 or 35 or after rewind (i.e. when the tape is in the load position between blocks 0 and 1) then Block 0 is written straight away, otherwise Block 0 is written, when the tape is next rewound.

When Block 0 is actually written onto the tape, OMP outputs a message on the Flexowriter. The printing of the items in brackets is optional; only those items which have been changed since Block 0 was last written are printed.

```
jobname    MTg    Serial-number    (Date)    (NSn)    (PAm)  (Ll)
          (Document name)
```

Date printing gives the Date (e.g. 10.2.1966 means the date field is 10, the month field is 2 and the year field is 1966 thus 10th Feb 1966), and the setting of the write permit bit (P printed before the Date means it is 1) and the date control bit (P after the Date means it is 1) and the date control bit (P after the Date means it is 1) and the density (H printed after the Date (and P, if present) means that the information on the tape is in high density).

NSn - n is the position of the last written non-sequential block.
PAm - m is the number of pre-addressed blocks.
Ll - l is the nominal length in hundreds of feet.

For example

```
jobname    MTD   125    1.2.1966P    NS7
           A/B/C/D
```

X must be a programmer's peripheral name for a tape deck otherwise impermissible operand action will occur. X must not be reserved as a "floating" peripheral, otherwise peripheral violation action will occur.

Y to Y+8 must be within core-store reservations otherwise reservation violation action will occur.

Y+1 to Y+8 must contain only legal characters for a document name (see 6.1), otherwise impermissible operand action will occur.

If OVR is set then writing with overflow set action will occur.

If the tape deck is isolated, then peripheral incident, writing on isolated deck action will occur.

150/42 Instruction - Current Block Address.

```
     150      *MT1     A1       42
```

X is a programmer’s peripheral name.
Y is the address of a core store register.

X must be a programmer's peripheral name for a tape deck which is reserved (not as a floating peripheral) for the job, otherwise peripheral violation action will occur.

Y must be within core store reservations.

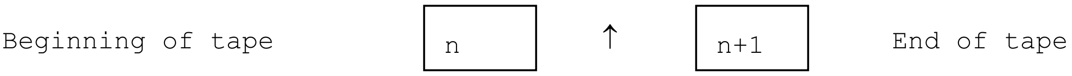
After the instruction has been obeyed then, the general case gives

y_u' = address of last block read on tape

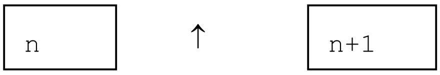
y_m' = address of next block, looking forwards on tape.

Note that y is the contents of the Block Address Register (see 4.7.2) shifted cyclically 24 places.

For the purpose of describing the information given by the 150/42 instruction then, given that the read-write heads are in the position shown by ↑, then assume that there is a block on each side of the read-write heads and that the address of the block nearer the beginning is n and that the address of the block nearer the end is n+1.



After <u>writing</u> operation

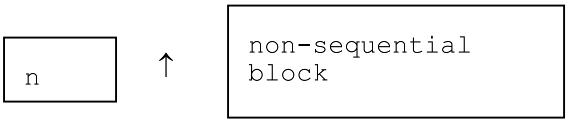


i.e. Block n has just been written. A 150/42 would give

y_u' = n

y_m' = n+1

After writing non-sequential block OMP positions the tape thus,



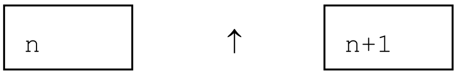
and a 150/42 would give

y_u' = n

$y_m' \neq 8191$ but n+1

After a <u>reading</u> operation

<u>Forwards</u>

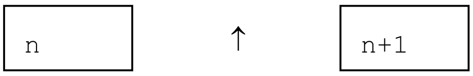


i.e. Block n has just been read forwards. A 150/42 would give

$y_u' = n$

$y_m' = n+1$

<u>Backwards</u>

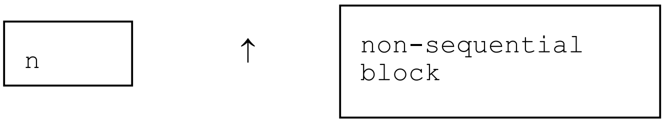


i.e. Block n+1 has just been read backwards. A 150/42 would give

$y_u' = n+1$

$y_m' = n+1$

After reading non-sequential block, OMP positions the tape thus,



A 150/42 would give

$y_u' = n+1$

$y_m' \neq 8191$ but n+1

The 150/43 Instruction - write non-sequential block

```
     e.g.     150      *MT1     0        43
```

X is a programmer’s peripheral name

Y is 0 or 1

Y=0

When writing to a tape, after writing the last information block, the program should obey a 150/43 with Y=0 to ask OMP to write a non-sequential block as the next block on the tape.

OMP writes a one word block giving it the address 8191 (if 8191 would be sequential then 8190 is used instead) and then OMP positions the tape between the last information block and the non-sequential block.

The position of the last written non-sequential block is an item of Block 0 information, so OMP stores this information in the drum working space for this tape and makes a note that Block 0 of this tape must be rewritten when the tape is next rewound.

Y=1

This asks OMP to re-write the Block 0 information as stored in its drum working space, to the tape as soon as it is convenient for OMP to do so. It does not cause a non-sequential block to be written and hence the record of the position of the last written non-sequential block is not altered.

X must be a programmer’s peripheral name for a magnetic tape deck, otherwise impermissible operand action will occur.

If X is for a tape deck reserved as a “floating” peripheral then peripheral violation will occur.

If OVR is set then writing with overflow set action will occur.

If the deck is isolated then peripheral incident, writing on isolated deck will occur.

Y greater than 1 will cause impermissible operand action.

150/44 Instruction - Write Block 0 conditionally

```
      e.g.     150      *MT2     A100     44
```

The purpose of this instruction is to ensure that Block 0 is re-written only e.g. if a non-sequential block is written. (e.g. only if the program comes to a successful conclusion). If the job is abolished before a non-sequential block is written, then Block 0 on the tape is left as it was and in the common case, the tape is left scratch, instead of having a future void date with rubbish on the tape.

The Block 0 information required on the tape is contained in Y to Y+8 (see 5.3.41)

The 150/44 causes OMP to store this required Block 0 information in its drum working space and asks OMP not to rewrite Block 0 straight away, but to re-write Block 0 only if subsequent instructions or events ask OMP to do so. Thus this Block 0 information is stored on the drum and is written to the tape only

(a) if subsequently, a non-sequential block is written on the tape or a 150/43 with Y =1 is obeyed (before X is relinquished that is)

or (b) if since Block 0 was last written, a non-sequential block has been written or a 150/41 instruction or a 150/43 with Y = 1 obeyed.

The 150/50 Instruction - Chapter Change

The chapter change instruction is usually compiled by Symbolic Input from the macro 1000. It occupies two words, the first being a 150/50 instruction - the second is not an instruction.

The X-address of the 150/50 instruction holds the working store starting address, X, and the Y-address is the length, N. The whole of the upper half of the second word holds the drum starting address, D, and the Y-address position of the second word is the entry point, J.

The effect is as if the following sequence of instructions were obeyed:

```
141.1    0        D
142      X        N
75       J        0
```

except that the drum transfer may overwrite the compound instruction. The address X and N may be pre—modified or replaced as usual; the addresses D and J may be replaced by setting the RX and RY bits respectively in the second word to 1 but they may not be premodified. The Z-address position of word 2 is used in the ENTER directive (see 5.7.3.4) but is ignored in the 150/50 instruction. Thus, the compound instruction:

```
150      (MOD1)   (MOD2)   50
00S      (MOD3)   (MOD4)   0        is equivalent to

141.1      0      (MOD3)
142      (MOD1)   (MOD2)
75       (MOD4)
```

A check is made that X, X+N-1 and J are within working store reservations and that D and D+N-1 are within drum reservations, otherwise reservation violation occurs. N=0 causes impermissible operand.

If the program is monitoring on drum references, the drum read implicit in this instruction is monitored as usual; if style 7 is used the address given is that of the 150/50 instruction, the usual return by 150/23 will work. The instruction is not regarded as a jump for the purpose of monitoring on jumps.

150/51 Instruction - Load Chapter of Semi-built-in program (SBIP)

```
      e.g.     150      3       A506     51
```

X is an integer
Y is the address of a core-store register.

This instruction asks OMP to load Chapter X of program y into the job's core-store.

If X is odd then the appropriate chapter is loaded and entered (i.e. the program's control number is set as requested by the chapter). If X is even then the appropriate chapter is only loaded and then return is to the instruction after the 150/51 unless Y is the register after the 150/51 in which case return is to Y+1 (i.e. if X=2n in 150/51 and Y=V2+1, then return is to V2+2, hence the name can always be safely put after the 150/51.) X=2n causes a chapter to be loaded whereas X=2n-1 causes the same chapter to be loaded and entered.

Basic Input and semi-built-in programs are stored on the drum outside any program's reserved drum region in a relocatable form. A chapter when required, is loaded by OMP into the job's reserved core region, the datum point being added when appropriate. Basic is stored permanently on the drum, whereas a semi-built-in program is loaded on to the drum when it is to be used and is removed when not in use if the drum space is needed.

Basic Input (see 7.1.5)

If y=0, then chapter X of Basic Input is loaded into the job's core-store. X=0 and y=0 will cause impermissible operand action.

Semi-built-in program (see 6.3)

If y≠0, then Y contains the name of a semi-built-in program. Chapter X is loaded into the job's core store.

Y contains the name e.g. PREADD, in character form right-justified, i.e. the name must conform to the rules governing a component of a document name (see 6.1) but + or - are not allowed, otherwise impermissible operand action will occur.

If X=0 and y≠0 then semi-built-in program y is regarded as no longer in use. If X=0 and the job has not been using y then impermissible operand action will occur.

If a job asks for a semi-built-in program which is not on the drum and conditions are such that it can be stored on the drum, then the job is "suspended loading a semi-built-in program". The job is temporarily given the jobname of the calling job with the character 0 (zero) appended e.g. BLOGGS0 while Basic Input loads this semi-built-in program on the drum. When finished a message on the Flexowriter informs the operator

```
      e.g.     BLOGGS0  PREADD   IN
```

The job continues normally by obeying the 150/51 again. Basic uses A64 to A255 of the job's core and so these will be overwritten. The job must have at least 256 words of core reserved otherwise reservation violation action will occur.

Note that jobs which use even-numbered chapters of SBIP'S must make sure that the SBIP is already in as otherwise the instruction following the 150/51 may be overwritten by semi-built-input (i.e. by BLOGGS0)

If the requested SBIP is not in the drum and, either the same or a different SBIP is being read, or there is not enough drum space available then the job is halted and a message on the Flexowriter informs the operator

```
      e.g.     BLOGGS0  PREADD  NOT     IN
```

If there are fewer than 2 words of drum available or there are already 8 SBIP'S then the job is halted awaiting space, the message on the Flexowriter being

```
               BLOGGS   (not BLOGGS0)   NO SPACE
```

The job continues when the reason for the stoppage has been removed. Directives allowed for BLOGGS0 are PRINTOUT, DIRENT, TIME, HALT, RUN, ABOLISH and ABANDON; the others are not allowed.

Reservation violation action will occur if Y is not within core reservations or if the job's core is not large enough for the chapter requested.

The 150/52 Instruction - Change Drum Reservations

```
              e.g.     150      A100     A1       52
```

The modifier half of X is the number of words of drum required for the job.

If the requested amount of drum, x_m is available then the drum reservations for the job are changed; a message is printed on the Flexowriter

```
              e.g.     jobname  DRUM     100
```

and the amount of drum reserved, x_u, is put into Y.

