A Copy 2

I.C.T. ATLAS COMPUTER

SUPERVISOR AND FIXED STORE ROUTINES

GENERAL DESCRIPTION

Both the Atlas computer and the routines described in this manual are the result of collaboration between Manchester University and Computer Equipment Division, International Computers and Tabulators Limited (formerly Computer Department, Ferranti Limited).

A Copy 2.

FERRANTI ATLAS COMPUTER

FIXED STORE ROUTINES

VOLUME ONE

GENERAL DESCRIPTION

Both the Atlas computer and the routines described in this manual are the result of collaboration between Manchester University and Ferranti Ltd.

This document describes the programs associated with the fixed store in Atlas. It is intended as a reference manual for those people writing these programs and for those requiring detailed knowledge of the running of an Atlas system. It is not expected that the average user of the computer will need to have recourse to this manual. This description is initially incomplete and it will augmented and amended from time to time.

Initially this description relates mainly to the programs written for the Atlas at Manchester University but the bulk of the information will be relevant to all Atlas computers. However, individual Atlas computers differ from one another in respect of the sizes of storage and numbers and types of input and output equipments, and resulting modifications to these programs will be supplied. Care must, therefore, be exercised in writing the relevant programs so that they can be altered for the different machines with a minimum of change.

The greater part of the programs listed in this document are stored in the fixed store of Atlas but in some cases due to shortage of space in the fixed store, it is necessary to store them permanently on magnetic drums from whence they can be brought to the main core store. Similarly, the working space required by these programs generally is the subsidiary store but in certain cases it is necessary to "extend" this store by locking down (and out) one or more pages of the core store . This manual will indicate whether a program is obeyed from the fixed store or the core store and also how much working space is used.

A companion volume to this manual is being issued giving the annotated program sheets for all the programs described here.

CONTENTS/1

Section 1. THE FIXED STORE

1.1 Description of the Fixed Store

The Fixed Store is constructed from ferrite and copper rods set in a woven wire mesh. These rods are 0.04" in diameter and 0.25" in length. A one is indicated by the presence of a ferrite rod and a zero by the presence of a copper rod. In addition to the information rods further ferrite rods are inserted as "keepers" and these also provide return paths for the flux in the information rods. As its name implies this store contains a standard or fixed pattern of ones and zeros which cannot be altered by machine program. The purpose of the store is to provide rapid access routines which

a) extend and complement the basic order code of the computer (Extracodes)

b) control the operation of the peripheral equipments

c) coordinate the operating system of the computer.

d) test the basic operations of the computer.

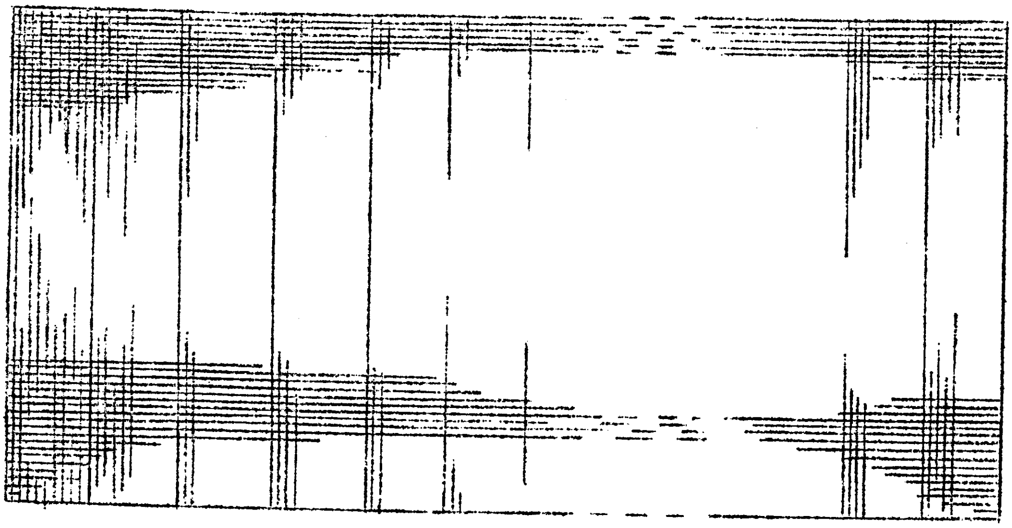
The store is built-up in multiples of 4096 forty-eight bit words and it is anticipated that 8,192 words will be sufficient for most installations. In addition to the information bits each twenty-four bit half-word has a parity digit associated with it and this digit is checked every time the word is read from the store. The access time for reading from the store is 0.2 μ sec. The time from sending a request for information from the Central Computer to receiving this information in the Central Computer is 0.6 μ sec.

The three most significant binary digits of an address in this store are 100 and the remaining address digits permit the specification of a theoretical maximum of 2^{18} forth-eight bit words. However for a machine containing 8,192 words address digits 20-16 inclusive are not decoded and hence addresses within the fixed store are treated modulo 8192.

Internally the Fixed Store is divided into "columns" of 256 words. Approximately one microsecond is required to switch from one column to another because various transient currents set up must be allowed to die down and whenever possible routines are written to avoid column changing.

The physical layout of a store of 4096 words each of 50 bits is given in the diagram overleaf.

1.1 Continued



50 sets of 16 read wires

The packing density of the storage rods is 80 digits per square inch and the size of the store is 3 feet by 8 feet i.e. 24 square feet. The mesh is folded over to give a double layer of 3 feet by 4 feet.

To simplify the read selection and to keep the leads short all the first digits of each word are grouped together, as are the second, third and subsequent digits. A read wire (vertical line in the diagram above) is connected to the same digit in each of 256 consecutively addressed words. The next read wire (within a set of 16) is connected to the same digit in each of the next 256 words. Thus first read wire on the left in the diagram is connected to digit 0 of words 0, 1, 2, ...255;
the second wire is connected to digit 0 of words 256, 257,.....511;
the sixteenth wire is connected to digit 0 of words 3840, 3841,.... 4095;
the seventeenth wire is connected to digit 1 of words 0, 1, 2....255
and so on.

Switching between the columns is done by the read selection circuits (taking approx. 1 μ sec.) and selection of a word within the column is done by switching on the appropriate drive circuits.

The basic unit used in assembling the fixed store is a plastic container from which 16 information rods and 16 keeper rods project. This is referred to as a "hairbrush", because of its appearance. Its size is 1" by $\frac{1}{4}$". The 16 information digits are in an array of eight by two and interspaced with eight by two keepers. When the hairbrush is loaded vertically the information rods give the digit values for a certain digit position of eight consecutive words together with the digit values for the same digit position of the eight consecutive words which are 256 words ahead. For example, the top left hand corner hairbrush in the earlier diagram gives the values of digit 0 for words 0 to 7 together with the values of digit 0 for words 256 to 263. Thus a vertical column of 32 hairbrushes gives the values of one digit position for a block of 512 words, so 50 such columns are needed to give all the digits for 512 words. There are eight columns to give digit 0 for all 4096 words, then eight for digit 1 and so on. These columns are distinguished by the colour of the hairbrushes, using a colour code; those for words 0 - 511

1.1 Continued

are brown, those for 512 - 1023 are red etc. Each hairbrush is further identified by a label on it which gives the digit number (00 - 49) and its frame position (0 - 31 measured downwards). For example, the top left hand hairbrush referred to before would be brown and its label would be 00/0, the one below it would be brown, with label 00/1, the one on its right would be red, with label 00/0 etc. Further, different coloured labels are used to distinguish between stacks of 4096 words. The mapping of the store is described in more detail in section 1.3.

It should be noted that the minimum amount of information which can be loaded is eight words with the eight words 256 away in the same block, and that the first word for this must have an address which is a multiple of 8.

1.2 Routines in the Fixed Store

1.2.1 Layout of the Fixed Store

Octal Address	Contents	No. of words Available	No. of words Allocated	Sibsidiary Store Words Required
40000000	Jump table; Magnetic Tape and Peripheral Extracodes 1000-1077	64	48	
40001000		192	192	
40004000	Jump table; Organisation Extracodes 1100-1177	64	46	
40005000		192	192	
40010000	Jump table, extracodes 1200-1277	64	60	
40011000	Extracodes 1200-1277	192	169	
40014000	Jump table, extracodes 1300-1377	64	54	
40015000	Extracodes, 1300-1377	192	192	
40020000	Jump table, extracodes 1400-1477	64	53	
40021000	Extracodes, 1400-1477	192	178	
40024000	Jump table, extracodes 1500-1577	64	58	
40025000	Extracodes 1500-1577	192	48	
40030000	Jump table, extracodes 1600-1677	64	58	
40031000	Extracodes, 1600-1677	192	28	
40034000	Jump table, extracodes 1700-1777	64	60	
40035000	Extracoes, 1700-1777	192	185	
				5

(As not all the extracode numbers are assigned some of the jump table entries have been replaced by extracode routines).

40040000	Initial Interrupts and Peripheral Routines	512	512	140
40050000	Engineers Initial Tests and Octal Input	512	512	
40060000	Drum Transfer Test	256		
40064000-	Parity Failure Routines	256	360	
40070000	Peripheral Routines	512	512	
40100000	Supervisor and Peripheral Routines	2048	2048	314
40140000	Drum routines including Store extracode	1024	1020	370
40160000	Magnetic Tape routines including extracodes	1024	1147	98
		8192	7732	927

1.2 ROUTINES in THE FIXED STORE

1.2.1 LAYOUT of THE FIXED STORE, MUSE

		FIXED STORE		ROUTINES
EXTRACODE JUMP TABLES		460		
EXTRACODE JUMP TABLE SPARES		52		
EXTRACODES		886		
ENGINEERS' TESTS		781		
SUPERVISOR				
Central Routines	1388		44	
Drum Routines	1131		25	
Magnetic Tape Routines	1001		18	
Peripheral Routines	2222		49	
Monitor and Fault Routines	212		9	
TOTAL SUPERVISOR		5954		145
		8123		
SPARE REGISTERS		69		
TOTAL FIXED STORE		8192		

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

ENGINEERS' TESTS

Initial Tests	256	*4005	-	*40053770
	256	*40054	-	*40057770
	8	*4006	-	*40060070
	8	*40064	-	*40064070
Call Tests from tape/drum	134	*40060100	-	*40062050
	3	*40063530	-	*40063550
Fixed store tests	8	*40063700	-	*40063770
	8	*40067700	-	*40067770
	8	*40173700	-	*40173770
	8	*40177700	-	*40177770
Binary input	84	*40064100	-	*40065330
	781			
Spare	10	*40063560	-	*40063670
	8	*40067600	-	*40067670
	18			
TOTAL	715			

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

CENTRAL SUPERVISOR ROUTINES.

		FS		FIXED STORE LOCATIONS
R200	Fixed store tables	9	3584-3592	0*4007 - 8*4007
R201	Enter Supervisor	41	3593-3633	9*4007 - 49*4007
R202	Program Scan	27	3634-3660	50*4007 - 76*4007
R203	Store location and lock out	114	3661-3774	77*4007 - 190*4007
R204	Halt main program	39	3775-3813	191*4007 - 229*4007
R205	Unlock store block	68	3840-3907	0*40074- 67*40074
R206	Enter SER to queue	14	3908-3921	68*40074- 81*40074
R207	Select main program	12	3922-3933	82*40074- 93*40074
R208	Update SER queues	26	3934-3959	94*40074- 119*40074
R211	Resume SER	14	3960-3973	120*40074- 133*40074
R212	Initial location of Supervisor block	8	3814-3821	230*4007 - 237*4007
R213	Halt SER	37	3974-4010	134*40074- 170*40074
R214	Free program	43	4011-4053	171*40074- 213*40074
R215	Set and Reset full recover switch	9	4054-4062	214*40074- 222*40074
R216	Establish tape exit	6	1224-1229	200*4002 - 205*4002
R217	Tape exit to Supervisor control	9	221-229	221*4000 - 229*4000
R218	Step block directory reference	11	1230-1240	206*4002 - 216*4002
R220	Reserve & free operators output	9	230-238	230*4000 - 238*4000
R221	Find tape deck number	17	239-255	239*4000 - 255*4000
R222	Supervisory program change	25	4071-4095	231*40074- 255*40074
R223	Check full program change	43	7168-7210	0*4016 - 42*4016
		2	1524-1525	244*40024- 245*40024
R226	Switch registers	91	7424-7514	0*40164- 90*40164
R227	Resume new program	35	7211-7245	43*4016 - 77*4016
R228	Fixed store program branch	15	7515-7529	91*40164- 105*40164
R229	Clock interrupt routine	34	7530-7563	106*40164- 139*40164
R230	One second SER	67	7246-7312	78*4016 - 144*4016
R233	Enter processing mode (fast)	30	956-985	188*40014- 217*400014
R235	Call supervisor to object program	18	3822-3839	238*4007 - 255*4007
R236	Fast exit from processing	25	986-1010	218*40014- 242*40014
R241	Central failure interrupt routine	29	3422-3450	94*40064- 122*40064
R242	Non-equivalence tapes and drums	20	3451-3470	123*40064- 142*400064
R243	Stop peripherals and print fault	59	3206-3264	134*4006 - 192*4006
R244	Emergency tape dump	97	3471-3567	143*40064- 239*40064
R245	Call tests	42	3265-3306	193*4006 - 234*4006
R247	Co-ordinate organisation extracodes	21	747-767	235*4001-255*4001
R248	Read/write isolated word	15	497-511	241*40004- 255*400004
R249	Special halt program	15	1241-1255	217*4002 - 231*4002
R251	Integrate system tapes	13	1011-1023	243*40014- 255*40014
R254	Organisation extracodes	31	7393-7423	225*4016 - 255*4016
		32	7648-7679	224*40164- 255*40164
R255	Output organisation extracodes	14	5439-5452	63*40124- 76*40124
R296	Idling hoot	1	4096-4096	0*4010 - 0*4010
		8	1216-1223	182*4002 - 199*4002
	(MANCHESTER has	8	3312-3319	240*4006 - 247*4006)
R297	Supervisor loop stop	1	2559-2559	255*40044- 255*40044
R298	Ignore lockout	2	1790-1791	254*4003 - 255*4003
R299	Anelex P.M.	90	8043-8132	107*40174- 196*40174

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

DRUM ROUTINES

		FS		FS ADDRESSES
R301	Instruction Counter Interrupt	17	64-80	64*4000 - 80*4000
R302	Page Selection Routine	9	81-89	81*4000 - 89*4000
R303	Learning Program	110	2449-2558	145*40044- 254*40044
R304	Select page	108	90-197	90*4000 - 197*4000
R311	Non - equivalence interrupt	18	320-337	64*40004- 81*40004
R312	Change page address register	8	4063-4070	223*40074- 230*40074
R313	Write to next available sector	56	2344-2399	40*40044- 95*40044
R314	Non-equivalence drum transfer routine	204	2088-2291	40*4004 - 243*4004
		3	1521-1523	241*40024- 243*40024
R315	Drum queue routine	63	338-400	82*40004- 144*40004
R317	Lose block B	30	401-430	145*40004- 174*40004
R318	Call to Cores	49	2400-2448	96*40044- 144*40044
R319	Set page address register	9	431-439	175*40004- 183*40004
R321	Read/Write up sector S	152	5149-5300	29*4012 - 180*4012
R322	Drum transfer complete interrupt	10	2292-2301	244*4004 - 253*4004
R323	Duplicate block B	70	6144-6213	0*4014 - 69*4014
R324	Rename block B	36	6214-6249	70*4014 - 105*4014
R327	Preserve and restore the accumulator	6	198-203	198*4000 - 203*4000
		6	7343-7348	175*4016 - 180*4016
R328	Duplicate block B to the drum	8	4344-4351	248*4010 - 255*4010
R329	Remove lock down	13	208-220	208*4000 - 220*4000
R331	Read to page P	55	6250-6304	106*4014 - 160*4014
R332	Clear store blocks	5	6305-6309	161*4014 - 165*4014
R333	Write/Release blocks	44	5301-5344	181*4012 - 224*4012
R340	Drum failure routine	33	440-472	184*40004- 216*40004
R398	Non-equivalence on I	3	2337-2339	33*40044- 35*40044
R399	Blister	6	8032-8037	96*40174- 101*40174

113/5/64.

LAYOUT of THE FIXED STORE, MUSE

MAGNETIC TAPE ROUTINES

		FS		FS ADDRESSES
R401	Block address interrupt routine	134	4709-4842	101*4011 - 234*4011
R403	Tape stopped interrupt routine	30	5376-5405	0*40124- 29*40124
R404	Alignment routine	17	5118-5134	254*40114- 14*4012
R405	Calculation of E.B.A.	24	4843-4866	235*4011 - 2*40114
R406	Channel failure routine	35	5453-5487	77*40124- 111*40124
R407	Parity 3 and Parity 6 routines	30	5488-5517	112*40124- 141*40124
R411	Prepare next tape order	72	4867-4938	3*40114- 74*40114
R412	Clear last tape order	14	5135-5148	15*4012 - 28*4012
R413	Start instructions	118	4939-5056	75*40114- 192*40114
R414	Organise store	61	5057-5117	193*40114- 253*40114
R421	Basic instructions to tape queue	101	4608-4708	0*4011 - 100*4011
		6	7337-7342	169*4016 - 174*4016
R431	Word search	12	6575-6586	175*40144- 186*40144
R432	Start variable length operations	84	5518-5601	142*40124- 225*40124
		3	6596-6598	197*40144- 199*40144
R433	Select deck	8	5602-5609	226*40124- 233*40124
R434	Transfer and skip instructions	175	6400-6574	0*40144- 174*40144
		2	6594-6595	194*40144- 195*40144
R435	Mark	7	6587-6593	187*40144- 193*40144
R436	Stop variable length operations	33	5406-5438	30*40124- 62*40124
		13	7635-7647	211*40164- 223*40164
R490	Organisational instructions	22	5610-5631	234*40124- 255*40124

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

PERIPHERAL ROUTINES

		FS		FS ADDRESSES
R500	Sort interrupts	40	2048-2087	0*4004 - 39*4004
R501	Load private store of any peripheral	77	5949-6025	61*40134- 137*40134
R502	Start reading form any input peripheral	36	6026-6061	138*40134- 173*40134
R503	Start writing to any output peripheral	9	6656-6664	0*4015 - 8*4015
R504	Free any peripheral	0	5967-5967	(79*40134- 79*40134)
R508	Peripheral one second subroutine	34	4352-4385	0*40104- 33*40104
		3	8133-8135	197*40174- 199*40174
R509	Find peripheral type	7	6341-6347	197*4014 - 203*4014
R511	Find store length available	12	6348-6359	204*4014 - 215*4014
R512	Shift up character in half word	14	4386-4399	34*40104- 47*40104
R513	Restore character positions in half word	8	4400-4407	48*40104- 55*40104
R514	Return to master routine from P.E.R.	7	6360-6366	216*4014 - 222*4014
R515	Start any peripheral	13	6665-6677	9*4015 - 21*4015
R516	Set code conversion parameters	11	6367-6377	223*4014 - 233*4014
R517	Character code conversion	72	6678-6749	22*4015 - 93*4015
R518	Preserve code conversion parameters	6	6750-6755	94*4015 - 99*4015
R519	Insert separator	22	6756-6777	100*4015 - 121*4015
R520	Set reserved block label	15	6778-6792	122*4015 - 136*4015
R521	Pick up record label	36	6793-6828	137*4015 - 172*4015
R522	Find peripheral buffer in part page	6	6829-6834	173*4015 - 178*4015
R523	Remove reserved block label	9	6378-6386	234*4014 - 242*4014
R527	Carriage control code conversion	28	6600-6627	200*40144- 227*40144
R530	Card reader, one second subroutine	23	7936-7958	0*40174- 22*40174
R531	column interruption	40	7959-7998	23*40174- 62*40174
R532	end of card interruption	29	4097-4125	1*4010 - 29*4010
R540	TR7 fault testing	28	6628-6655	228*40144- 255*40144
R541	TR7 interrupt	35	6835-6869	179*4015 - 213*4015
R550	Anelex printer, one second subroutine	17	6062-6078	174*40134- 190*40134
R551	character interruption	25	4126-4150	30*4010 - 54*4010
R553	P.E.R.	200	4408-4607	56*40104- 255*40104
R560	Creed 3000 fault testing	22	6870-6891	214*4015 - 235*4015
R561	Creed interrupt	13	6387-6399	243*4014 - 255*4014
R565	TR5 fault testing routine	20	6892-6911	236*4015 - 255*4015
R566	TR5 Interrupts	35	6079-6113	191*40134- 225*40134
R568	TR5 P.E.R.	256	6912-7167	0*40154- 255*40154
R570	Teletype fault testing routine	17	6114-6130	226*40134- 242*40134
R571	Seven channel teletype interruption	13	6131-6143	243*40134- 255*40134
R573	Teletype punch P.E.R.	304	5632-5935	0*4013 - 47*40134
R575	Card punch, one second subroutine	43	7564-7606	140*40164- 182*40164
R576	punch row interruption	30	6310-6339	166*4014 - 195*4014
R577	check read interruption	24	7313-7336	145*4016 - 168*4016
R578	end of card interruption	31	5345-5375	225*4012 - 255*4012
R585	Teleprinter fault testing routine	0	6114-6114	(226*40134- 226*40134)
R586	Teleprinter interruption	13	5936-5948	48*40134- 60*40134
R590	Peripheral Extracode linkage	25	1765-1789	229*4003 - 253*4003
R595	Input extracodes	193	4151-4343	55*4010 - 247*4010
R596	Output extracodes	184	7680-7863	0*4017 - 183*4017
	Graphical output	30		
	Talking	30		
	Goneometer	50		
R599	Peripheral working space	0		
		2222		