If the requested amount of drum as not available then OVR is set and the maximum amount available is put into Y (i.e. y will be less than x_m). The programmer can then decide if it is worth trying to compromise on the drum space available.

If the length of the reserved region is being shortened the contents of the reserved region remaining will be unaltered. If it is being lengthened the contents of the old reserved region will be unaltered and the additional space will be clear.

The minimum amount of drum that a job can have is 2 words, so that if x_m = 0 or 1, then 2 words will be reserved.

When a drum reservation is made other programs may be moved about, as all programs are relocatable on the drum.

X and Y must be within reservations otherwise reservation violation action will occur.

The 150/53 Instruction - Change Core Store Reservations.

This instruction will, depending on the setting of Y, increase or decrease core store reservations in one of three ways if the requested space is available. In all cases X is the number of words of core store required; it may be larger or smaller than the current reservations.

a) General Case

Often an increase in reservations will not be available without a change in the location of the program within the core store. In this case the program may be relocated with a different datum point. Y is the address of a core store register which, after the old and the prospective datum points have been added in, is within the old and the prospective new reservation respectively; if it is not reservation violation will occur. If the requested amount is available the new datum point is put into Y. The new core length and the new datum point (if changed) will be output on the monitoring peripheral or the flexowriter in the form:-

```
BLOGGS       CORE    4096    512
```

the second figure being the datum point.

If the amount requested is not available the amount that is available is put into Y and OVR is set. Whether or not the amount of core requested in X is available, monitoring conditions are returned to standard; monitoring on *OWN will remain as before, so the starting address for the *OWN routine is checked to be within the proposed new reservations, if it is not reservation violation will occur. The instruction will normally be followed by a jump if OVR instruction and a 150/51 instruction so that Basic Input or the semi-built-in program can enter the correctly relativised chapter.

b) Same Datum Point Case

In this case the change in core store reservations, if possible, is carried out with same datum point. Y is to be A0 in the 150 instruction. If the amount is available, the new core length is output on the monitoring peripheral or flexowriter. If it is not available OVR is set. Monitoring conditions are not set to standard; *OWN monitoring is still checked to be within the proposed new reservations, else reservation violation will occur. However, the addresses of any style 7 restarts are not checked, so care must be taken that any such subroutines occur within the new reservations.

c) Different Datum Point Case

In this case Y is set to an integer, and the instruction will attempt to reserve X words of core store with datum point 64Y. If it is available the new core length and datum point will output on the monitoring peripheral or flexowriter. OVR will be set if the space is not available. If 64Y is greater than the core size, reservation violation will occur. In this case all monitoring is set to standard and any *OWN addresses are checked.

In all cases OVR is set if the amount requested is not available. Thus it is advisable to follow the 150/53 instruction with a jump if OVR instruction. All three cases may be used with branched programs.

The 150/54 Instruction - Semi-built-in Program In

This instruction may only be obeyed by the chapter of basic input which stores semi-built-in programs on the drum.

Any other program attempting to obey a 150/54 instruction will be suspended due to illegal instruction.

5.4.1 Dealing with Own Programming Failures

There is a special fast facility to enable programmers to deal partially with their own program failures. This facility is mainly intended to be used by NEBULA and other compilers; it is not recommended that ordinary programmers should use it unless they are very experienced. Only style 0 (off - standard) and style 7 are available - if the facility is on, any program failure will cause a jump to the specified address, with the address of the offending instruction (and state of OVR) in the preceding register. The program should then see if the failure is of a type it recognises - if it is it takes appropriate action, otherwise it should return by a 150/23 instruction.

In order to make the facility fast the monitor program is not able to decode the type of failure. It may be any of the following:

(a) Illegal instruction i.e. 35-37, 46, 47, 96, 105-107, 113, 127-137, 147, 151-157. Disengage for tape deck.

(b) Illegal two-address form i.e. 125, 144-146, 150.

(c) Impermissible operand in the instructions 40-45, 90-95, 97, 100, 101, 103, 121, 125, 140, 141, 142, 143.

(d) Isolated 140, 141, 142 or string longer than 13.

(e) Reservation violation in working or drum store or peripherals.

(f) Replaced or pre-modified 150 instructions - these will be obeyed correctly after the 150/23.

(g) Writing to drum or peripheral with OVR set.

(h) Signals, successful jumps, overflow, floating-point overflow, drum and peripheral transfers if these events are being monitored.

(i) Rewind on magnetic tape (Mode 14)

Note that illegal operands or Z-address in 150 instructions will not be picked up at this stage and the program cannot therefore deal with these itself.

The facility is switched on or off by a 150/20 instruction with X-address *OWN.

N.B. The style 7 routine to deal with program failures must not contain any program failures. The monitor program temporarily regards premodified or replaced 150 instructions as program failures and for this reason they must not be contained in *OWN style 7 routines.

Contents 5.5 Peripheral Incidents

5.5 Peripheral Incidents (see 5.5.21 and 5.3.25)

There are a number of events or incidents occurring on peripheral devices which cause an interruption and entry into OMP.

The possible events are described in this section. Some are given code numbers for use with the 150/21 instruction (see 5.3.21) for specifying a restart should an event occur.

Default Action

If no restart has been specified and an incident occurs OMP carries out default action. This is, in general to halt (suspend in the case of magnetic tape) - though in some cases the job is allowed to continue, to disengage the device (except for magnetic tape when it is left engaged unless the event is deck fail when it is disengaged) and to print a message (see 5.8.4) on the Flexowriter e.g.

```
      jobname TRB HALTED PARITY FAIL
```

The default action for each device and incident is given in the appropriate sub-section.

Note that the programmer can control in some cases for slow devices whether the restart is entered with the device engaged or not and also can control whether the incident message is printed and also whether the link and information word are to be given - this is done by the setting of specific bits with the 150/21 instruction. Magnetic tape restarts are entered with device engaged, except for deck fail. If several incidents occur simultaneously then restarts (with links) are chained in Code number sequence; if one incident causes the device to be disengaged then it will be before any restart is entered.

Information Word

When specifying the restart, the programmer may indicate to OMP whether the link (into Y - 1) and the information word (into Y - 2) are to be given should the event occur (see 5.3.21).

The bit by bit information word describing the state of the state of the device may be obtained at any time by obeying a 140.13/142 instruction pair (see 3.14). This bit by bit information is given here and in the appropriate subsection for each type of device.

Bits of the Information Word.

D9 to D23 (X-address field) for fast devices, (magnetic tape) gives the starting address of transfer. For slow devices this is the current core-store address reached by the transfer when it failed, if possible. On Orion 2 it will be close to the current address, for example for paper tape devices it may be the beginning of the current chunk.

D33 to D47 (Y-address field) give the finishing address of the transfer.

Bits D0 to D7 and D24 to D31 are used to give information about the type of failure and the state of various staticisors in the device. Some of these bits will not cause programmer's action but they are, for the sake of completeness, given in the appropriate sub-section. A bit is 1 if the event has occurred or the device is in that state. Note that Orion 2, does not have address failure.

5.5.1 Paper Tape Devices

Code 1 Wrong parity character on 7-track when read on a reader or when sent to a punch. It is not possible to determine exactly which character has failed; restart procedure for a reader might involve operator action in pulling the tape back, for example. PARITY FAIL

Code 2 Tape low in punch. OPERATOR*

Code 3 Address failure. (Only Orion 1) ADDRESS FAIL*

Code 4 Device disabled. The device is disengaged whether the programmer has asked for this or not. DISABLED*

Code 5 Wrong mode. This is an attempt to use a 5-track mode on a 7-track device or vice versa. WRONG MODE

Default Action

For events with Codes 1, 3, 4, 5 the default action is to halt the program, print a message (see 5.8.4.) and to disengage the device.

For event with Code 2 the default action is to print the message, to disengage the device and allow the program to continue.

Bits in the Information Word

D0 Wrong parity character.

D1 Attempt to use five-track mode on a seven-track device or vice versa.

D2 Device disabled.

D3 Device disengaged.

D4 Device set to read five-track tape.

D5 Address failure. (Only Orion 1)

D6 Select (reader) or tape low (punch)

5.5.2 Card Readers

For other information which may be of use see 4.4.7.

Code 1 Failure of read check (Misread) READ CHECK FAIL*

Code 2 Illegal punching in modes 1,2 or 4 ILLEGAL PUNCHING

Code 3 Engineer's attention required because of card misfeed or wreck. The reader is disengaged whether the programmer has asked for this or not DISABLED*

Code 4 Operator's attention required because hopper below 200 cards without weight in place or stacker full or reject pocket full etc. OPERATOR*

Code 5 Address failure. ADDRESS FAIL*

Default Action

For events with codes 1,2,3, and 5 the default action is to halt the program, to print a message and to disengage the device.

For event with Code 4 the action is to print a message, disengage the device and to allow the program to continue. For hopper empty this default action may not be adequate; it may be advisable to have a Code 4 restart.

Bits in the Information Word

D0 address failure. Only Orion 1.

D1 Operator's attention - hopper low, stacker full etc.

D2 Engineer's attention - card misfeed, wreck

D3 Disengaged

D4 Illegal punching in modes 1, 2 or 4

D5 Switched to 65 columns

D6 Select button

D7 Failure of read check

D24 to D30 (Only Orion 1) are used to find where the failure of a read check occurred. (4 columns are stored per word)

D24 1st column
D25 2nd column
D26 3rd column
D27 4th column
D28 Rows 5, 4, 10, 11
D29 Rows 7, 6, 1, 0
D30 Rows 8, 9, 2, 3

D31 Interstage On Orion 2 the interstage button must be pressed to read interstage.

5.5.3 Card Punches and Line Printers

For other information which may be of use see 4.5.7 and 4.5.8.

Code 1 Operator's attention required because of cards low or hopper empty or stacker full (Punch) or Paper low (Printer) OPERATOR *

Code 2 Buffer overflow (i.e.- more than 120 characters) BUFFER OVR

Code 3 Checksum failure. This indicates the contents of either the code or data buffer has been lost - it is not possible to determine which. When this occurs OMP fills the code buffer with the standard code table whether a restart has been specified or not. This failure may be overcome by repeating the transfer. BUFFER FAIL *

Code 4 Address failure (Orion 1 only) ADDRESS FAIL *

Code 5 Card wreck (Punch only) CARD WRECK *

Code 6 Failure of read back check - at the read station. (Punch only) READ BACK FAIL *

Code 7 Device disabled. The device is disengaged whether the programmer has asked for this or not. DISABLED *

Default Action

For event with Code 1 the action is to print a message (see 5.8.4) to disengage the device and to allow the program to continue.

For events with Codes 2 to 7 the action is to halt the program, to print a message and disengage the device.

Bits in the Information Word

D0 Failure of read back check.

D2 Device disabled

D3 Device disengaged

D4 Checksum failure (Orion 1 only)

D5 Address failure (Orion 1 only)

D6 Card wreck

D27 Operator’s attention - cards low, paper low, etc.

D31 Buffer overflow

5.5.4. Magnetic Tape Decks. (see 4.7.6)

Code 1 Repeated read fail READ FAIL

Code 2 Repeated write fail WRITE FAIL

Code 3 Writing on isolated deck WRITE INHIBIT

Code 4 Writing after reading backwards. Note that a restart may be specified but it will never be entered as OMP does the necessary repositioning and the block is written.

Code 5 End of tape END OP TAPE

Code 6 Failure to write last block LAST FAIL

Code 7 Deck failure DECK FAIL.
In this case the deck is disengaged. When the deck becomes idle, on pressing the engage button OMP rewinds the tape and unloading is requested. A restart for this can only relinquish the deck.