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

MONITOR AND FAULT ROUTINES

		FS		FS ADDRESSES
R630	Acquire one block	33	7999-8031	63*40174- 95*40174
R650	Active Scheduler	6	1759-1764	223*4003 - 228*4003
R700	Program monitor interrupts	29	668-696	156*4001 - 184*4001
R701	On line monitor SER	23	697-719	185*4001 - 207*4001
R702	One line monitor extracodes and traps	20	477-496	221*40004- 240*40004
R703	Block monitor	29	927-955	159*40014- 187*40014
R704	Instruction counter monitor	39	2002-2040	210*40034- 248*40034
R708	Require blocks for compiler	6	2041-2046	249*40034- 254*40034
R709	Off line Program error SER	27	720-746	208*4001 - 234*4001

212

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

FIXED STORE USED BY SUPERVISOR AND TESTS

	used	total used		not used	total not
4000	64-203,208-255	188		0-63,204-207	68
40004	64-216,220-255	188		0-63,217-220	68
4001	156-255	100		0-155	156
40014	159-255	97	0-158	159	
4002	192-231	40	0-191,232-255	216	
(MANCHESTER has 200-231			32	0-199,232-255	2247)
40024	241-245	5	0-240,246-255	251	
4003	223-255	33	0-222	223	
40034	210-254	45	0-209,255	211	
4004	0-253	254		254-255	2
40044	33-35,40-255	219		0-32,36-39	37
4005	0-255	256		none	0
40054	0-255	256		none	0
4006	0-237,248-255	246		238-239,240-247	10
(MANCHESTER has 0-234,240-255			254	1-333,235-239	2)
40064	0-7,94-239,248-255		162	8-93,240-247	94
4007	0-255	256		none	0
40074	0-255	256		none	0
4010	0-255	256		none	0
40104	0-255	256		none	0
4011	0-255	256		none	0
40114	0-255	256		none	0
4012	0-255	256		none	0
40124	0-255	256		none	0
4013	0-255	256		none	0
40134	0-255	256		none	0
4014	0-195,197-255	255		196	1
40144	0-195,197-255	255		196	1
4015	0-255	256		none	0
40154	0-255	256		none	0
4016	0-180,225-255	212		181-224	44
40164	0-182,211-255	228		183-210	28
4017	0-183,248-255	192		184-247	64
40174	0-101,107-196,248-255		200	102-106,197-247	56
(MANCHESTER has 0-101,107-199,248-255			203	102-106,200-247	53)
		6503		1689	
				6503	
				8192	

SUPERVISOR ROUTINES NOT LOADED

Graphical output	30
I.C.T.Data links	30
A.T.&E DATA Links	50
Talking	30
Goneometer	50
TOTAL	190
SUPERVISOR AND TESTS ROUTINES LOADED	6503
TOTAL SUPERVISOR AND TESTS	6693

13/5/64.

LAYOUT of THE FIXED STORE, MUSE

col:	4014	196	1		
col:	40144	196	1		

---

col:	4016	175 - 224	50	Not made	24|184 - 207
col:	40164	183 - 210	38		24|184 - 207
col:	4017	184 - 255	72		56|200 - 255
	40174	200 - 255	56		56|200 - 255
			227		

13/5/64.

1.2.2 List of Supervisor Routines

		Fixed Store	Subsidiary Store
R 201	Enter Supervisor	39	90
R 202	Program Scan	27	3
R 203	Store location and lock out	115	141½
R 204	Halt main program	42	44
R 205	Unlock store block	63	6
R 206	Enter SER to queue	17	
R 207	Select main program	10	½
R 208	Update SER queues	26	
R 209	Record supervisor pages	10	
R 210	Clear supervisor pages	23	
R 211	Resume SER	21	½
R 213	Halt SER	45	1½
R 214	Free program	54	
R 215	Set and reset Full recover switch	9	
R 216	Establish tape exit	6	
R 217	Tape exit to supervisor control	13	
R 218	Step block directory reference	9	
R 219	Aquire free block	8	1
R 221	Find tape deck number	19	4
R 222	Supervisory program change	27	1
R 223	Check full program change	20	1
R 224	Locate dump block	21	½
R 225	Update old program	18	6½
R 226	Switch registers	86	12
R 226	Resume new program	44	
R 228	Fixed Store program branch	50	
R 229	Clock interrupt routine	36	1
R 230	One second SER	57	
R 231	Timed scheduler	(45)	
R 232	Timed check on program	12	
R 233	Enter processing mode (fast)	20	½
R 234	Slow entry to processing	20	
R 235	Call sector to main program	21	½
R 236	Fast exit from processing	13	
R 237	Slow exit from processing	15	
R 238	General organisational extracodes	32	
R 239	Enter operators output	8	1
		1101	317

1.2.3 List of Drum Routines

		Fixed Store	Subsidiary Store (full words)
R301	Instruction Counter Interrupt Routine	10	17½
R302	Page Selection Routine	9	
R303	Learning program	201	146
R311	Non-equivalence interrupt	23	
R312	Change page address register	7	16
R313	Write to next available sector	59	12
R314	Non-equivalence drum transfer routine	167	†
R315	Drum queue routine	54	161½
R316	Read/write up block b	21	
R317	Lose block b	31	
R318	Call to cores	40	
R319	Set page address register	20	
R320	Reserved sectors location	50	16
R321	Read/write up sector s	150	
R322	Drum transfer complete interrupt	6	
R323	Duplicate block b	72	
R324	Rename block b	39	
R325	Drum absent interrupt	10	
R326	Drum failure interrupt	10	
R327	Preserve and restore the accumulator	10	
R328	Duplicate block b to the drum	8	
R329	Remove lock down	13	
R330	Non-equivalence on Interrupt Control	10	
		1020	370

1.2.4 List of Magnetic Tape Routines

Interrupt and Long Interrupt Routine		Fixed store	Subsidary Store (full words)
R400	Entry	10	8
R401	Leading block address interrupt	85	9
R402	Trailing block address interrupt	90	9
R403	Tape stopped interrupt	35	
R404	Alignment routine	30	
R405	Calculation of expected block address	25	
R406	Prepare next tape order	40	8
R407	Clear last tape order	40	
R408	Organize store	85	
R409	Tape stopped long interrupt	95	
R410	Channel failure interrupt	45	
R411	Parity three interrupt on tape transfers	15	
Tape Extracodes			
R412	Tape queue	35	48
R413	Search	15	
R414	Read forwards	15	
R415	Read backwards	20	
R416	Write	15	
R417	Skip forward	10	
R418	Skip backwards	10	
R419	Organizational instructions	70	
Variable Length Transfers			
R420	Start instructions	100	16
R421	Transfer instructions	200	
R490	Addressing Routine		
		1085	98

1.2.5 List of Peripheral Routines

The amount of working space required by the Peripheral routines depends on the number of equipments fitted to a particular installation. In general all peripherals require $5\frac{1}{2}$ - $6\frac{1}{2}$ forty-eight bit words of Subsidiary store each. In addition an index of the positions of these words is required. The size of this index is related to the number of peripheral connections provided on the computer but is independent of the total number of peripherals attached at any time.

	Title	Fixed Store	Subsidiary Store (full words)
R500	Initial interrupt	74	0
	List of private store used by each peripheral		20
R502	Transfer the contents of an Input Buffer to the store	41	
R503	Calculate the portion of the Input Buffer to be used	35	
R504	Prepare to read from any input peripheral	30	
R505	Free an input peripheral	4	
R506	Transfer any part of the store to any other	22	
R508	One second teletype subroutine		
R509	One second T.R.5 subroutine		
R510	Allocate buffer in input or output pages	7	
R511	Fill Output Buffer	65	
R512	Prepare to write to any output peripheral	31	
R513	Exit to the Supervisor from a Peripheral S.E.R.	6	
R514	Exit to the Coordinator from a Peripheral S.E.R.	5	
R540	T.R.5 Interrupt	280	
R541	T.R.7 Interrupt		
R542	I.C.T. Card Reader Interrupt (Column Read)	250	
R543	I.C.T. Card Reader Interrupt (End of Card)		
R550	I.B.M. Magnetic Tape Interrupt (Buffer Attention)		
R551	I.B.M. Magnetic Tape Interrupt (End of Operation)		
R552	I.B.M. Magnetic Tape, Mechanical Failure Routine		
R560	Seven Channel Teletype punch interrupt	150	$5\frac{1}{2}$ per punch
R561	Creed 3000 interrupt		
R562	Five Channel Teletype interrupt	150	$5\frac{1}{2}$ per punch
R563	Teleprinter interrupt	2 (additional to R560)	$5\frac{1}{2}$ per teleprinter
R564	I.C.T. Card Punch interrupt (Check Read)	300	
R565	I.C.T. Card Punch interrupt (Punch row)		
R566	I.C.T. Card Punch interrupt (End of Card)		

		Fixed Store	Subsidiary Store (full words)
R567	I.C.T. Hammer Printer interrupt (Character Reader)	222	156
R568	I.C.T. Hammer Printer interrupt (Line Count)		
R570	Graphical Output.		

1.2.4 List of Magnetic Tape Routines

		Fixed Store Registers	Subsidiary Store Registers
	Interrupt Routines		
R 400	Entry	11	
R 401	Block address interrupt routine	142	8
R 402	Tape stopped interrupt routine	44	8
R 403	Alignment routine	22	
R 404	Calculation of E.B.A.	27	
R 405	Channel failure routine	52	
R 406	Parity 3 and parity 6 routines	16	
	Long Interrupt Routines		
R 407	Prepare next tape order	59	3
R 408	Clear last tape order	14	
R 409	Tape stopped	126	8
R 410	Organize store subroutine	74	
R 411	Clear last subroutine	25	
	Basic Tape Extracodes		
R 412	Basic instructions	75	16
R 413	Tape queue	44	48
R 414	Organizational instructions	70 (Est).	
	Variable Tape Extracodes		
R 415	Start instructions	100 (Est).	12
R 416	Transfer instructions	200 (Est).	
		1,101	103

1.2.5. List of Peripheral Routines

The amount of working space required by the peripheral routines depends on the number of equipments fitted to a particular installation. In general, each peripheral requires $5\frac{1}{2}$ - $6\frac{1}{2}$ subsidiary store registers. In addition, an index of the position of these words is required, the size of which is related to the number of peripheral connections provided on the computer but is independent of the total number of peripherals attached at any time.

	TITLE	REGISTERS.
R500	Sort interrupts.	39
R501	Load private store of any peripheral.	77
R502	Start reading from any input peripheral.	34
R503	Start writing to any output peripheral.	9
R504	Free any peripheral.	None
R508	Peripheral one second.	34
R509	Find peripheral type.	7
R511	Find store length available.	12
R512	Shift up character in half word.	14
R513	Restore character position.	8
R514	Return to Master Routine from P.E.R.	7
R515	Start any peripheral.	13
R516	Set code conversion parameters.	11
R517	Character code conversion.	72
R518	Preserve code conversion parameters.	6
R519	Insert separator.	22
R520	Set reserved block label.	15
R521	Pick up record separator.	36
R522	Find peripheral buffer in part page.	6
R523	Remove reserved block label.	9
R527	Carriage control conversion.	28
R530	Card reader fault test.	23
R531	Card reader column interruption.	40
R532	Card reader end of card interruption.	29
R533	Card reader P.E.R.	181
R540	TR7 fault test.	28
R541	TR7 interruption.	35

	TITLE	REGISTERS
R550	Anelex fault testing.	17
R551	Anelex interruption.	25
R553	Anelex P.E.R.	200
R555	IBM tape.	
R560	Creed 3000 fault test.	22
R561	Creed 3000 interruption.	13
R565	TR5 fault test.	20
R566	TR5 interruption (tape readers 0 - 7)	35
R568	TR5 P.E.R.	256
R570	Teletype fault test.	17
R571	Teletype punch interruption (Teletypes 0-7)	13
R573	Teletype punch P.E.R.	304
R575	Card punch fault test.	43
R576	Card punch, punch row interruption.	30
R577	Card punch, check read interruption.	24
R578	Card punch, end of card interruption.	31
R579	Card punch P.E.R.	277
R585	Teleprinter fault test.	None.
R586	Teleprinter interruption.	13
R595	Input extracodes.	193
R596	Output extracodes.	189
R599	Working space for peripheral routines.	None.
	Total	2517

1.2.7 Monitor and Fault Routines

Monitor Routines

		Fixed Store	Subsidiary Store
R 700	Program monitor interrupts	18	1
R 701	On line monitor SER	15	
R 702	On line monitor extracodes and traps	41	5
R 703	Block Label monitor	29	1
R 704	Instruction counter monitor	(48)	1
R 705	Extracode exit from trap	14	
R 706	Instruction counter extracodes	23	
R 707	Page trap SER	10	
R 708	Off line trap entry to program	25	
R 709	Off line program error SER	23	
		246	8

Fault Routines

		Fixed Store	Subsidiary Store
R 750	Parity 1, 4, 5 interrupt routine	15	11
R 751	Parity 2 interrupt routine	30	$\frac{1}{2}$
R 752	Parity 2 S.E.R.	10	
R 753	Equivalence Tape/Drum interrupt routine	32	
R 754	General parity SER	15	
R 755	Stop peripherals	62	
R 756	Emergency error print	50	
R 757	Position dump tape	(29)	1
R 758	Interrupt control of tape	(24)	
R 759	Dump main store	(43)	
R 760	Dump working registers	(28)	
R 761	Call test routines	25	
		363	$12\frac{1}{2}$
	Total	609	$20\frac{1}{2}$

1.2.8 LAYOUT of THE FIXED STORE, LONDON and N.I.R.N.S.

R196 64 *7
R197 64
R430 64
R400 64
R301 64
R302 81
R304 90
R327 198
R329 208
R217 221
R220 230
R221 239
FIXED STORE COLUMN 40004

R311 320 64*40004
R315 338 82
R317 401 145
R319 431 175
R340 440 185
R702 477 221
R248 497 241
FIXED STORE COLUMN 4001

R700 668 156*4001
R701 697 185
R709 720 208
R247 747 235
FIXED STORE COLUMN 40014

R703 927 159*40014
R233 956 158
R236 986 218
R251 1011 243
FIXED STORE COLUMN 4002

R216 1224 100*4002
R218 1230
R249 1241
FIXED STORE COLUMN 4003

R659 1759
R590 1765
R298 1790

LAYOUT of THE FIXED STORE, LONDON and N.I.R.N.S.

FIXED STORE COLUMN 40034

R704 2002
R708 2041

FIXED STORE COLUMN 4004

R500 2048
R314 2088
R322 2292

FIXED STORE COLUMN 40044

R313 2344 40
R318 2400 —46
R303 2449 145

FIXED STORE COLUMNS 4006 and 40064

R243 3206
R245 3265

FIXED STORE COLUMN 40064

R241 3422
R242 3451
R244 3471

FIXED STORE COLUMN 4007

R200 3584 0
R201 3593 9
R202 3634 50
R203 3661 77
R204 3775 191
R212 3814 230
R235 3822 238

FIXED STORE COLUMN 40074

R205 3840 0*70074
R206 3908 68
R207 3922 82
R208 3934 94
R211 3960 120
R213 3974 134
R214 4011 171
R215 4054 214
R312 4063 223
R222 4071 177

FIXED STORE COLUMN 4010

R296 4096
R532 4097
R551 4126
R595 4151
R328 4344

113/5/64.

LAYOUT of THE FIXED STORE, LONDON and N.I.R.N.S.

FIXED STORE COLUMN 40104

R599 4352
R508 4352
R512 4386
R513 4400
R553 4408 —

FIXED STORE COLUMN 4011

R421 4608 0*
R401 4709 101* 4011
R405 4843 235*

FIXED STORE COLUMN 40114

R411 4867 3* 40114
R413 4939 75*
R414 5057 193*
R404 5118 254* 40114

FIXED STORE COLUMN 4012

R412 5135 15* 4012
R321 5149
R333 5301
R578 5345

FIXED STORE COLUMN 40124

R403 5376 0* 40124
R436 5406
R255 5439
R406 5453 77*
R407 5488 112*
R432 5518
R433 6600
R490 5610 234* 40124

FIXED STORE COLUMN 4013

R573 5632

FIXED STORE COLUMN 40134

R585 5936
R586 5936
R501 5949
R502 6026 –
R550 6062
R566 6079
R570 6114
R571 6131

13/5/64.

LAYOUT of THE FIXED STORE, LONDON and N.I.R.N.S.

FIXED STORE COLUMN 4014

R323 6144
R324 6214 70
R331 6250 106
R332 6305 161x
R576 6310
R509 6341
R511 6348
R514 6360
R516 6367
R523 6378
R561 6387

FIXED STORE COLUMN 40144

R434 6400
R431 6596 1758
R435 6587
R527 6600
R540 6628

FIXED STORE COLUMN 4015

R503 6656
R504 6665
R515 6665
R517 6678
R518 6750
R519 6756
R520 6778
R521 6793
R522 6829
R541 6835
R560 6870
R565 6892

FIXED STORE COLUMN 40154

R568 6912

FIXED STORE COLUMN 4016

R223 7168
R227 7211
R230 7246
R577 7313

FIXED STORE COLUMN 40164

R226 7424
R228 7515
R229 7530
R575 7564

FIXED STORE COLUMN 4017

R596 7680

FIXED STORE COLUMN 40174

R530 7936 0
R531 7959 23
R630 7999 63 [illegible]
R399 8032
R299 8038

EXTRACODE JUMPS AND ORGANISATION

R254 7393
R100 7680
R123 320

1.3 The Fixed Store Mapping Program

The fixed store consists of sets of copper and ferrite rods, as explained in section 1.1, in "hairbrush" units of 8 x 2 information rods in a plastic holder. These hairushes are assembled from loading sheets prepared by a Ferranti Pegasus program. The loading sheets give the digit values and information to fix the hairbrush position. The purpose of this section is to enable those concerned with writing programs to be built into the fixed store to prepare their data for the Pegasus program, and to give a short description of the program.

1.3.1 Preparation of Atlas program as data

The Atlas program is punched on 5 or 7 hole paper tape and written in Atlas Intermediate Input notation, with the following exceptions:

a) Floating point numbers are not recognised

b) The first half of a halfword pair must not begin with a decimal digit

c) Routine directives are not allowed

d) The program is limited to 512 words in length and must all be in one block of 512

d) With 7-hole tape only, comments preceded by [or ↑ may be included.

The program is then converted to a data tape as follows:

a) Before label 0 is first set a steering word A/0 must be punched, where half word A is the address of any word in the relevant block of 512.

b) The program is terminated by the directive E followed by CR LF and a title. The title may be of any length and is terminated by blank tape.

1.3.2 Detailed of the Mapping program and the loading sheets

There are two versions of the program; the Mapper, used for initial loading of information, and the Correction Mapper, used for making changes to already loaded information.

The program operates as follows

1. The data is read in, label values inserted etc. until the E directive is met.

2. A parity bit is calculated for each half word.

3. Words are read from the current record on magnetic tape to fill up incomplete hairbrush data, if this is necessary, and the magnetic tape record is updated.

4. If the program used is the correction mapper, a comparison is made of the new and current information during the updating, and a record kept of those hairbrushes which have been altered.

5. The information is punched out. If the correction mapper is being used, only those changed hairbrushes are output; if the mapper, all hairbrushes concerned are punched. There is a page of output for each digit. The digits are numbered from the least significant end 00 - 47, and digits 48 and 49 are the parity digits associated with digits 00 - 23 and 24 - 47 respectively. Each page is headed by the title, and by a 'label colour' and 'carrier colour'. The colour of the label identifies the Atlas installation and the stack of 4096 words as follows:

Installation	1st Stack	2nd Stack
Manchester University	White	Yellow
Harwell	Lilac	Light green
London Univerisity	Light blue	Pink

The colour of the plastic carrier identifies which 512 block the words are in i.e tells the columns in the store, as follows

Carrier colour	Address of words (Note: +4096 if stack 2)
Brown	0 - 511
Red	512 - 1023
Orange	1024 - 1535
Yellow	1536 - 2047
Green	2048 - 2559
Blue	2560 - 3583
Mauve	3072 - 3583
Grey	3584 - 4095

At the left of the page the digit number and the frame numbers are given, separated by an oblique stroke. The frame numbers, 0 - 31, give the hairbrush position within the column (0 for first 8 words, 1 for next 8 etc.) and the digit numbers identify the 50 digit columns. Alongside each digit number/frame number is given the loading information, and 8 x 2 array of noughts and ones.