Code 8 First fail FAIL
If this restart has been specified OMP does not attempt the repeated read or write sequence but enters this restart without repositioning the tape - i.e. it "leaves you past the incorrect block". If no restart has been specified code 1 or 2 action occurs.

Default Action

For events 1, 2, 3, 5 and 6 OMP suspends the program, prints a message and leaves the deck engaged. For code 7, the difference is that the deck is disengaged.

Bits in the Information Word

D0 Orion 1 - this is Address failure
Orion 2 - this is parity failure when reading or writing from tape.

D1 Address discrepancy low (reading) or address discrepancy (writing).

D2 Writing on isolated deck or after reading backwards.

D3 Orion 1 - checksum failure in first adder (writing) or address discrepancy high (reading). Orion 2. - Address high(reading)

D4 Orion 1 - checksum failure in second adder
Orion 2 - checksum failure.

D5 First sensing post - i.e. near end of tape.

D6 Deck interrupt - produced by various disabled conditions.

D7 Engage button.

D24   Writing. On Orion 2 D24=0 and D25=1 when erasing

D25   Writing short gap mode.

D26,27 Values 0-3 for tape control being used.

D28   Tape moving forward in last transfer.

D29   Deck engaged - note that the significance of this bit is the reverse of D3 in other devices.

D30   Writing permitted

D31   Both sensing posts.

5.5.5 Drums

Some failures of the drum store are treated as failures of the central computer and cause all programs to be abandoned (see 5.8.4.1). There are no events which are programmers' option.

Bits in the Information Word

D0 parity failure in core store (on writing transfers), and current address parity failures.

D1 1 for writing.

D2 Drum address parity failure.

D3 Write parity failure.

D4 Read parity failure.

D6 Drum not at correct speed (Orion 2 only).

With a 141.13, 154 instruction pair (but not with a 141.13, 142 pair) a second word is obtained, which is written into X+1, containing the drum address. This drum address is relative to the start of the drum control specified in the 141 instruction - thus bits 0-3 of the drum address are not obtained (since they are already known). Bits 4-6 are always zero, bits 7-9 specify the drum, bits 10-16 the track and bits 17-23 the sector within the track. The 17 bits obtained are written into X+1 as given below (the rest of X+1 is cleared):

Bit of Drum Address	Bit in store (4096 word store)	Bit in store (8192 word store)	Bit in store (12288 or 16384 word store)	Bit in store (Store > 16384 words)
7	12	11	10	9
8	13	12	11	10
9	14	13	12	11
10	15	14	13	12
11	16	15	14	13
12	17	16	15	14
13	18	17	16	15
14	19	18	17	16
15	20	19	18	17
16	21	20	19	18
17	42	41	40	39
18	41	40	39	38
19	40	39	38	37
20	39	38	37	36
21	38	37	36	35
22	37	36	35	34
23	36	35	34	33

For Orion 2, the drum address is given in unscrambled form at the l.s. end of the word.

5.5.6 Hough-Powell Device

Code 1

A11 events

Bits in the Information Word

DO	Parity failure between DDC and Orion
D1	End of frame
D2	Buffer overflow
B3	Disabled
D4	Last transfer outward
D5	Current address parity failure
D6	Accept (select)
D?	Enable

5.5.7 I.B.M. Tapes

Code 1

All events

Bits in the Information Word

D0 Parity failure (addresses or in core store)

D1 Noise fault

D2 Reading forwards after writing or writing on isolated deck

D3 Zero character (writing BCD) or File mark (reading)

D4 Tape parity failure

D5 Load point (i.e. beginning of tape)

D6 Immediate interrupt

D7 Accept button

D24,25 - 00 for reading, 01 for erasing, 10 for write binary, 11 for write BCD

D26,27 - Control

D28 - 1 for last transfer writing

D29 - 1 for ready (engaged)

D30 - 1 for high density

D31 - End of tape.

5.5.8 Electro-data Units

Code 1

All events

Bits in the Information Word

D0 Wrong parity character read from the tape.

D1 Wrong Mode (e.g. mode 2 on *EAn).

D2 There is a switch on the deck called auto rewind. If this is set then if the end of tape is reached the tape is automatically rewound and there is a peripheral incident with D2=1. If the deck is interrogated while rewinding D2 will also be 1.

D3 1 for device disengaged.

D4 Always 1 on *EAn, always 0 on others.

D5 Address failure (address in core store not on tape).

D6 This event occurs if address information is found while reading data (i.e. *EAn). This event will always produce D0 as well as D6. It can be a hardware fault but it can also be a program error due (for example) to giving two read data transfers without an intermediate read address transfer.

5.5.9 Ericsson Key Boards and Flexowriters Type C

For the purpose of section 5.5 Ericsson Key Boards and Flexowriters type C are considered as seven track paper tape readers and seven track paper tape punches respectively. (See 5.5.1 for detailed information).

Contents 5.6 Codes and Tables

The input routines, Primary (OMP's), Basic and Symbolic accept information in character form from paper tape and cards. The characters to be input in the corresponding code are converted by these routines into characters in the Standard Internal Code given in 5.6.1

The code accepted from paper tape is the Flexowriter Code for 7-track and the Pegasus/Mercury/Sirius Code for 5-track (see CS 308 for both codes). When Basic and Symbolic routines and OMP output to paper tape then the output is in the corresponding code.

The code accepted from cards is the Standard Card Code given in 5.6.2. When Basic and Symbolic routines and OMP output to cards then the output is in this code (see 5.6.2)

Basic and Symbolic routines and OMP also output to line printers (the standard code table in the code buffer, see 5.6.3)

Output to magnetic tape and drum is in standard internal code.

5.6.1 Standard Internal 6-bit Code

Primary, Basic and Symbolic Input Routines convert input characters into this standard internal code.

Special OMP 150 instructions which operate on characters in this code are those with Z equal to 12, 13, 14, 32, 33, 34, 35, 40, 41 and 51.

Internal Code	Character	Internal Code	Character
0	Space	33	A
1	Asterisk *	34	B
2	Newline NL	35	C
3	Left bracket [	36	D
4	Right bracket]	37	E
5	Equals =	38	F
6	Apostrophe ’	39	G
7	Dummy	40	H
8	Left parenthesis (	41	I
9	Right parenthesis)	42	J
10	Less than <	43	K
11	Greater than >	44	L
12	Colon :	45	M
13	Vertical bar |	46	N
14	Error	47	O
15	Solidus /	48	P
16	Zero 0	49	Q
17	1	50	R
18	2	51	S
19	3	52	T
20	4	53	U
21	5	54	V
22	6	55	W
23	7	56	X
24	8	57	Y
25	9	58	Z
26	Character 10	59	Percent %
27	Character 11	60	Pound £
28	Character ½	61	Query ?
29	Plus +	62	Ampersand &
30	Minus -	63	Erase
31	Point .		
32	Comma ,		

5.6.2 Standard Card Code

The card code below is that in which all cards to be read by Primary, Basic and Symbolic Input Routines must be punched. These routines convert the input characters into standard internal code.

Note that for input, alternative punchings (those marked with an asterisk) are allowed for certain characters.

When Basic and Symbolic routines and OMP output to cards then for those characters with alternative punchings for input, the punching not marked with an asterisk is output.

Internal Code	Character	Punching	Internal Code	Character	Punching
		Blank			
0	Space	Column	31	Point	10,3,8
1	Asterisk	11,4,8	32	Comma	0,3,8
3	Left bracket	11,2,8	33	A	10,1
3	Left bracket	*11,7,8	34	B	10,2
4	Right "	7,8	35	C	10,3
5	Equals	0,6,8	36	D	10,4
5	Equals	* 3,8	37	E	10,5
6	Apostrophe	* 10,6,8	38	F	10,6
6	Apostrophe	* 4,8	39	G	10,7
8	Left		40	H	10,8
	parenthesis	5,8	41	I	10,9
8	Left		42	J	11,1
	parenthesis	*0,4,8	43	K	11,2
9	Right		44	L	11,3
	parenthesis	6,8	45	M	11,4
9	Right		46	N	11,5
	parenthesis	* 10,4,8	47	O	11,6
10	Less than	11,6,8	48	P	11,7
11	Greater than	11,5,8	49	Q	11,8
12	Colon	10,5,8	50	R	11,9
13	Vertical bar	10,7,8	51	S	0,2
13	Vertical bar	* 11,3,8	52	T	0,3
15	Solidus	0,1	53	U	0,4
16	0	0	54	V	o,5
17	1	1	55	W	0,6
18	2	2	56	X	0,7
19	3	3	57	Y	0,8
20	4	4	58	Z	0,9
21	5	5	60	£	0,2,8
22	6	6	61	Query	0,5,8
23	7	7			
24	8	8			
25	9	9			
26	10	* 10,0			
27	11	* 11,0			
29	Plus	10			
29	Plus	* 10,2,8			
30	Minus	11			

5.6.2.1 Standard Card Code Table

When a card punch is reserved OMP fills the code buffer with the standard card code table. OMP then outputs (see 5.3.33 (c)) the specified document name preceded by the directive DOCUMENT.

When there has been a check-sum failure in the buffer store then OMP fills the code buffer with the standard code table. (If a card punch is being used as the monitoring peripheral OMP assumes that the code buffer is filled with the standard table and if it is not then output will be unintelligible).

The standard card code table is:

```
+12.13.29.31.33.34.35.36
+37.38.39.40.41.63.1.3
+10.11.30.42.43.44.45.46
+47.48.49,50.63.5.15.16
+32.51.52.53.54.55.56,57
+58.60.61.63.15.17.33.42
+63.3.18.34.43.51.60.63
+19.31.32.35.44.52.63.1
+20.36.45.53.63.8.11.12
+21.37.46.54.61.63.5.9
+10.22.38.47.55.63.4.13
+23.39.48.56.63.1.3.4
+5.8.9.10.11.12.13.24
+31.32.40.49.57.60.61.63
+25.41.50.58.63...
```

5.6.3 Standard Printer Codes

There are depending on the installation standard printer codes for each make of printer and print barrel used. At the time of writing there are four print barrels for the I.C.T. 665 printer and two print barrels for the Anelex printer. It is not possible to have printers with different barrels at an installation.

For each character on a print barrel there is a corresponding internal code representation. The standard internal code (5.6.1) has been used as far as possible but a few small changes have had to be made because for the line printers the internal values 3 and 4 are reserved for paper throw and tabulate respectively and so the characters [and] have been given different internal values. Also some of the characters on a print barrel are not in the standard internal code. For example the character ‰ (internal value 9) in the Gothenburg code is called per mille.

The Anelex printer has 60 characters on the print barrel but only 59 can be coded.

In the Anelex standard and Swedish barrel the character omitted is underline.

Standard Printer Codes

Internal Code	ICT Commercial	ICT Scientific	ICT Swedish (Gothenburg)	ICT Swedish (Stockholm)	Anelex Standard	Anelex Swedish	Internal Code	ICT Commercial	ICT Scientific	ICT Swedish (Gothenburg)	ICT Swedish (Stockholm)	Anelex Standard	Anelex Swedish
0	Space (fixed by hardware)						32	,	,		,	,	,
1	*	*	x	*	*	*	33	A	A	A	A	A	A
2	New line (fixed by hardware)						34	B	B	B	B	B	B
3	Paper Throw (fixed by h/w)						35	C	C	C	C	C	C
4	Tabulate (fixed by hardware)						36	D	D	D	D	D	D
5		=		=	=	=	37	E	E	E	E	E	E
6		'			'	□	38	F	F	F	F	F	F
7					[	[	39	G	G	G	G	G	G
8	(	(	%	(	(	(	40	H	H	H	H	H	H
9	)	)	°/oo	)	)	)	41	I	I	I	I	I	I
10		<			<	<	42	J	J	J	J	J	J
11		>			>	>	43	K	K	K	K	K	K
12				:	:	:	44	L	L	L	L	L	L
13					|	|	45	M	M	M	M	M	M
14					]	]	46	N	N	N	N	N	N
15	/	/	/	/	/	/	47	O	O	O	O	O	O
16	0	0	0	0	0	0	48	P	P	P	P	P	P
17	1	1	1	1	1	1	49	Q	Q	Q	Q	Q	Q
18	2	2	2	2	2	2	50	R	R	R	R	R	R
19	3	3	3	3	3	3	51	S	S	S	S	S	S
20	4	4	4	4	4	4	52	T	T	T	T	T	T
21	5	5	5	5	5	5	53	U	U	U	U	U	U
22	6	6	6	6	6	6	54	V	V	V	V	V	V
23	7	7	7	7	7	7	55	W	W	W	W	W	W
24	8	8	8	8	8	8	56	X	X	X	X	X	X
25	9	9	9	9	9	9	57	Y	Y	Y	Y	Y	Y
26	10		¼		10	?	58	Z	Z	Z	Z	Z	Z
27	11		¾		11	&	59	%		Å	Å	%	Å
28	½		½	%	½	%	60	£	π	Ä	Ä	£	Ä
29	+	+	+	+	+	+	61		?	Ö	Ö	?	Ö
30	-	-	-	-	-	-	62	&		&		&	Ü
31	.	.	.	.	.	.	63	Unusable by hardware					

In ICT Swedish (Gothenburg) x (internal value 1) is multiplication sign

In ICT Swedish (Gothenburg) °/oo (internal value 9) is per mille

In Anelex Swedish □ (internal value 6) is lozenge.

5.6.3.1 Standard Printer Code Tables

When a line printer is reserved OMP fills the code buffer with the appropriate standard printer code table. OMP then outputs the specified document name preceded by the directive DOCUMENT. See 5.3.33(c).

When there has been a check-sum failure in the buffer store then OMP fills the code buffer with the standard table and so if the program was using a non-standard table then the restart action will need to refill the code buffer with the required code table.

If a line printer is being used as the monitoring peripheral OMP assumes that the standard code table is in the code buffer and if it is not then the output will be unintelligible. Similarly if a program is run at another installation and if the code tables provided are not the same then the output will be unintelligible Each type of print barrel has been assigned a number (see Section 14)

I.C.T. 665 Commercial

```
+16.63.28.63.17.63.30.63
+18.63.29.63.19.63.31.63
+20.63.32.63.21.63.60.63
+22.63.59.63.23.63.8.63
+24.63.9.63.25.63.1.63
+26.63.62.63.27.63.15.63
+33.63.34.63.35.63.36.63
+37.63.38.63.39.63.40.63
+33.63.34.63.35.63.36.63
+41.63.42.63.43.63.44.63
+45.63.46.63.47.63.48.63
+49.63.50.63.51.63.52.63
+53.63.54.63.55.63.56.63
+57.63.58.63....
```

I.C.T. 665 Swedish (Gothenburg)

```
+16.63.26.63.17.63.28.63
+18.63.27.63.19.63.30.63
+20.63.29.63.21.63.31.63
+22.63.1.63.23.63.8.63
+24.63.9.63.25.63.62.63
+15.63.33.63.34.63.35.63
+36.63.37.63.38.63.39.63
+40.63.41.63.42.63.43.63
+36.63.37.63.38.63.39.63
+44.63.45.63.46.63.47.63
+48.63.49.63.50.63.51.63
+52.63.53.63.54.63.55.63
+56.63.57.63.58.63.59.63
+60.63.61.63....
```

I.C.T. 665 Scientific

```
+16.63.5.63.17.63.30.63
+18.63.29.63.19.63.31.63
+20.63.32.63.21.63.60.63
+22.63.6.63.23.63.8.63
+24.63.9.63.25.63.1.63
+10.63.61.63.11.63.15.63
+33.63.34.63.35.63.36.63
+37.63.38.63.39.63.40.63
+33.63.34.63.35.63.36.63
+41.63.42.63.43.63.44.63
+45.63.46.63.47.63.48.63
+49.63.50.63.51.63.52.63
+53.63.54.63.55.63.56.63
+57.63.58.63....
```

Anelex Standard

```
+16.63.28.63.17.63.30.63
+18.63.29.63.19.63.31.63
+20.63.60.63.21.63.62.63
+22.63.8.63.23.63.9.63
+24.63.1.63.25.63.15.63
+26.63.27.63.33.63.34.63
+35.63.36.63.37.63.38.63
+39.63.40.63.41.63.42.63
+43.63.44.63.45.63.46.63
+47.63.48.63.49.63.50.63
+51.63.52.63.53.63.54.63
+55.63.56.63.57.63.58.63
+12.63.6.63.7.63.14.63
+10.63.11.63.5.63..63
+13.63.61.63.32.63.59.63
```

I.C.T. 665 Swedish (Stockholm)

```
+16.63.1.63.17.63.12.63
+18.63.32.63.19.63.30.63
+20.63.29.63.21.63.31.63
+22.63.5.63.23.63.28.63
+24.63.8.63.25.63.9.63
+15.63.33.63.34.63.35.63
+36.63.37.63.38.63.39.63
+40.63.41.63.42.63.43.63
+36.63.37.63.38.63.39.63
+44.63.45.63.46.63.47.63
+48.63.49.63.50.63.51.63
+52.63.53.63.54.63.55.63
+56.63.57.63.58.63.59.63
+60.63.61.63....
```

Anelex Swedish

```
+16.63.12.63.17.63.30.63
+18.63.29.63.19.63.31.63
+20.63.32.63.21.63.27.63
+22.63.59.63.23.63.60.63
+24.63.1.63.25.63.15.63
+61.63.62.63.33.63.34.63
+35.63.36.63.37.63.38.63
+39.63.40.63.41.63.42.63
+43.63.44.63.45.63.46.63
+47.63.48.63.49.63.50.63
+51.63.52.63.53.63.54.63
+55.63.56.63.57.63.58.63
+8.63.9.63.7.63.14.63
+10.63.11.63.5.63.0.63
+13.63.26.63.6.63.28.63
```

5.6.4 6-bit Internal Code of Characters from 5-track

Primary, Basic and Symbolic Input Routines convert 5-track input character into this internal code.

Internal Code	Character	Internal Code	Character
0	SP	33	A
1	*	34	B
2	LF	35	C
5	=	36	D
8	(	37	E
9	)	38	F
11	>	39	G
12	≠	40	H
13	→	41	I
14	≥	42	J
15	/	43	K
16	0	44	L
17	1	45	M
18	2	46	N and n
19	3	47	O
20	4	48	P
21	5	49	Q
22	6	50	R
23	7	51	S
24	8	52	T
25	9	53	U
29	+	54	V and v
30	-	55	W
31	.	56	X and x
32	,	57	Y
		58	Z
		60	£
		61	?
		63	ER

Contents 5.7 Primary Input

5.7 Primary Input

Primary Input is OMP's input routine. It is called in by pressing the select button on the Flexowriter or on a paper tape or card reader. It can also be called in because a job has obeyed a 150/31 with Y ≠ 0 instruction (see 5.3.31). It accepts only directives and is not responsible for program or data input. There are 3 types of directives.

(a) Directives which are not preceded by a jobname (see 5.7.2). They may not appear on the Flexowriter.

(b) Directives which are preceded by a jobname - referring to a particular job already in the machine, (see 5.7.3). Except for ABANDON and ALLOCATE, which can only occur on the Flexowriter, these may appear on either a reader or the Flexowriter.

(c) Directives which do not refer to a particular job. These are preceded by the conventional jobname ORION (see 5.7.4).

Note that on Orion 2, pressing the Select button causes the extracode buffers to be cleared.

5.7.1 Punching Conventions for Primary Input

5.7.1.1 Flexowriter Input

Primary Input is called in or switched on by pressing the select button. The operator can then type only one line of information. A line is terminated by NL character. Redundant shift characters in a line of information are allowed. ER characters are ignored. Vertical bar character means "ignore the rest of this line". BS character with ER character may be used to correct punching mistakes. Cross parentheses mean "ignore all of this line". A blank line is ignored.

Primary Input is switched off after reading a line.

5.7.1.2 Paper Tape or Card Reader Input

Primary Input is called in or switched on by pressing the select button on a reader. If the reader is allocated to a job, then the peripheral is rereserved as a "floating peripheral" for the job, the reader then being idle. The floated message is printed on the Flexowriter. The operators may use the ALLOCATE directive (see 5.7.3.8) to allocate a specific reader for the job.

It can also be called in because a job has obeyed a 150/31 with Y≠O (see 5.3.31).

Primary Input then reads from the reader.

For 7-track input, the 7-track code characters are allowed (see 4.3.3). A line should end in UC shift and is terminated by NL character. Redundant shift characters in a line of information are allowed. ER characters are ignored. Vertical bar character means "ignore the rest of this line". BS character ER character may be used to correct punching mistakes. Cross parentheses mean "ignore all of this line".

For 5-track input, the 5-track code characters are allowed (see 4.3.3). A line is terminated by CR LF characters. Redundant shift characters in a line of information are not allowed. ER characters are ignored. → character means "ignore rest of this line".

For card input, the standard code is used (see 5.6.2). One card is one line. Vertical bar character means "ignore rest of this line".

Depending on the type of directive in the line, Primary Input is either switched off after reading the line, or it goes back to read another line containing a directive. Blank lines (i.e. lines with no printing characters except ER characters, if present) are ignored and Primary Input goes on reading until a line of information is read.

5.7.1.3 Presumptive Jobnames

Lines containing directives which refer to a job already in the machine (see 5.7.3) effectively must begin with the jobname. If several of these directives (5.7.3) are to be given for the same job, then the jobname need be punched before the first of these only, the other directives being preceded by VS. VS replaces the jobname. Thus when Primary Input reads on a reader or .the Flexowriter a line whose first character is VS and contains a directive of this type (see 5.7.3), then the jobname assumed is the jobname last read, if a reader and is the jobname last read or output if the Flexowriter.

5.7.1.4 Error action by Primary Input

If there is an error in a line of information, OMP prints a message on the Flexowriter informing the operator of the error. If the incorrect line was read on a paper tape or card reader then Primary Input is switched off and the reader is disengaged. The incorrect line of information is forgotten.

A syntactical error (i.e. an error in the punching or format of the line) produces a message which tells the operator in which field of the line the error occurred. In general each line of information contains more than one field; fields are separated by VS or solidus. FIELD 1 is the first field of the line and so on. Other useful information such as the directive, the incorrect character and on which device the incorrect line was read may be given. For example, the incorrect line

```
            BLOGGS   PRI      AE½      Al

            would produce, if read on TRA

            BLOGGS   ERROR    PRI      CHAR½    FIELD 3  TRA
```

Assuming no job BLIGGS in the machine then

```
            BLIGGS   AB0

            would produce if read on TRA

            ORION    ERROR    FIELD 1  TRA
```

The incorrect line

```
            DOC      AB/ED/EF)GH/JK

            would produce if read on TRB

            ORION    ERROR    DOC      FIELD 4  TRB
```

Semantic error. OMP may not be able to implement a correctly punched line. e.g. no device is available or reservations for that job would be violated etc., the error report is for RES.VIOL. or IMP.OPER. The directive and the device on which the line was read is also given. For example

```
            BLOGGS   OUT      *MT1
```

would produce if read on the Flexowriter

```
            BLOGGS   OUT      IMP.OPER FLX
```

since a magnetic tape deck is not allowed as a monitoring peripheral.