1.3.3 An example of Atlas data and mapping output

Print out of the data tape

```
    +2560*4/0              -first halfword must not
                            begin with a decimal digit
    (0) = 2576*4

    (99) = 6*6

    152, 98, 119, 0
    225, 126,  0, 3(0)
    152,  99,119, 0.4
    224, 126,  0, (97)
    101,  91,  0, (99)
    163,  91,  0,   0
    572,  91,  0,   0
97) 521,   0,  0,   0
```

1.3 continued

```
(0)  =  2832*4          - 256 words ahead

+1  /0.7
+0.6/0.5
n4 / n3
n2 / n1
*4 / *52525252
(97)/-1(97)
147, 98, 119, 0
547, 99, 119, 0.4

E                       - end directive
DATA EXAMPLE            - title
```

Note that in this case no information is needed from the magnetic tape to fill hairbrushes; the data given is self sufficient.

The first three digits of output would be:

```
DATA  EXAMPLE                          - title

WHITE LABEL       BLUE CARRIER         - 1st stack
   00/2           00000000
                  00001111

DATA EXAMPLE

WHITE LABEL       BLUE CARRIER
   01/2           00000000
                  00010101

DATA EXAMPLE

WHITE LABEL       BLUE CARRIER
   02/2           00000100
                  10000011
```

The frame number is 2, for words 16-23 within the block.

The loading information as printed out has to be rotated 90 degrees anticlockwise to correspond with the data.

1.4 Programming Restrictions

Definitions

The basic instructions are divided into two classes.

a) unscrambled where the instruction is completed before the request for the next one is made.

These instructions consist of all B and Test codes where

(i) Ba = 120 - 127 except 122

or (ii) Ba = 122 and the contents of B121 = 120 - 127

or (iii) the unmodified operand is in the V-store.

b) scrambled where the request for the next instruction is made before the request for the operand for the first one.

These instructions are

(i) all B and Test codes not included above

(ii) all A codes.

The private store consists of those parts of the store whose three most significant address digits are

(a) 101 (not allocated)

(b) 110 the V-store

or (c) 111 the Subsidiary store

Restrictions on Programs on Main Control

(1) A Sacred Violation interrupt occurs if any reference to the private store is made. This includes references

(a) by not allocated s-type B-code (e.g. 151)

(b) by instructions where the address is not specified in the code (e.g. 34

(c) to the V-store by A-codes.

(2) As instructions are taken from the core store in pairs, the first instructio in a pair must not modify the second instruction in that pair. Normally the unmodified instruction is obeyed but the occurrence of an interrupt at the appropriate time causes the modified instruction to be obeyed after the interrupt has been dealt with.

(3) Similarly the odd instruction in a pair, unless it is an unscrambled instruction, must not modify either instruction in the next pair.

Restrictions on Programs on Interrupt or Extracode Control

(4) A Sacred Violation Look At Me digit (SVO) is set if the modified address is in the private store and the unmodified address is in the non-private store. In addition to the setting of this digit

a) on Interrupt control the instruction is omitted and the program continue

b) on Extracode control the instruction is omitted and one more instruction obeyed before an interrupt occurs.

(5) For instructions in either the Core or the Subsidiary store modification of the next instruction by means of a scrambled instruction on Extracode control must not be attempted. If modification of the next instruction by a scrambled instruction is carried out when an Interrupt control the unmodified instruction is obeyed. If modification of the next instruction by an unscrambled instructio on either Interrupt or Extracode control is carried out the modified instruction is obeyed.

(6) A control transfer to a routine in the core store from the subsidiary or fixed stores must be to an even addressed register. Otherwise if the "Pair" flip-flop is set to "Pair" the instruction held in the PIO register is obeyed first. Alternatively the Pair flip-flop can be set to "Not Pair" by writing to B127 before the control transfer and the transfer can then be to either an odd or even addressed register. This flip-flop is set to "Not Pair" when an instruction with an even address is encountered in the core store and hence control transfers within the routine are obeyed in the normal manner.

(7) The classification of an instruction must not change as a result of modification of the operand address. Thus modification into or out of the V-store is only permitted when Ba is such that the instruction is unscrambled.

(8) If the (unmodified) operand address for an A-code is in the V-store the machine stops on the next B-code.

(9) A request for an instruction in either the Not Allocated store or the V-store (three most significant digits 101 or 110) causes the machine to stop.

(10) All interrupt routines should end with the instruction 121, 125, 0, 2048*4

This transfers control to the beginning of the sequence which examines the look at me digits. If no more of these digits are set control is switched to either main or extracode by means of the instruction

113, 0, 0, 3 X 6

i.e. reset the I/ME digit to ME.

If an interrupt routine were terminated by the latter instruction and further look at me's are set, one instruction is obeyed on either main or extracode control before switching back automatically to interrupt control. If one of these further look at me's is for a non-equivalence or lock-out interrupt, where either one or two is added to the current control before switching to I, the supervisor may not re-enter the main program at the correct address.

SECTION 2. The Supervisor

2.1 The Co-ordination of Routines

The Structure of the Supervisor

The supervisor program controls all those functions of the system that are not obtained merely by allowing the central computer to proceed with obeying an object program, or by allowing peripheral equipments to carry out their built-in operations. The supervisor therefore becomes active on frequent occasions and for a variety of reasons - in fact, whenever any part of the system requires attention from it. It becomes activated in several different ways. Firstly, it can be entered as a direct result of obeying an object program. Thus, a problem being executed calls for the supervisor whenever it requests an action that is subject to control by the supervisor, such as a request for transfer to or from peripheral equipments or the initiation of transfers between core store and magnetic drums; the supervisor is also activated when an object program requires monitoring for any reason such as exponent or division overflow, or exceeding store or time allocation. Secondly, the supervisor may be activated by various items of hardware which have completed their assigned tasks and require further attention. Thus, for example, drums and magnetic tapes call the supervisor into action whenever the transfer of a 512 word block to or from core store is completed; other peripheral equipments require attention whenever the one character or row buffer has been filled or emptied by the equipment. Lastly, certain failures of the central computer, store, and peripheral equipments call the supervisor into action.

The central computer thus shares its time between these supervisor activities and the execution of object programs, and the design of Atlas and of the supervisor programs is such that there is mutual protection between object programs and all parts of the supervisor. The supervisor program consists of many branches which are normally dormant but which can be activated whenever required. The sequence in which the branches are activated is essentially random, being dictated by the course of any object program and the functioning of the peripheral equipments.

Interrupt Routines

The most frequent and rapidly activated parts of the supervisor are the interrupt routines. When a peripheral equipment requires attention, for example, an interrupt flip-flop is set which is available to the central computer as a digit in the V-store; a separate interrupt flip-flop is provided for each reason for interruption. If an interrupt flip-flop is set and interruptions are not inhibited, then before the next instruction is started, the address 2048 of the fixed store is written to the interrupt control register, B125, and control is switched to interrupt control. Further interruptions are inhibited until control reverts to main or extracode control, (apart from a "Non-equivalence on Interrupt Control" interrupt which only occurs due to either a Fixed Store program or machine fault). Under interrupt control, the fixed store program

---

This section and also sections 3.1, 4.1, 5.1 and 6.1 are based on a paper by Professor T. Kilburn, Dr. R.B. Payne and Dr. D.J. Howarth read at the E.J.C.C., December 1961.

which is held at address 2048 onwards detects which interrupt flip-flop has been set and enters an appropriate interrupt routine in the fixed store. If more than one flip-flop is set, that of highest priority is dealt with first, the priority being built-in corresponding to the urgency of action required. By the use of special hardware attached to one of the B registers, B123, the source of any interruption may be determined as a result of obeying between two and six instructions.

The interrupt routines so entered deal with the immediate cause of the particular interrupt. For example, when the one-character buffer associated with a paper tape reader has been filled, the appropriate interrupt flip-flop is set and the "Paper tape reader interrupt routine" is entered. This transfers the character to the required location in store after checking parity where appropriate. The paper tape reader meanwhile proceeds to read the next character to the buffer. Separate interrupt routines in the fixed store control each type of peripheral equipment, magnetic tapes and drums. The interrupt technique is also employed to deal with certain exceptional situations which occur when the central computer cannot itself deal adequately with a problem under execution, for example, when there is an overflow or when a required block is not currently available in the core store. There are therefore interrupt flip-flops and interrupt routines to deal with such cases. Further routines are provided to deal with interruptions due to detected computer faults.

During the course of an interrupt routine further interruptions are inhibited, and the interrupt flip-flops remain set in the V-store. On resumption of main or extracode control, interruptions are again permitted. If one or more interrupt flip-flops have been set in the meantime, the relevant interrupt routines are obeyed in the sequence
5 determined by there relative priority. In order to avoid interference with object programs or supervisory programs, interrupt routines use only restricted parts of the central computer, namely, the interrupt control register, B-lines 123 and 111 to 118 inclusive, private registers in the subsidiary store and the V-store and locked out pages in the core store. With the exception of the B-lines, no object program is permitted to use these registers. No lock out is imposed on the B-lines, but interrupt routines make no assumptions concerning the original contents of the B-lines and hence, at worst, erroneous use of interrupt B-lines by an object program can only result in erroneous functioning of that particular program. Switching of control to and from an interrupt routine is rapid, since no preservation or resetting of working registers is required.

The interrupt routines are designed to handle calls for action with the minimum delay and in the shortest time; the character-by-character transfers to and from peripheral equipments, for example, occur at high frequency and it is essential that the transfers be carried out with the minimum possible use of the central computer and within the time limit allowed by the peripheral equipment for filling or emptying the buffer. Since several interrupt flip-flops can become set simultaneously, but cannot be acted upon while another interrupt routine is still in progress, it is essential that a short time limit be observed by each interrupt routine. The majority of calls for interrupt routines involve only a few instructions, such as the transfer of a character, stepping of counts etc., and on conclusion the interrupt routine returns to the former control, either main or extracode. On some occasions, however, longer sequences are required; for example, on completion of the input of a paper tape or deck of cards routines must be entered to deal with the characters collected in the store, writing them to magnetic tape where appropriate, decoding and listing titles and so on. In such cases, the interrupt routine initiates a routine to be obeyed under extracode control, known as a supervisor extracode routine.