The line

```
            BLOGGS   MON      *JUM     A5000
```

would produce if A5000 were outside reservations

```
            BLOGGS   MON      RES.VIOL TRA
```

Semantic errors in ORION directives (see 5.7.4) give a message such as

```
            ORION    ERROR    Directive  Reason FLX
```

Reason can be one of DOC NAME, INHIBIT or STATE. For example

```
            ORION    ERROR    VOI      DOC NAME FLX
```

means that the document name given with the VOIDDATE directive is not the same as that on the tape.

Peripheral Incidents

If a peripheral incident, e.g. parity fail, occurs on the reader from which OMP is reading then a message is output on the Flexowriter e.g.

```
            ORION    TRB      OMP      PARITY FAIL
```

Primary Input is switched off and the reader is disengaged. The line containing the directive is forgotten and will have to be re-input (select button used).

If the Flexowriter fails while reading a line, then ╫ NL is output and the Flexowriter is left in the select state so that the line can be repunched by the operator.

[╫ is used in this transcription to represent crossed brackets produced by outputting) BS (as this symbol is unavailable - Ed]

5.7.2 Directives not preceded by a jobname

These directives which are not preceded by a job name may be regarded as directives concerned with the operating system. They are not allowed on the Flexowriter. Only the first three letters need be punched. On reading a line containing one of these directives Primary Input is switched off. VS may precede the directive.

5.7.2.1 JOB directive

```
e.g.     JOB      EDU5B    1584
```

A tape (or sequence of cards) with JOB is known as a job-tape and is for getting a new job into the machine and setting up initial conditions etc.

If the job is accepted, OMP outputs a message on the Flexowriter, e.g.

```
jobname  ACCD     1584     4096
         11.23.08
         RESD     *SR20-TRB
```

i.e. core allocated to the job is 1584 words.

i.e. datum point is 4096.

i.e. 11.23.08 is the time at which the job was accepted.

i.e. drum allocated to the job is 64 words.

and the free device on which JOB was read is allocated to the job with programmer's peripheral name having number 20.

On a high density installation the first line is followed by vertical bar and either by H to mean a high density job or by L to mean a low density job. Nothing is printed on a low installation.

Monitoring for this job is set to default, its timer to 1 minute and it starts at the top of the time sharing list, unless there are jobs of URG 2 in the machine. OMP then loads into the job's core-store, Basic Input Routine and enters it to continue reading from the device (i.e. Basic reads the next line on the job-tape).

JOB (this is FIELD 1) must be followed by VS character and then by another field, the name of the Job, and this optionally by 2 other fields, each separated by VS character.

The jobname (FIELD 2)

This is a set of up to 7 characters; only letters and digits are allowed and the last character must be a letter. An illegal jobname is one whose first three characters are either ORI, RER, DOC, JOB or END. Violation of these rules will produce an error message on the Flexowriter, e.g.

```
      ORION    ERROR    JOB      FIELD 2  TRB
```

A legal jobname but which is the name of a job already in the machine gives

```
      jobname  ERROR    JOB      FIELD 2  TRC
```

The amount of core, FIELD 3 (optional)

This field, if present, is a request for core-store. The integer is rounded up to the next higher integer of the form 64n-16, unless it is of this form already. An error for FIELD 3 will be given if the amount requested is greater than the total available to object programs.

OMP hands over this request for core to Basic. If the request is for 1008 or fewer, OMP allocates 1008 words and if it is for more, then this amount is allocated. Basic remembers the request and on reading ENTER (7.1.4.2.12) implements the request for core. (Note that if the job at any time has more than 5 peripherals reserved then one word for each beyond 5 is subtracted from the reservation, and that monitoring jumps in Style 2 reduces the reservation by 16 words etc. The directory is never reduced.)

If this field is not present, then no request is handed over to Basic, and OMP allocates 1008 words. If no RES *CORE directive is read then Basic enters the job with 1008 or 1136 words of core allocated to the job.

If at the time of reading JOB not enough core is available, OMP rejects the job; the message on the Flexowriter being e.g.

```
      jobname  REJECTED CORE
```

Datum point, FIELD 4 (optional)

This is used only if a specific datum point is required - this facility is intended to be used when a job has to be restarted at a dump point when it is in absolute form on magnetic tape. This field, if present, is an integer which is the datum point required. An error for FIELD 4 is given if it is not a multiple of 64 or if it is less than 512, or if the area of core specified by the requested datum point plus the core store requested, is greater than the last area available to object programs.

If at the time of reading JOB, the required datum point is not available, OMP rejects the job; the message on the Flexowriter being, e.g.

```
      jobname  REJECTED DATUM
```

Other reasons why OMP may reject the job are

that 64 words of drum are not available; the message being REJECTED DRUM

or that 15 jobs are already in the machine; the message being REJECTED 15 JOBS or that the device on which JOB was read is not free and cannot be allocated (i.e. OMP’s counter of free devices of this type is zero), the message being REJECTED READER.

When a job is rejected, the message is printed on the Flexowriter, the reader is disengaged, Primary Input switched off and the line with JOB is forgotten.

5.7.2.2 RERUN directive

e.g. RERUN BLOGGS

This directive is used to rerun a job which is in the machine and is suspended awaiting rerun (150/10 with Y=2, see 5.3.10).

RERUN: (FIELD 1) is followed by VS character and then by a jobname (FIELD 2); this must be the name of a job in the machine, otherwise an ERROR in FIELD 2 will be given and the reader disengaged. For branched programs. it is allowed for Branch 1 only.

If the reader on which RERUN was read, is free then OMP allocates this device to the job (the programmer’s peripheral name having the number 20); the reservation and allocation message being printed.

The core-store and drum-store reservations and other peripherals reserved for the job are the same as they were before RERUN was read.

Monitoring for this job is re-set to default and Basic is loaded into the job’s core-store to continue reading from this device. Basic overwrites, except for A3 to A12 inclusive, the job’s core-store. The job’s drum-store is preserved unless Basic is asked to overwrite it.

Chapter 1 of Basic is loaded.

5.7.2.3 DOCUMENT directive

e.g. DOCUMENT MY/DATA/OF/QUEST/5

OMP stores the name of the document in 8 registers in its own drum working space, makes a note on which device this document is loaded, and then “switches on” all jobs which are halted awaiting document. A job requesting (by a 150/33, 34 or 35) this document will be allocated this device. The job will then read the document on this device. The job can obey a 150/40 to get the document’s name.

DOCUMENT (field 1) is followed by VS character and then by the name of the document.

Document name (fields 2 to 9)

A document name has 8 components. On the line, a component is separated from the next component by solidus. Non-significant right hand null components and corresponding solidi need not be punched. A component is a set of up to 8 characters; characters allowed are letters, digits, and point. Plus (+) component means the document is composite (see 6.1). VS character may be punched in fields 2 to 9 and is ignored. The stored form of a document name is that each of the 8 registers contains one component with the characters, if less than 8, right justified.

Impermissible characters will cause an ERROR message on the Flexowriter, the reader to be disengaged and the line forgotten.

```
e.g.     ORION    ERROR    DOC      FIELD 4  TRC
```

If it is known that a document has been loaded by mistake (i.e. no current job will request the document), then the operator may type UNLOAD directive (see 5.7.4.3) and then remove the document.

5.7.2.4 END directive

The directive in 5.7.3 when read from a reader, after implementation, cause Primary Input to read further directives from the reader; this is so that a series of directives can be read from the same paper tape or cards without the need for the select button to be pressed between each. The series may be terminated by END which causes Primary Input to be switched off and the reader disengaged.

5.7.3 Directives preceded by a Jobname

```
      e.g.     jobname  PRINTOUT AE       A1       10
```

The directives refer to a job already in the machine, otherwise an error in FIELD 1 will be reported.

VS before the jobname (FIELD 1) is not allowed. VS is allowed instead of a jobname (see 5.7.1.3). The directive (FIELD 2) is separated from the jobname by VS: only the first 3 letters of the directive need be typed. If further fields are required by the directive, they are separated from it and each other by VS.

Line with these directives are allowed in the Flexowriter, paper tape reader, card reader, except for ALLOCATE and ABANDON directives which may appear on the Flexowriter only.

If a second directive for a job is typed (say on the Flexowriter) and a first directive has not been implemented, then the first directive is lost. A message of the following form is printed in the Flexowriter.

```
      jobname  PRI      LOST     TRB
```

i.e. PRINTOUT directive read on TRB has been lost and will not be implemented.

In general, if several of these directives are to be given for a job then they are input via a reader - this tape is often known as a post mortem tape. On reading a line from a reader, OMP remembers and implements the directive, and when finished, "goes back" to read the next line containing a directive. END (see 5.7.2.4) may be used to terminate such a series.

A syntactical error (see 5.7.1.4) will produce a message on the Flexowriter, stating the FIELD in which the error occurred as well as the directive, the device on which the line was read and the incorrect character.

A semantic error will produce a message on the Flexowriter, stating the directive, the device on which it was read and either RES.VIOL or IMP.OPER as the reason for the error.

Branched Programs (see 10.1)

Each branch has a number and the "jobname" for a branch has the branch number as the last character of the name, e.g. BLOGGS3 is the name for Branch 3 of the branched job BLOGGS. Note that BLOGGS means Branch 1 and the name BLOGGS1 is illegal.

The directive ANSWER, MONITOR, OUTPUTON, DIRENT (if output is to the Flexowriter) and TIME refer to that branch only.

The other directives are allowed for a branch but affect all other branches (e.g. BLOGGS3 HALT causes all branches to be halted and then BLOGGS3 RUN will cause all branches to continue). DIRENT, if output is to a device other than the Flexowriter causes information about all the branches to be output.

5.7.3.1. HALT directives

This directive halts the job. It is similar to 150/10 with X=Y=0 (see 5.3.10) except that there is no printing on the Flexowriter.

5.7.3.2. ABOLISH directive

This directive abolishes the job. (See 5.3.11 with X=0). When read on a reader, Primary Input "goes back" to read the next directive which may be JOB directive initiating a new job.

5.7.3.3. RUN directive

This directive causes a job which is halted to continue. It has something in common with the 150/23 instruction in that, if the job has been halted due to a signal or overflow, the appropriate condition is switched off until the offending instruction has been obeyed.

5.7.3.4. ENTER directive

```
e.g.                    jobname     ENTER 3
```

This directive causes a job to continue even if it is halted or suspended etc. by transferring a chapter (the ENTER sequence) from the drum and entering it. The third field of this line is an integer n. Drum addresses 2n and 2n+1 contain the chapter changing information.

Drum address 2n contains the core starting address in the X-address field (D9 to D23) and the length of the chapter in the Y-address field (D33 to D47) of this word.

Drum address 2n+1 contains the drum starting address in the upper half (D0 to D23) and the entry-point in the Y-address field (D33 to D47) of this word.

If the Z-address field (D26 to D31) of the word in 2n+1 is non-zero, then the link is stored in this accumulator; this is ENTER (with a link). If this field is zero then this ENTER (without a link).

If the job has a monitoring peripheral other than the Flexowriter, a message is output on it, when the enter sequence is entered.

```
e.g.  ENTERED     3
```

If the job is monitoring on drum transfers, then no monitoring takes place on the transfer of the enter sequence.

(a) Non-branched Programs (also see 5.3.25)

(i) ENTER (without a link) causes no "pushing down" and causes any previous "pushing down" to be forgotten.