Supervisor Extracode Routines

Supervisor extracode routines (S.E.R.'s) form the principal "branches" of the supervisor program. They are activated either by interrupt routines or by extracode instructions occurring in an object program. They are protected from interference by object programs by using subsidiary store as working space, together with areas of core and drum store which are locked out in the usual way whilst an object program is being executed. They operate under extracode control, the extracode control register of any current object program being preserved and subsequently restored. Like the interrupt routines, they use private B-lines, in this case B-lines 100 to 110 inclusive; if any other working registers are required, supervisory routines themselves preserve and subsequently restore the contents of such registers. The S.E.R.'s thus apply mutual protection between themselves and an object program.

These branches of the supervisor program may be activated at random intervals. They can moreover be interrupted by interrupt routines, which may in turn initiate other S.E.R.'s. It is thus possible for several S.E.R.'s to be activated at the same time, in the same way as it is possible for several interrupt flip-flops to be set at the same time. Although several S.E.R.'s may be activated obviously not more than one can be obeyed at any one moment; the rest are either halted (see below) or held awaiting execution. This matter is organised by a part of the supervisor called the "co-ordinator routine" which is held in fixed store. Activation of an S.E.R. always occurs via the co-ordinator routine, which arranges that any S.E.R. in progress is not interrupted by other S.E.R.'s. As these are activated, they are recorded in subsidiary store in lists and an entry is extracted from one of these lists whenever an S.E.R. ends or halts itself. Once started, an S.E.R. is always allowed to continue, if it can; a high priority S.E.R. does not "interrupt" a low priority S.E.R. but is entered only on conclusion or halting of the current S.E.R. The co-ordinator has the role of the program equivalent of the "inhibit interrupt flip-flop", the lists of activated S.E.R.'s being the equivalent of the setting of several interrupt flip-flops. The two major differences are that no time limit is placed on an S.E.R., and that an S.E.R. may halt itself for various reasons; this is in contrast to interrupt routines, which observe a time limit and are never halted.

In order that the activity of each branch of the computing system be maintained at the highest possible level, the S.E.R.'s awaiting execution are recorded in four distinct lists. Within each list, the routines are obeyed in the order in which they were activated, but the lists are assigned priorities, so that the top priority list is emptied before entries are extracted from the next list. The top priority list holds routines initiated by completion of drum transfers, and also routines entered as a result of computer failure such as core store parity. The second list holds routines arising from magnetic tape interruptions and the third holds routines arising from peripheral interruptions. The lowest priority list contains one entry for each object program currently under execution, and entry to an S.E.R. through an extracode instruction in an object program is recorded in this list. On completion of an S.E.R., the co-ordinator routine selects for execution the first activated S.E.R. in the highest priority list.

The central computer is not necessarily fully occupied during the course of an S.E.R. The routine may, for example, require the transfer of a block of information from the drum to the core store. in which case it is halted until the drum transfer is completed. Furthermore, the queue of requests for drum transfers maintained in the subsidiary store may be full, in which case the S.E.R. making the request must be halted. When an S.E.R. is halted for this or similar reasons, it is returned to the relevant list as halted, and the next activated S.E.R. is entered by the co-ordinator routine. Before an S.E.R. is halted, a restart point is specified. A halted routine is made free to proceed when the cause of the halt has been removed - for example, by the S.E.R. which controls drum transfers and the extraction of entries from the drum queue. The S.E.R. lists can therefore hold at any one time routines awaiting execution and halted routines; interrupt routines are written in such a way that the number of such S.E.R.'s activated at any one time is limited to one per object program, and one or two per interrupt flip-flop, depending upon the particular features of each interrupt routine. When an S.E.R. is finally concluded, as distinct from halted, it is removed from the S.E.R. lists and becomes dormant again.

Although S.E.R.'s originate in many cases as routine to control peripheral equipment, magnetic tapes and drums, it should not be supposed that this is the sole function of these routines. Entrances to S.E.R.'s from interrupt routines or from extracode instructions in an object program initiate routines which control the entire operation of the computing system, including the transfer of information between store and peripherals, communication with the operators and engineers, the initiation, termination and, where necessary, monitoring of object programs, the monitoring of centrol computer and peripheral failures, the execution of test programs and the accumulation of logging information. Each branch of supervisory activity is composed of a series of S.E.R.'s, each one activated by an object program or an interrupt routine and terminated usually by initiating a peripheral or magnetic tape transfer or by changing the status of an S.E.R. list or object program list. The most frequently used routines are held in the fixed store; routines required less frequently are held on the magnetic drums and are transferred to the core store when required. Supervisor routines in the core and drum store are protected from interferences by object programs by use of hardware lock-out and the basic store organisation routines in the fixed store.

Object Programs

The fuction of all supervisor activity is, of course, to organise the progress of problems through the computer with the minimum possible delay. Object programs are initiated by S.E.R.'s, which insert them into the object program list; they are subsequently entered by the co-ordinator routine effectively as branches of lower priority than any S.E.R. Although object programs are logically sub-programs of the supervisor, they may function for long periods using the computer facilities to the full without reference to the supervisor. For this reason, the supervisor program may be regarded as normally dormant, activated and using the central computer for only a small proportion of the available time.

In order to allow object programs to function with the minimum of program supervision, they are not permitted to use extracode control or interrupt control directly, enabling protection of main programs and supervisor programs to be enforced by hardware. Object programs use the main control register, B127, and are therefore forbidden access to the V-store and subsidiary store. Reference to either of these stores causes the setting of an interrupt flip-flop and hence entrance to the supervisor program.

Access to private stores is only obtained indirectly by use of extracode functions, which switch the program to extracode control and enter one of a possible maximum of 512 routines in the fixed store. These extracode routines form simple extensions of the basic order code, and also provide specific entry to supervisor routines to control the transfer of information to and from the core store and to carry out necessary organisation. Such specific entrances to the supervisor program maintain complete protection of the object programs. Protection of magnetic tapes and peripheral input and output data is obtained by the use, in extracode functions, of logical tape and data numbers which the supervisor indentifies within each program with the titles of the tapes or information. Blocks of core and drum store are protected by hardware and by the supervisor routines in the fixed store.

An object program is halted (S.E.R.'s) whenever access is required to a block of information not immediately available in the core store. The block may be on the drums, in which case a drum transfer routine is entered, or it may be involved in a magnetic tape transfer. In both cases the program is halted until the block becomes available in core store. In the case of information involved in peripheral transfers, such as input data or output results, the supervisor buffers the information in core and drum store, and "direct" control of a peripheral equipment by an object program is not allowed. In this way, immobilisation of large sections of store whilst a program awaits a peripheral transfer can be avoided. A program may however, call directly for transfers involving drums or magnetic tapes by use of extracode functions, which cause entrance to the relevant supervisor routines. Queues of instructions are held in the subsidiary store by these routines in order to allow the object program to continue, and to achieve the fullest possible overlap between tape and drum transfers and the execution of an object program.

Whilst one program is halted, awaiting completion of a magnetic tape transfer for instance, the co-ordinator routine switches control to the next program in the object program list which is free to proceed. In order to maintain full protection, it is necessary to preserve and recover the contents of working registers common to all programs such as the B-lines, accumulator, and control registers, and to protect blocks in use in the core store. The S.E.R. to perform this switching from one object program to another occupies the central computer for around 750 + 12p μ secs. where p is the number of pages, or 512 word blocks in core store. On the Manchester University Atlas, which has 32 pages of core store, the computing time for the round trip to switch from one program to another and to return subsequently is around 2.5 m.secs. This is in contrast to the time of around 60 μ secs. to enter and return from an S.E.R. and even less to switch to and from an interrupt routine. It is therefore, obvious that the most efficient method of obtaining the maximum overlap between input and output, magnetic tape transfers, and computing is to reduce to a minimum the number of changes between object programs and to utilise to the full the rapid switching to and from interrupt and supervisor routines. The method of achieving this in practice is described in Section 6.

Compilation of programs is treated by the supervisor as a special case of the execution of an object program, the compiler comprising an object program which treats the source language program as input data. Special facilities are allowed to compilers in order that their allocation of storage space may be increased as need arises, and to allow exit to the supervisor before the execution of a problem or the recording of a compiled object program.

Error Conditions

In addition to programmed entrances to the supervisor, entrance may also be made in the event of certain detectable errors arising during the course of execution of a problem. A variety of program faults may occur and be detected by hardware, by programmed checks in extracodes, and in the supervisor. Hardware causes entry to the supervisor by the setting of interrupt flip-flops in the event of overflow of the accumulator, use of an unassigned instruction, and reference to the subsidiary store or V-store. Extracode routines detect errors in the range of the argument in square root, logarithm, and arcsin instructions. In the extracodes referring to peripheral equipment or magnetic tapes, a check is included that the logical number of the equipment has been previously defined. In extracodes for data translation, errors in the data may be detected. The supervisor detects errors in connection with the use of the store. All problems must supply information to the supervisor on the amount of store required, the amount of output, and the expected duration of execution. This information is supplied before the program is compiled, or may be deduced after compilation. The supervisor maintains a record of store blocks used, and can prevent the program exceeding the present limit. In addition, an interrupt flip-flop is set by a clock at intervals of 0.1 secs., and another flip-flop is set whenever 1024 instructions have been obeyed using main or extracode control. These cause entrances to the supervisor which enable a program to be "monitored" to ensure that the present time limit has not expired, and which are also instrumental in initiating routines to carry out regular timed operations such as logging of computer performance and initiation of routine test programs.

The action taken by the supervisor when a program "error" is detected depends upon the conditions previously set up by the program. Certain errors may be individually trapped, causing return of control to a preset address; a private monitor sequence may be entered if required enabling a program or a compiler to obtain diagnostic printing; failing specification of these actions, some information is printed by the supervisor and the program is suspended, and usually dumped to magnetic tape to allow storage space for another program.

2.2 Interrupt and Supervisory Routines

In order that short interrupt and supervisory extracode routines can be incorporated in the overall supervisory scheme, it is important that writers of such routines should observe certain rules and principles. These rules are given below.

1. Definition of Interrupt and Supervisor Routines

1.1 Interrupt Routines

Entered following the setting of a "look at me"
Use interrupt control (B125)
May use B111 - B118 inclusive and B123
May alter only private registers in subsidiary store
May not use Bt, Accumulator, or any other registers unless preserved and reset within the interrupt routine
May use main store only if this is previously locked down and out, and the relevant page address registers are locked out
Exit to current program by resetting I/ME flip-flop to zero (i.e. to ME) or to supervisor extracode routine via the co-ordinator or by direct exit. In the latter case the entry address and contents of B111 only are preserved.