(ii) ENTER (with a link) causes "pushing down" and any previous pushing down to be remembered. Return should be by a 150/25 rather then by an 87 instruction.

(b) Branched Programs (also see 5.3.25 and 10.1) ENTER is allowed only for Branch 1.

(i) ENTER (with link) or peripheral incident routine causes "pushing down" action, OMP also temporarily unbranches the program (interlocks are remembered and all branches are switched off awaiting Branch l). Branch 1's control number is used as the link if required. Any previous "pushing down" is remembered. The enter-sequence is entered - it being Branch 1. A 150/25 will "return to" the program, restoring the conditions.

(ii) ENTER (without link) does not cause "pushing down" and any previous "pushing down" is forgotten as is any temporarily unbranched state. The branches are then switched off awaiting Branch 1. The enter-sequence may have 150/2 or 150/24 instructions to switch on the branches again.

5.7.3.5. ANSWER directive

```
e.g.  jobname     ANSWER      YES
```

This directive is used for a job which is suspended awaiting answer, (see 5. 3 14). Unlike an immediate answer, this postponed answer can be given either via a reader or the Flexowriter.

If given on the Flexowriter, the answer must follow on the same line as the directive. If given on a reader the answer may follow on the line after.

5.7.3.6. MONITOR directive

```
e.g.                  jobname     MONITOR     *OVR        2
```

This directive used to set a monitoring style for the job (see 5.3.20 and 5.2).

The event is the third field of the line.
The style is the fourth field of the line.

5.7.3.7. OUTPUTON directive

This directive is used to set the monitoring peripheral for the job.

Case I

```
e.g.            jobname     OUTPUTON    *LP1
```

If the peripheral (it must be a slow output one) is already reserved with a specific device allocated, OMP ensures it is engaged and on printers and card punches, it refills the code buffer with the standard code table, and sets it as the monitoring peripheral.

If the peripheral is not reserved, then if there is a free device of that type (otherwise RES.VIOL error is given) OMP reserves it and allocates an engaged device of that type and sets it as the monitoring peripheral; the reservation and allocation message is printed. If there is a free device but it is disengaged, then OMP asks for it to be engaged and then implements the directive.

Case II

e.g. jobname OUTFUTON *7

If a slow output peripheral is already reserved for the job with this number (not zero), then OMP ensures it is engaged, refills the code buffer if a printer or card punch, and sets it as monitoring peripheral.

If the job has not reserved a slow output peripheral with this number (not zero), then OMP will find any free engaged output device, complete the reservation and allocation and set it as monitoring peripheral. The reservation is with the appropriate programmer's peripheral name for the device found e.g. *LP7 or *SP7 or *FP7 or *CP7

The order in which devices are chosen is specified separately for each installation. If no free output devices are engaged, the first disengaged one is found, the operator asked to engage it and then the directive is implemented. If there are no free output devices of any sort then a RES.VIOL error is given.

Case III

jobname OUTPUTON *0

This makes the job have no monitoring peripheral, other than the Flexowriter.

5.7.3.8. ALLOCATE directive

e.g. jobname ALLOCATE *LP1 LPB

This may appear only on the Flexowriter. It is used by the operator to allocate a specific engaged device (not magnetic tape) to the job. On Orion 2 it causes the extracode buffer of slow input devices to be cleared.

If the allocation can be completed and the specified device is free but disengaged, OMP asks for it to be engaged and then implements the directive.

Peripheral of FIELD 3 not reserved for the job

If the device specified in FIELD 4 is free and so can be allocated to the job, OMP completes the reservation and allocation; the message is output.

If the device specified in FIELD 4 cannot be allocated to the job, then OMP halts the job and a message giving the reason is output on the Flexowriter.

e.g. BLOGGS LPB BELONGS TO FRED

i.e. LPB is already allocated to job FRED

If the device is wrong, then the device belongs to ORION.

If the allocate directive is for an input device and the specified device does not belong to a specific job or ORION but is not free because other jobs have reserved "floating" peripherals for this type of device the message is

```
e.g.  BLOGGS      NO       TR
i.e.  no readers are free.
```

Peripheral of FIELD 3 already reserved for the job

If the device specified in FIELD 4 can be allocated to the job, then OMP allocates this device to the job, the allocation message printed on the Flexowriter, informs the operator of the new allocation. The device that had been allocated to the job becomes idle; appropriate terminating action takes place, if it was engaged and then it is disengaged.

If the device specified in FIELD 4 cannot be allocated to the job then OMP halts the job, prints a message indicating to which job the device belongs and then relinquishes the peripheral of FIELD 3 (see 5.3.31 Y=0) the relinquish message informing the operator which device is now free is printed.

5.7.3.9 PRINTOUT directive

```
e.g.     jobname     PRINTOUT    AFP      A15      12
```

This is used to output contents of a region of a job’s core or drum store on the monitoring peripheral.

If the monitoring peripheral is disengaged OMP asks the operator to engage it and then the directive is implemented.

FILED 3 specifies the styles of printing required; these are one or more of the letters **APEIFG** and at least one style must be specified. **A** means print the address, **P** the content as program, **E** as octal, **I** as integer, **F** as fraction and **G** as floating point (see 5.3.1 6 for the form of printing obtained). This field may also contain the condition X or Z. If Z is present then all printing for clear words will be suppressed. If X is present then this asks for one word, to be output to the Flexowriter only, even though the job may have a monitoring peripheral; in this case only one register and fewer than 5 styles must be specified otherwise IMP.OPER error will be given.

FIELD 4 specifies the starting address of the region. To specify a core region, the basic address of the first register is given e.g. A5. To specify a drum region, the drum address preceded by D of the first register is given e.g. D125.

FIELD 5 specifies the length of the region. An integer separated from the starting address by VS gives the length. This may be omitted if one word only is to be output. (Alternatively the address of the last register of the region may be specified in which case it is separated from the starting address by -(minus), for example instead of   A5 12 then   A5-A16 may be given)

If the job has no monitoring peripheral other than the Flexowriter, and one word and fewer than 5 styles are specified, then output will be to the Flexowriter otherwise the job is halted and the message NO MON.PER output to the Flexowriter (condition X not present).

The printing of several words of store, using PRINTOUT is carried out by printing 20 words at a time, and so if a second directive is typed before the printing of the next 20 words is carried out then the rest of the printing is abandoned and the second directive remembered and implemented. A message on the Flexowriter states that the directive has been lost.

5.7.3.10 DIRENT directive

e.g. jobname DIRENT

This directive gives information about the job's directory. (See 5.3.17)

If the job has a monitoring peripheral then the information is output on it; otherwise to the Flexowriter.

The directive DIRENT 1 causes, information to be output to the Flexowriter only, even though the job may have a monitoring peripheral.

5.7.3.11 RELINQUISH directive

e.g. jobname RELINQUISH *SR1

This directive causes the job to relinquish (see 5.3.31 with Y=0) the peripheral specified in FIELD3. The device that had been allocated becomes free, terminating action takes place if it is engaged and then it is disengaged; the relinquished message is output on the Flexowriter.

5.7.3.12 TIME directive

e.g. jobname TIME 5

This directive re-sets the job's timer to the number of minutes specified in FIELD3.

5.7.3.13 ABANDON directive

e.g. jobname ABANDON

This directive is allowed only on the Flexowriter. The log analysis program will not charge for jobs that have been abandoned. This directive causes the job to be abolished; the printing is as for the ABOLISH directive (see 5.7.3.2) except that ABAND is printed instead of ABOLD

5.7.4 Directives that do not refer to a particular job

These are used by the operator to give or obtain information about the jobs and semi-built-in programs in the machine, about the peripherals etc. rather than information about a particular job. They are known as "ORION directives". They are accepted only from the Flexowriter. As these are allowed for any job, it is necessary only to type VS. (a presumptive jobname) and ORION need never be typed. The directive then follows (the first three letters need only be typed). Other fields, if required are separated from it and each other by VS.

Errors (see 5.7.1.4) give a message on the Flexowriter; semantic errors giving

```
            ORION    ERROR    Directive Reason  FLX
```

where the reason may be one of DOC NAME, INHIBIT or STATE.

5.7.4.1 WRQNG directive

```
      e.g.     ORION    WRONG    TRB
```

FIELD 3 is the geographical name of a device. This informs OMP that this device is not in use because it is being taken out of service either for routine maintenance or because it is unserviceable, and hence the device will not be allocated to a job.

If the device has already been declared WRONG then the directive is a dummy.

If the device is allocated to a job, then the job is halted and the peripheral relinquished; the relinquished and halted messages are printed on the Flexowriter, e.g.

```
            jobname  RLQD     *SR1-TRB
                     HALTED
```

If an idle reader or tape deck is made WRONG and the counter of free devices of that type was already zero, then a job having reserved one as a "floating" peripheral has to relinquish it, but the job is not halted; the relinquished message is printed on the Flexowriter.

When the device is in service again, the RIGHT directive is used to cancel the effect of tie WRONG directive.

If a drum is not working, DRUM is given as FIELD 3. The remaining drums will have been wired up and OMP reloaded. It is accepted only if there are no jobs or SBIP's in.

5.7.4.2 RIGHT directive

```
            e.g.     ORION    RIGHT    TRB
```

FIELD 3 is the geographical name of a device. This is the converse of the WRONG directive. It restores this device to OMP's list of free devices. This is a dummy if the device is already RIGHT. It is not given for the drum since OMP will be reloaded.

5.7.4.3 UNLOAD directive

e.g. ORION UNLOAD CRA

FIELD 3 is the geographical name of an idle device on which the operator has loaded a document (See 5.7.2.3) which it is know, will not be asked for by one of the jobs at present in the machine. This causes OMP to forget the name it has stored and to disengage the device.

In the case of tape decks, OMP also rewinds the tape, updates Block 0 if necessary and outputs a message on the Flexowriter asking the operator to unload the deck (see 5.8.4.2).

ORION MTC UNLOAD serial-number

If since the tape was loaded the number (n) of errors (e.g. reading, writing incidents) have occurred is non-zero, then ERn follows the serial-number.

Special Use

On Orion 1, UNLOAD for an allocated device is an error but on Orion 2 UNLOAD is allowed for allocated slow input devices (TR and CR) in which case it sets a bit in the extracode which has the effect of clearing the extracode buffers (this bit says ignore what is in the buffers).

5.7.4.4 SPACE directive

e.g. ORION SPACE

This gives the amounts of core-store, drum-store and peripherals not allocated to object programs. It outputs on the Flexowriter up to 4 lines which are:

(i) ORION CORE followed by a list of the available lengths. 0. is printed if none is available.

(ii) DRUM followed by the amount of drum available. 0. is printed if none is available. This may be followed by another integer which is the extra amount of drum which could be made available by overwriting semi-built-in programs not in current use (see REMOVE directive 5.7.4.10). If this is zero then no printing occurs.

(iii) IDLE followed by the geographical names of the devices which are not allocated.

(iv) FLOAT followed by the types of devices which jobs have reserved as “floating” peripherals. e.g.

```
IDLE MTB MTC MTF MTG MTH
FLOAT MT MT
```

mean that 5 decks have not been allocated but that the counter of free decks is 3 as 2 have been reserved as “floating” peripherals.

Lines (iv) or both (iii) and (iv) are omitted if there are no floating or idle peripherals respectively.

5.7.4.5 NAMES directive

e.g. ORION NAMES

This lists on the Flexowriter the jobs and semi-built-in programs in the machines. The first line contains JOBS followed by the jobnames in their current order of priority. If there are more than 8 jobs, then the rest appear on a second line. The next line contains SBIP's present (8 is the maximum allowed).

```
ORION    JOBS.  SMITH CHARLIE SMITH2
         SBIPS. PREADD PUNCH
```

If there are no jobs or SBIP's then the appropriate lines are omitted. If there are neither jobs nor SBIP's then message is

```
ORION    EMPTY
```

5.7.4.6 SERIALNO directive

e.g.   ORION SERIALNO MTA 3625 36

FIELD 3 is the geographical name of the deck (it must be non-isolated otherwise semantic error INHIBIT is given) on which is loaded the tape to be serialised. This can be either

(i) a non standard tape (i.e. Block 0 is non-standard) see 5.8.4.2

or (ii) a new tape (i.e. there is no Block 0)

or (iii) a scratch tape (i.e. D24=0 and Date reached).

FIELD 4 is the new serial number (it must be in the range 0 to 16,777,215) which the tape is to be given.

FIELD 5 is the nominal length of the tape in hundreds of feet (it must not be greater than 63) which this tape is to be given.

OMP rewinds the tape, rewrites a standard Block 0 onto the tape, such that the tape is a scratch tape (date D0, D24 are zero and the components are null) and leaves the tape loaded.

5.7.4.7 PERIPHERAL directive.

```
      e.g.     ORION    PERIPHERAL        TRA
```

This is used to obtain information about a peripheral device. FIELD 3 is the geographical name of a device.

If the device is allocated to a job then OMP outputs on the Flexowriter a message saying to which job, e.g.

```
      ORION    TRA      BELONGS  TO       BLOGGS
```

(If the device has been declared WRONG then it belongs to ORION)

If the device is idle (and in the case of input devices and decks with no document loaded) then the message is, e.g.

```
      ORION    TRA      IDLE
```

For idle input devices loaded with a document the document name is given e.g.

```
      ORION    TRA      EDU/DATA/Z
```

For idle decks loaded with a document, the message gives all the Block 0 information (as described in 5.3.41), i.e.

```
      ORION    MTX      serial-number     Date  NSn  PAm  Ll
               document name
```

For example

```
      ORION    MTB      132  P5.2.1965    NS12  PA0  L36
               EDU/TAPE
```

5.7.4.8 VOIDDATE directive

```
      e.g.     ORION    VOIDDATE MTB      25.1.1266P  EDU/1/TAPE
```

This is used to change the Date and settings of the write permit and date control bits in Block 0 of a tape. FIELD 3 is the geographical name of the deck; it must be idle and non-isolated otherwise STATE or INHIBIT semantic error will be given.

FIELD 4, is the new settings of the Date and bits (D0 and D24) required. The date is punched as 3 integers separated by 2 points, e.g. 25.1.1966; the first integer is the day of the month (25 meaning 25th of the month); the second integer is the month (1 meaning January); the third is the year which may be punched as 66 or 1966 meaning year 1966). The day of the week bits are cleared. If P precedes the date then D0 is set to 1 otherwise 0 and if P follows then D24 is set to 1 otherwise 0. The density bit is not altered.

FIELDS 5 to 12, separated by solidus are the components of the document name. Only non-significant right-hand null components need not be specified.

If the tape on the deck is correctly specified OMP rewrites Block 0 giving it the new Date and then asks the operator to unload the tape.

5.7.4.9 ENGINEER directive

```
      e.g.     ORION    ENGINEER 43
```

FIELD 3 is an octal integer which is stored in a 15-bit field.

If the l.s. bit of this is 1 then extra printing on magnetic tape incidents takes place. (See 5.8.4.2)

If the "2 bit" (i.e. the second bit from l.s. end) is 1 then a peripheral incident causes the message (see 5.8.4.3) and information word in octal to be printed, even though the job may have asked for the printing of the incident message to be suppressed.

If the "16 bit" and the "8 bit" (i.e. the fifth and fourth bits from the l.s. end) are

(i) 00 then the time is printed every minute and the date every ten minutes.

(ii) 01 then the time and date are both printed every ten minutes only.

(iii) 10 then the time is printed every minute and the date is printed only at midnight.

(iv) 11 then the time is never printed and the date is printed only at midnight

Example of the time and date

```
      ORION    20.20.00 3DEC64
```

If the "32 bit" (i.e. the sixth from the l.s. end) is 1 then the punching of the log tape is suppressed.

The "64 bit" (i.e. the seventh from the l.s. end) on a low density machine is not used. On a high density machine, if this bit is 0 then a job when accepted will be a high job, whereas if this bit is 1 then a job when accepted will be a low job. The ACCD message on the Flexowriter will be followed by vertical bar and then an H or L to indicate high or low density job. In Word 5 of the program directory D31 will be 1 to indicate high density and will be 0 to indicate low. When this bit is 1 DIRENT will print an H. This bit, thus the state of the job, will remain unchanged throughout the run of the job.

5.7.4.10 REMOVE directive

```
      e.g.     ORION    REMOVE   PUNCH
```

This directive is read to remove the specified semi-built-in program from the drum so that the amount of drum occupied by it will be freed to be used for other purposes. Only if not being used may a SBIP be removed.

When a semi-built-in program is removed either as a result of a REMOVE directive, or because the drum space is needed, a message of the following form is printed on the Flexowriter.

```
            ORION    REMOVED  PUNCH
```

More than one semi-built-in program may be removed at one time, in which case more than one name will be printed.

5.8. Flexowriter Messages Output by Orion Monitor Program.

5.8.0. This section is a list of messages which may be output on the Flexowriter; for this purpose OMP has been divided into five sections.

5.8.1. Program failures and monitoring on program events.
5.8.2. Printing produced by 150 instructions.
5.8.3. Messages in response to directives.
5.8.4. Message concerned with peripheral incidents.
5.8.5. Miscellaneous messages.

In this description a message which is preceded by a jobname is indicated by (b) and one preceded by ORION by (o). This name is in fact replaced by TB character if it is the same as the previous name. The symbol (p) is used to indicate "punch on" messages, i.e. they are punched onto the log tape.

Some of the messages appear on the Flexowriter only. Some are intended to appear on the Job's monitoring peripheral, but if the job hasn't one then in some cases the message appears on the Flexowriter instead, whereas in other cases the job is halted and the message

(b) NO MON.PER

is output on the Flexowriter.

The message on the Flexowriter asking the operator to engage a specific device is output in several circumstances,

(b) ENGAGE SPB*

When this is done the job continues.

5.8.1. Messages produced on the Flexowriter - Program Events.

(b)	IMP.OPES	See 5.2.0.1
(b)	ILL.INST	5.2.0.1
(b)	W.W.OVH	5.2.0.1
(b)	RES. VIOL	5.2.0.1
(b)	PER. VIOL	5.2.0.1
(b)	JUMP SIG	5.2.1
(b)	HLTD.SIG	5.2.1
(b)	FLT.OVR	5.2.3.0
(b)	FXD.OVR	5.2.3.1.

The instruction (or string) causing the event will be output on the next line if the job has no monitoring peripheral. The job will be suspended in the first four cases and halted in the others. (In the last four cases RUN will cause the job to continue and in the case of PER.VIOL, ALLOCATE followed by RUN will cause the job to continue - not for magnetic tape).

(b) TIME UP 5.2.4

is printed if the job has exceeded the time it asked for, the job will then be halted.

Monitoring jumps in Style 2 may produce

(b) J TO FROM (see 5.2.2)

and other lines giving the last few successful jumps.

These messages are produced only when the .job has to be stopped - if a restart has been specified then the restart is entered and no message appears on the Flexowriter.

5.8.2 Messages produced on the Flexowriter by 150 instructions

150 instructions can cause the Program events described in 5.8.1 to occur.

The 150/10 (see 5.3.10) causes one of the following messages, indicating that the job is not going.

```
(1)    (b)    HALTED         n
(2}    (b)    SUSPD          n
(3)    (b)    SUSPD.RERUN    n
(4)    (b)    NO SPACE       n
(5)    (b)    NO CORE        n
(6)    (b)    NO DRUM        n
```

n denotes that an integer may be printed. In case (1) the job will continue if RUN is given but not in case (2); in (3) a Rerun tape should be loaded; in (4), (5) and (6), the program will continue when any store, or device becomes free or if RUN is given.

The 150/11 produces

```
(p)   (b)   ABOLD   y  z  n
(p)   (b)   t
```

informing the operator that the job is abolished. y is the mill time, z is the peripheral time and t the local civil time (y, z and in being in hours, minutes and seconds). n denotes an optional integer. This message may be preceded by a series of lines relinquishing all the peripherals reserved for the job and which devices are now free.

The 150/13 (see 5.3.13) may produce a line of information - it will be terminated by full stop.

150/14 (see 5.3.14) produces two lines; the first is terminated by question mark and the second is

```
(b)   ANS
```

and the Flexowriter is left in the select state so that the operator may type the answer; if not known the NL key is pressed to free the Flexowriter and the job is then suspended awaiting answer.

The 150/16 (see 5.3.16) may produce the content of one register in various formats.

The 150/17 (see 5.3.17) may produce information about the job's directory, e.g.

```
(b)      CORE 1136   DATUM 512   DRUM 640
(b)      CN A136   OVR 0
(b)      USED 0.01.51   LEFT 0.01 .53
(b)      MON 3  F
(b)      *SP1   SPB
```

The 150/20 (see 5.3.20) may produce

```
(p)      (b)  URG  n
```

where n is the new urgency.

The 150/24 (see 5.3.24) may produce

```
      (b) CANNOT BRANCH
```

if there is a total of 15 branches in the machine; the job is halted.

The 150/30, 33, 34 and 35 instructions (see 5.3.33) produce messages informing the operator of the reservations of peripherals and/or allocation of specific devices. These are for example

```
      (p)   (b)  RESD *SR2
      (p)   (b)  RESD *SP2-SPA
      (p)   (b)  *MT2-MTD
```

The message

```
      (p)   (b)  FLTD *SR3-TRA
```

informs the operator that the peripheral *SR3 is now reserved as a floating peripheral and TRA is now idle.

Messages asking the operator to do something are terminated by * and those connected with reservation of peripherals are, for example

```
      (b)   LOAD SR MY/DATA/-*
      (b)   LOAD MT 36  SCR*
      (b)   INHIBIT WRITING ON MTC AND RUN*
```

The job continues when this is done.

A message may inform the operator that no device of the required type is free e.g.

```
      (b)   NO MT
```

The job continues when a device becomes free.

The 150/31 (see 5.3.31) produces a message. The RLQD message informs the operator which peripheral has been relinquished and which device is now free

```
      (p)   (b)   RLQD *SR2
      (p)   (b)   RLQD *MT2-MTC
      (p)   (b)   FLTD *SR2-TRC
```

In some cases OMP will read from the device that had been allocated as though the select button had been pressed; in others the device is disengaged.

The 150/36 (see 5.3.36) produces

```
      (p)   (b)   *SR2 NOW *SR5
```

informing the operator that the job has given the device a new programmer's peripheral name.

The 150/51 (see 5.3.51) may produce

```
      BLOGGS0  PRINT NOT IN
      BLOGGS  NO SPACE
```

if the specified semi-built-in program cannot be loaded, the job is then halted. The message

```
      BLOGGS0  PRINT IN
```

is printed when the specified SBIP is loaded on the drum; the job continues.

The 150/5 2 (see 5.3.52) produces

```
      (p)   (b)   DRUM n
```

where n is the amount of drum now reserved for the job.

The 150/53 (see 5.3.53) produces

```
      (p)   (b)   CORE  n  m
```

where n is the amount of core now reserved, and if the datum point has been changed, m is the new datum point.