1.2 Supervisor Extracode Routines (SER)

Enter from (a) Interrupt routines ("long interrupt")
(b) Extracode routines in main programs
(c) other SER
Normal entrances are controlled by the Co-ordinator.
Use extracode control (B126) and hence may be interrupted
May use B lines 100-110 inclusive and central computer V store line 6
May use and alter registers in subsidiary store
May not alter other registers (including accumulator and main control) unless these are preserved and restored within the SER.
Exit (a) to main program via the co-ordinator
(b) to another SER via the co-ordinator or
(c) to another SER by direct transfer of control.

Use of the main store is allowed under certain conditions and this is discussed in detail in Section 2.3.
No time limit is placed on SER. Short interrupts are permitted, but no further SER are entered until the conclusion of the current SER or until the current SER is halted or is interrupted by a special entrance to SER. (Note that in this respect SER differ from main programs, which may be interrupted at any time and control transferred to a main program of higher priority).

2. The SER Queues and the Program List

During the course of an SER other SER's may be "entered" via interrupt routines. The co-ordinator forms queues of such SER's and enters them in turn on completion of the current SER. These queues are also used to record halted SER's. In exactly the same way, the program list is used to record main programs awaiting execution, the main programs which are halted. The method of recording SER differs from the recording of main programs in that it is necessary to hold more information for each entry in the program list, but the number of entries is much less than in the SER queues.

Three SER queues are provided:-

1) Top priority queue including Drum SER, Non Equivalence and Lock out etc.
2) Tape queue including Magnetic Tape SER
3) Slow queue including Slow Peripheral SER

When the current SER is concluded or halted, the co-ordinator scans these queues in turn, and within each queue initiates the SER's in the order in which they were entered to the queue. A separate entry, the "current entry" is provided for the current SER.

When halted, SER's are recorded at the end of the relevant SER queue, or are recorded in the main program list if they arise from an instruction in the current main program. It is convenient to introduce the concept of a base for each SER, where it is recorded if halted. This base will be either a main program or an entry in one of the SER queues, generally corresponding to a particular interrupt. This base for halts will generally be the same as the "base for waiting" i.e. the position where the SER is recorded when awaiting execution, but the two may differ when the base for halting is a main program.

The SER queues in the subsidiary store must be of finite maximum length. It is essential that this be borne in mind when entries are made to the SER queues from interrupt routines. The total number of SER's in course of execution at any one time must never exceed the available number of bases and a predictable upper limit must be set on this number. The time between entry of an SER to a queue and its completion will be unpredictable since:

1) The time it may wait in the queue before being entered depends upon the existing state of the SER queues at the time of entry to the queue.
2) An SER may be halted, thus occupying a "base" for a long period after it is started.

In order to avoid an unpredictable overflow of bases, it is necessary to know explicitly whether a base for an SER is available before it is entered to a queue. In no circumstances must the prescribed quota of bases be exceeded; it should not be "assumed" that a previous SER has been completed simply through lapse of time.

3. Halting of SER

The need to halt an SER is detected by subroutines of the supervisor which are common to all relevant SER. When halted, an SER is recorded in the relevant base, either in the main program list or in an SER queue. SER are resumed subsequently when the cause of the halt is removed, by entry to another supervisor subroutine from SER which have removed a possible cause of halts (e.g. tape and drum SER). When several SER are halted for the same reason, they are resumed by scanning in order the top priority queue, tape queue and finally the main program list.

SER based on a main program or a queue may be halted for the following reasons:

1) Full Drum Queue
2) Full Tape Queue
3) Operators' output busy
4) Peripheral cannot be started
5) No free store block available

An SER based on a main program may be halted for:

6) Block of supervisor or program not available in core store

An SER based on an SER queue may be halted for:

7) Non equivalence interrupt involving a supervisor store block, not a main program store block.
8) Request for a drum transfer to core store, via a subroutine "Call to Cores". (R318).
9) Request for a drum transfer to a reserved sector, via a subroutine "Write to Sector". (R321).

Halts (1) to (6) are imposed by entry to one of two subroutines "Halt Main Program" (R204) and "Halt Supervisor Routine" (R213). Halts 7 - 9 are imposed by the relevant subroutines "Call to Cores" and "Write to Sector".

After being halted for reasons (1) to (5), an SER is re-entered to the active list (either SER queue or main program list) via the subroutine "Free Program" (R214). It is started in the normal priority sequence. Similarly a main program SER is made active after halt (6) via the subroutine "Unlock Store Block (R205) which is entered on conclusion of a drum transfer which caused halts (7) to (9), the SER is resumed immediately on completion of the drum SER.

When an SER based on an SER queue requests a drum transfer, the drum routines effectively form an open subroutine interrupting the SER and resuming it again on completion of the drum transfer. When tape or slow peripheral transfers are requested, a new "branch" is effectively started and control returns after initiation of the transfer (or after entering it to a queue) to the original SER. On conclusion of the transfer, a new SER is started, with in general a new base; usually the original SER will be terminated after requesting the transfer.

An entry in the SER queue comprises two half-words, the "Supervisor Entry Parameters". One holds the "Entry Address" in digits 23-3; the other holds "Entry Information " in digits 23-0. When the SER is started, the "Entry Information" is placed in B100 and the "Entry Address" in the "Current Entry Address" position in subsidiary store before control is transferred to the Entry Address. The following actions can occur when an SER is halted:

1) For a main program SER, B100 - 104 and the current entry address are preserved, and subsequently restored when the SER is restarted. The SER is re-entered at the Current Entry address in extracode control.

2) A main program SER may also halt the program so that the program resumes "out of Supervisor", with no Supervisor information recorded (e.g. after a main program halt for non-equivalence).

3) In the case of an SER whose base is an SER queue, if the "current entry address" is even (digit 0 = 0), B100 and the current entry address only are preserved and restored on re-entry. The contents of B101 - 110 are lost. The SER is re-entered at the current entry address.

If the "current entry address" is odd (digit 0 = 1), then a register in subsidiary store, "SER dump" holds the address of a dump in the main store in which B100 - 104 are preserved. B100 - 104 are restored on re-entry. This dump address is private to the particular SER and must be known to be in core store at the time of the halt. Again the SER is re-entered at the current entry address.

When the SER is halted through non-equivalence interrupt, the "Current Entry Address" is taken to be the address of the instruction causing the interrupt, and resumption is in Main or Extracode control as appropriate. The significance of digit 0 of the "Current Entry Address" in subsidiary store remains as usual.

In all cases when an SER may be halted, careful thought must be given to establishing a suitable restart point, whose address must be written to subsidiary store before the halt. It must be emphasised that only limited B-line information is preserved on a halt, and the possibility of multiple halts in a sequence of operations must be recognised and considered in detail in each SER.

4. <u>Entry to SER</u>

"Normal" entries are via subroutines of the co-ordinator.

1) <u>Entrance from Extracode</u>

SER are entered from all extracodes requiring access to supervisor information or routines (e.g. peripheral and organisational extracodes). Entrance is via entry A (1/R201) of the co-ordinator. The SER is entered at address S, held in B96 on entry. On conclusion of the SER, the main program is resumed under main control, or under extracode control at address E, held in B97 on entry. The "Supervisor Entry Address" is set to S; the "Supervisor Entry Information" is not set. The SER may use information from subsidiary store and from B lines 0 - 99.

These SER always occupy the "current entry", and never enter an SER queue. If necessary, they are halted by halting the relevant main program. According to the type of halt, the main program is subsequently resumed by recovering B0-99 or by resuming "In Supervisor", resetting the "Current Entry Address" and B100 - 104 and entering the SER at the "Entry Address".

2) Entrance from Interrupt

Entry is under interrupt control via entries B, C or D, (2, 3 or 4/R201), of the co-ordinator, entry B for top priority queue, entry C for tape queue, and entry D for slow queue. Initial "Supervisor Entry Parameters" are held in B111 (information) and B112 (entry address) on entry to the co-ordinator, and on starting the SER these are recorded in B100 (information) and in the "current entry address". If necessary, entries are held in the relevant SER queue before being started.

3) Special entrance from Interrupt

Certain interrupt routines (for example, non-equivalence interrupt when "In Supervisor") will require to interrupt the current SER. This may be done by setting B126 appropriately and entering extracode control in the usual manner, without entry to the co-ordinator. It is the responsibility of such interrupt routines to preserve, if necessary, the original value of B126 and any supervisor B lines B100 - 110.

4) Creation of a new SER

A new SER may be entered to a queue and control returned to the current SER. This technique of "branching" has many applications, one example being when the One Second SER detects a call for attention from a peripheral equipment or magnetic tape. The co-ordinator is entered at F (4/R206) under extracode control with B108, B109 holding the supervisor entry parameters to be queued (B109 = Information, B108 = Entry address). The queue is specified on entry by B107 which holds 2.0 (top priority queue), 1.0 (tape queue), 0.0 (slow queue). After the item has been added to the appropriate queue, control returns to the address held in B110 on entry. B106, 107 are altered by the co-ordinator.

A special entry point to R206 enables the drum supervisor to cause an item to be entered in a queue on completion of a drum transfer.

5) Entrance via Program Scan Exit

The normal conclusion of an SER is by entry to the co-ordinator at G (R202) under extracode control. A half word in subsidiary store, "Program Scan Exit", can cause continuation to another SER. If Program Scan Exit is zero, no special action is taken. If it is non-zero and digits 2 - 0 are zero, then control is transferred to the address contained in Program Scan Exit with B110 altered but B100 - 109 preserved. The new SER so entered is regarded as a continuation of the previous SER, with the same base. If digits 2 - 0 of Program Scan Exit are not all zero, the co-ordinator initiates a new SER with a new base. Digits 2, 1 define the SER queue on which the SER is to be based (01 = Tape queue, 10 = Top priority queue, 11 = Slow queue). The new SER is entered with B100 - 108 preserved. In all cases, before any transfer is made, Program Scan Exit is reset to zero by the co-ordinator.

5. Length of SER Queues

Only one SER can be currently initiated at any one time via an extracode entry. This SER has as base the main program, and never enters an SER queue.

The interrupt routines which can initiate SER are listed below. The list shows the queue in which the SER is entered originally, the queue on which it is based for subsequent halts, and the suggested upper limit to the number of SER based on each interrupt for which provision is necessary. The SER queues are numbered 1, 2, 3, in order of priority (1 = top priority). P denotes the main program list, S denotes a special interrupt entry.

If more bases are required, these should be requested in the reasonably near future in order that allocation of space in subsidiary store may be made. At present it would appear necessary to reserve:

7 entries in Queue 1

24 entries in Queue 2

around 32 entries in Queue 3 on the Manchester University Atlas.