5.8.3 Messages output on the Flexowriter in response to Directives.

Errors in directives are recognised at two levels, syntactical and semantic.

A Syntactical error (see 5.7.1.4) produces a message containing ERROR and also the directive, in which field the incorrect character, and on which device the incorrect line was read, e.g.

```
      (b) or (o)  ERROR PRI CHAR½ FIELD 3 TRA
```

A semantic error (see 5.7.1.4) produces a message containing IMP.OPER or RES.VIOL and also the directive and on which device the incorrect line was read e.g.

```
      (b)   OUT   RES.VIOL FLX
```

Semantic errors in ORION directives produce a message containing ERROR and one of DOC NAME, INHIBIT and STATE as the reason for the error e.g

```
      (o)   ERROR VOI   DOC NAME FLX
```

An error causes Primary Input to be switched off and the incorrect line to be forgotten. If read on a reader it is disengaged.

If a second directive is given and a first directive has not been implemented then the first one is lost; a message is printed e.g.

```
      (b)   PRI   LOST  TRB
```

Directive JOB (see 5.7.2.1), if the job is accepted causes the ACCD information to be printed, e.g.

```
      (p)   (b)   ACCD 1584 4096
      (p)   (b)   11.23.08
      (p)   (b)   RESD *SR20-TRA
```

If the job cannot be accepted then the rejected message is given, Primary Input switched off, the reader disengaged and the line forgotten.

```
      (b) REJECTED CORE or DATUM or DRUM or READER or 15 JOBS
```

Directive ABOLISH (see 5.7.3.2) causes the ABOLD information as described for 150/11 in 5.8.2 to be printed.

Directive MONITOR (see 5.7.3.6) may cause the URG message as described for 150/20 in 5.8.2 to be output.

Directive OUTPUTON (see 5.7.3.7) may produce the reservation and allocation message, e.g. RESD *SP1-SPB if a new reservation is made.

Directive ALLOCATE (see 5.7.3.8) will produce one or more of

```
      (p)   (b)   RESD *SP2-SPB
      (p)   (b)   *CR2-CRB
      (p)   (b)   RLQD  *LP2-LPA
            (b)   LPB BELONGS TO jobname
            (b)   NO TR
```

The job is halted in the last 2 cases.

Directive PRINTOUT (see 5.7.3.9) may produce the content of one register in various formats. If the message NO MON.PER is output in response to this directive, it will have to be re-input.

Directive DIRENT (see 5.7.3.10) may produce directory information (DIRENT 1 will). For the form of the printing see 5.8.2 for 150/17 instruction.

Directive RELINQUISH (see 5.7.3.11) will produce a relinquish message, e.g.

```
      (p)   (b)   RLQD *SR1
      (p)   (b)   RLQD *MT5-MC
```

Directive ABANDON (see 5.7.3.13) will produce

```
      (p)   (b)   ABAND y     z     n
      (p)   (b)   t
```

the job is abolished. See 5.3.2 for 150/11.

Directive WRONG (see 5.7.4.1) may produce

```
      (p)   (b)   RLQD *SR1-TRB
            (b)   HALTED
      (p)   (b)   RLQD *CRI
```

Directive UNLOAD (see 5.7.4.3) for a magnetic tape deck produces

```
      (o)   MTX   UNLOAD Serial-number    ERn
```

ERn is only printed if the number of errors is non-zero. The device is then disengaged.

Directive SPACE (see 5.7.4.4) lists the amounts of core, drum and peripherals not allocated to object programs, e.g.

```
            (o)   CORE  512  1088  1280
            (o)   DRUM  7621  2450
            (o)   IDLE  MTA  MTB  TRB
            (o)   FLOAT MT
```

Directive NAMES (see 5.7.4.5) list the jobs in the current order of priority and the semi-built-in programs in the machine e.g.

```
            (o)   JOBS.  SMITH  CHARLIE  SMITH2
            (o)   SBIPS.  PREADD  PUNCH  PRINT
            (o)   EMPTY
```

Directive PERIPHERAL (see 5.7.4.7) gives information about the device whether it is allocated or idle e.g.

```
            (o)   TRA   BELONGS TO  BLOGGS
            (o)   LPB   IDLE
            (o)   TRB   EDU/DATA/7/ABC
```

For an idle tape deck with a document loaded the information is

```
            (o)   MTX Serial-number Date NSn PAm Ll
            (o)   document name
```

Directive VOIDDATE causes Block 0 to be rewritten and the operator is asked to unload the tape (see 5.8.4.2).

Directive REMOTE (see 5.7.4.10) causes e.g.

```
            (o)   REMOVED PUNCH
```

to be printed when the specified SBIP's have been removed.

5.8.4 Messages produced on the Flexowriter by Peripheral Incidents.

5.8.4.1 Drum Failures.

OMP repeats drum transfers which read fail. Transfers are repeated up to five times and if they still fail the job concerned is suspended.

If the repeat is successful the message
(b) REPEAT n DRUM a TRCK b WORD c is printed, where n is the number of repeats. If the 5th repeat is unsuccessful then the above message is followed by

```
      (b) SUSPD
```

Other failures cause the whole machine to stop - this also happens if there are two distinct failures in quick succession or if a repeated failure occurs in a transfer initiated by OMP. The message is
(o) DRUM FAILURE DRUM a TRCK b word c where a, b and c are decimal integers giving the address of the failed words. On the next line some of the letters COAWR (for failure in Core, Outward transfer, Address failure, Write failure and Read failure respectively) are printed, as appropriate, followed by the current and finishing addresses (in decimal) of the transfer. The machine then comes to a loop stop and an engineer should be called. The failure message is followed by two lines which are punched on to the log tape to help the log program

```
      (p)   (o)  ╫
      (p)   (o)  END
```

These two lines also are output if there is a core store parity failure and by post mortem.

5.8.4.2 Magnetic Tape Incidents.

When the engage button is pressed OMP tries to read Block 0. If there is no Block 0 or it is non-standard then a message is output

```
      (o)  MTX   NEW TAPE
      (o)  MTX   NONSTAND TAPE
```

Directive SERIALNO or UNLOAD should be typed.

When OMP comes to rewrite Block 0, if the deck is isolated then the message

```
      (o) or (b)  MTX PERMIT WRITING
```

is printed. When Block 0 has been-rewritten OMP outputs

```
      (o) or (b)  MTX   Serial-number  (Date)  (NS)   (PA)  (L)
      (o) or (b)  (document name)
```

For those items in brackets only those which have been changed since Block 0 was last written are output (see 5.3.41)

When a tape is to be unloaded OMP outputs

```
      (o) MTX  UNLOAD serial-number ERn
```

and disengages the deck. ERn is printed only if n the number of errors since the tape was last loaded is non-zero.

Peripheral incidents on magnetic tape may produce printing.

```
      (b)  MTX READ FAIL         Code 1
      (b)  MTX WRITE FAIL        Code 2
      (b)  MTX WRITE INHIBIT     Code 3
      (b)  MTX END OF TAPE       Code 5
      (b)  MTX LAST FAIL         Code 6
      (b)  MTX FAIL              Code 8
      (o) or (b) MTX DECK FAIL   Code 7
```

If no restart is specified the job is suspended and SUSPD is printed between the geographical name and the incident message.

There is a facility for use by engineers which prints out details of each entry into OMP’s magnetic tape section. This is that after the message vertical bar is followed several octal integers

(o) or (b) MTX Message | a b c d e f

a is the difference between the finishing and starting address, unless this is not sensible in which case it is the finishing address.

b is the starting address

c is D24 to D31 of the interrogation information at the failure.

d is D0 to D7 of the interrogation information at the failure.

e is normally the block address of the failure block, except that at the beginning or end of tape where, for example, it has values like 77772.

f is D0 to D7 of the interrogation information at a subsequent failure, if any, during the automatic repeats.

5.8.4.3 Slow Peripherals

Peripheral incidents on slow peripherals may produce printing. These are of the form

(o) or (b) geographical name Incident(s)

If the incident (or one of them) has no restart then the default action is to halt the job and HALTED is printed between the geographical name and the incident(s). If there is an incident but no 1 bits have appeared in the interrogation information, then the incident message is omitted. The setting of the ENGINEER directive (see 5.7.4.9) may be such that extra information is printed.

The possible incidents are listed below. Those messages terminated by * mean that the operator’s or engineer's attention is required.

PARITY FAIL	Paper Tape
WRONG MODE	Paper Tape
ILLEGAL PUNCHING	Card reader
READ CHECK FAIL*	Card reader
BUFFER FAIL*	Card punch or printer
BUFFER OVR	Card Punch or printer
READ BACK FAIL*	Card punch
CARD WRECK*	Card punch
INTERRUPT	HPD, EDU or IBM deck
ADDRESS FAIL*	Any device
OPERATOR*	Any device
DISABLED*	Any device

If a peripheral incident occurs on the reader from which Primary Input (i.e. OMP) is reading, then a message is output on the Flexowriter, e.g.
(o) or (b) TRB OMP PARITY FAIL
Primary Input is switched off, the line forgotten and the reader disengaged.

If the Flexowriter fails while OMP is reading from it, then ╫ NL is output and the Flexowriter is left in the select state, so that the line can be repunched.

If a peripheral incident occurs on the monitoring peripheral when OMP is outputting to it, then OMP outputs a failure message on the Flexowriter and then repeats and/or continues outputting the information.

```
(b)      LPA      OMP      BUFFER   FAIL
```

If the failure is of operator type then OMP also disengages the device and when engaged continues.

If the Flexowriter fails while OMP is outputting to it then ╫ NL is output and the message repeated.

When the tape on the Flexowriter runs low the message

```
Tab      RELOAD   FLX
```

is output on the Flexowriter which is then disengaged.

5.8.5. Miscellaneous Messages

The time and date are output in the form.

```
(p)      (o)      13.40.00 7MAY65
```

The time only may be output every minute and both the time and date every ten minutes (the frequency of this message is controlled by the ENGINEER bits (see 5.7.4.9)).

The message

```
(b)      IDLE 4
```

reminds the operator of the existence of a job which has been waiting for 4 minutes, perhaps for a document to be loaded. The operator should normally attempt to find out why, and either get it going again or abolish it.

The message

```
(o)      RESERVATION PARITY FAILURE
```

is printed if the parity check on the reservation lines fails. No harm is done since the reservation lines are refilled every time the Time sharer enters a program. The message is printed so that the engineer can take appropriate action if it happens frequently.

A core store parity failure cause the message

```
(o)      CORE PARITY FAILURE  DEC  a  OCT  b
```

where a is the failed address in decimal and b is the failed address in octal. If OMP cannot find the failed register then NOT FOUND is printed instead of the failed address information. The machine then comes to a loop stop and an engineer should be called.

When OMP is first loaded the line

SWITCH TO NORMAL MODE

is output (though if the key is switched before this message is completed or started its output is abandoned). When this has been done, all the peripherals are disengaged and the line

DATE

is output and the Flexowriter left in the select state. The operator then types the date, e.g.

TUE 9 JULY 63

The day and month may be typed in full though at least 3 letters must be typed. The year may be punched as 63 or 1963. If the format is wrong or the day of the week does not correspond with the date, the line

TRY AGAIN

is output and the Flexowriter left in the select state.

The date may be followed by the ENGINEER digits required. If the log tape is suppressed then so is runout on the Flexowriter.

When the correct date has been typed there is a printout of Upper Case characters on the Flexowriter followed by

```
(p)  DOCUMENT ORION/LOG/9JUL63/10.02.17/2.3.7
```

The third component is the date, the fourth the time and the fifth the mark number.

The machine is then ready to receive programs.