Slow Interrupt		Entry Position	Base for Halts	Max. No. of SER in Queue
Peripherals		3	3	1 per "look at me"
Unassigned function Division Overflow S.V. Instruction S.V. Operand Exponent Overflow		3	3	1
Lock out or Non-equivalence, Normal		1	P	1
Lock out or Non-equivalence, Supervisor		S	None	0
Non-equivalence Interrupt, Tapes or Drums		S	1	1
Drum Transfer complete		1	None	1
Drum Fail		1	1	1
Drum Cabinet Absent		1	1	1
Magnetic Tape Block Address		2	2	2 per channel + 1 per deck
Tape Fail		2	2	Replace block address
Instruction Counter	-Program	P	P	0
	-Timers	1	1	1
Clock	-Program	P	P	0
	Scan	1	None	1
	Tape Action	3	3	1
	Timed exit	3	3	1
Parity	1 Core	S		
	2 Drum	1	1	1
	3 Tape	None	-	-
	4 Sub Store	S		
	5 Fixed Store	S		
	6 Tape	None	-	-

SECTION 3. The Magnetic Drum Routines

3.1 Store Organisation

Indirect addressing and the One-level Store

The core store of Atlas is provided with a form of indirect addressing which enables the supervisor to re-allocate areas of store and to alter their physical addresses, and which is also used to implement automatic drum transfers. With each page, or 512 word block, of core store there is associated a "page address register" which contains the most significant address bits of the block of information contained in the page. Every time access is required to a word of information in the core store, the page containing the word is located by hardware. This tests for equivalence between the requested 'block address', or most significant address bits, and the contents of each of the page address registers in parallel. Failure to find equivalence results in a "non-equivalence" interruption. The page address registers are themselves addressable in the V-store and can thus be set appropriately by the supervisor whenever information is transferred to or from the core store.

One of the most important consequences of this arrangement is that it enables the supervisor to implement automatic drum transfers. The address in an instruction refers to the combined core and drum store of the computer, and the supervisor records in the subsidiary store the location of each block of information; only one copy of each is kept, and the location is either a page of core store or a sector of the drum store. At any moment, only some of the blocks comprising a particular program may be in the core store and if only these blocks are required, the program can run at full speed. When a block is called for which is not in the core store, a non-equivalence interruption occurs, which enters the supervisor to transfer the new block from a sector of the drum to a page of the core store. During this operation the program that was interrupted is halted by the supervisor.

The block directory in the subsidiary store contains one entry for each block in the combined core and drum store. It is divided into areas for each object program which is in the store; a separate program directory defines the area of the block directory occupied by each program. The size of this area, or the number of blocks used by a program, is specified before the program is obeyed in the job description. The entry for block n contains the block number n together with the number of the page or sector occupied by the block, and, if possible, is made in the n^{th} position in the area; otherwise the area is filled working backwards from the end. In this way, blocks used by different object program are always kept distinct, regardless of the addresses that are used in each program. A program addresses the combined "one-level" store and the supervisor transfers blocks of information between the core and drum store as required; the physical location of each block of information is not specified by the program, but is controlled by the supervisor.

There are occasions when an object program must be prevented from obtaining access to a page of the core store, such as one involved in a drum or tape transfer. To ensure complete protection of such pages, an additional bit, known as a lock out bit, is provided with each page address register. This prevents access to that page by the central computer, except when on interrupt control, and any reference to the page causes a non-equivalence interruption. By setting and resetting the lock out bits, the supervisor has complete control over the use of core store; it can allow independent object programs to share the core store, it can reserve pages for peripheral transfers and can itself use parts of the core store occasionally for routines or working space, without any risk of interference. This is done by arranging that, whenever control is returned to an object program, pages that are not available to it are locked out.

A block of information forming part of an object program may also be locked out from use by that program because an operation on that information, controlled by the supervisor, is not complete. A drum, magnetic tape, or peripheral equipment transfer involving this block may have been requested. The reason for the lock out of such a block is recorded in the block directory, and if the block is in the core store, the lock out digit is also set. If reference is made to such a block by the object program, a non-equivalence interruption occurs and a supervisor extracode routine halts the program. This S.E.R. is restarted by the co-ordinator routine when the block becomes "unlocked", and the object program is re-entered when the block is available in core store.

The Drum Transfer Routine

The drum transfer routine is a group of S.E.R.'s which are concerned with organising drum transfers, and updating page address registers and the block directory. Once initiated, the transfer of a complete block to or from the drum proceeds under hardware control; the drum transfer routine initiates the transfer and identifies the required drum sector by setting appropriate bits in the V-store. It also identifies the core store page involved by setting a particular "dummy" block address, recognised by the drum control hardware, in the page address register; at the same time, this page is locked out to prevent interference from object programs whilst the transfer is in progress.

On completion of a transfer, an interruption occurs which enters the drum transfer routine. The routine can also be entered from the non-equivalence interrupt routine, which detects the number of the block requested but not found in the page address registers. Finally, the drum transfer routine can be activated by other parts of the supervisor which require drum transfers, and by extracode instructions which provide a means whereby object programs can if they wish exert some control over the movement of blocks to and from the drum store. A queue of requests for drum transfers, which can hold up to 64 requests, is stored in the subsidiary store; when the drum transfer routine is entered on completion of a transfer, the next transfer in the queue is initiated.

Whenever the supervisor wishes to enter another request for a drum transfer, three possible situations arise. Firstly, the queue is empty and the drum transfer can be started immediately. Secondly, the queue is already partly filled and the request is entered in the next position in the queue. Thirdly, the queue is full. In this case the routine making the request is halted by the co-ordinator routine, and is resumed when the queue can receive another entry. In the first two cases the supervisor routine is concluded when the request reaches the queue.

A non-equivalence interruption, which implies a drum transfer is required, is dealt with as follows. The core store is arranged to always hold an empty page with no useful information in it, and when required, a transfer of a block of information from the drum to this empty page is initiated. Whilst this drum transfer is proceeding, preparation is made to write up the contents of another page of core store to the drum to maintain an empty page. The choice of this page is the task of the "learning program" which keeps details of the use made of blocks of information. This learning program predicts the page which will not be required for the largest time, and is arranged with a feed-back so that if it writes up a block which is almost immediately required again, it only does this once. The number of the chosen page is recorded in the subsidiary store and the drum queue entry is converted to a request to write this page to the drum. This supervisor routine is now concluded and returns control to the co-ordinator routine.

When the drum transfer is completed, the drum transfer routine is again entered. This updates the block directory and page address register, makes the object program free to proceed and initiates the next drum request, which is to write the chosen page to the drum. This routine is now concluded and the co-ordinator is re-entered. The supervisor is finally entered when the write to drum transfer is complete. The block directory is updated, a note is made of the empty page, and the next drum request is initiated.

The Use of Main Store by the Supervisor

Some routines of the supervisor are obeyed in the main store and these and others use working space in the main store. Since the supervisor is entered without a complete program change, special care must be taken to keep these blocks of store distinct and protected from interference. The active supervisor blocks of main store are recorded in the area for program 0 in the block directory. There are also some blocks of the supervisor program which are stored permanently on the drum; when one of these permanent blocks is required, it is duplicated to form an active block of the supervisor or, as in the case of a compiler, to become part of an object program.

Of the possible 2048 block numbers, 256 are "reserved" block numbers which are used exclusively by the supervisor and are not available to object programs; object programs are restricted to using the remaining "non-reserved" block numbers. Blocks with reserved block numbers may be used in the core store at any time by the supervisor, and the co-ordinator routine locks out these pages of core store before returning control to an object program. The supervisor also uses some blocks having non-reserved block numbers to keep a record of sequence of blocks of information such as input and output streams. When a non-reserved supervisor block is called to the core store, the page address register is not set, since there may be a block of an object program which has the same block number already in the core store. Instead, the page address register is set to a fixed reserved block number while it is in use, and is cleared and locked out before control passes to another routine.

Not all the reserved block numbers are available to the supervisor for general use, since certain block numbers are temporarily used when drum, tape and peripheral transfers are proceeding. These block numbers do not appear in the block directory. For example, when a magnetic tape transfer is taking place, the page of core store is temporarily given a block number which is recognised by the hardware associated with that tape channel. When the transfer is complete, the appropriate block number is restored. During a peripheral transfer, and also on other occasions, it is necessary that a block should be retained in the core store and should not be transferred to the drum. The relevant page of core store is "locked down" by setting a digit in the subsidiary store; this is referred to by the learning program which never selects for transfer to the drum a page for which this lock-down digit is set.

SECTION 4. The Magnetic Tape Routines

4.1 Magnetic Tape Supervisor Routines

The Magnetic Tape Facilities

The tape mechanism used on Atlas is the Ampex TM2 (improved FR 300) using one inch wide magnetic tape. There are sixteen tracks across the tape-twelve information tracks, two clock tracks, and two tracks used for reference purposes. The tapes are used in a fixed-block, pre-addressed mode. Information is stored on tape in blocks of 512 forty-eight bit words, together with a twenty-four bit checksum with end around carry. Each block is preceded by a block address and block marker and terminated by a block marker; the leading block address is sequential along the tape, and what is effectively the trailing block address is always zero. Tapes are tested and pre-addressed by special routines before being put into use, and the fixed position of the addresses permits selective overwriting and simple omission of faulty patches on the tape. Blocks can be read when the tape is moving either in the forward or reverse direction, but writing is only possible when the tape is moving forward. The double read and write head is used to check read when writing on the tape. When not operating the tape stops with the read head midway between blocks.

Atlas may control a maximum of 32 magnetic tape mechanisms. Each mechanism is connected to the central computer via one of eight channels, all of which can operate simultaneously, each controlling one read, write or positioning operation. It is possible for each tape mechanism to be attached to either one of a pair of channels, the switching being under the control of supervisory programs through digits in the V-store. Fast wind and rewind operations are autonomous and only need the channel to initiate and, if required, terminate them. Transfer of a 512-word block of information between core store and tape is effected via a one-word buffer, the central computer hesitating for about $\frac{1}{2}$ μsec., on average, each time a word is transferred to or from the core store. During a transfer the page of core store is given a particular reserved block number and the contents of the page address register are restored at the end of the transfer.

Supervisory programs are only entered when the block addresses are read before and after each block, and when the tape stops. As each block address is read, it is recorded in the V-store and an interrupt flip-flop is set, causing entrance to the block address interrupt routine.

The Block Address Interrupt Routine

This routine is responsible for initiating and checking the transfer of a single block between tape and core store, and searching along the tape for a specified block address. Digits are available in the V-store to control the speed and direction of motion of the tape and the starting and termination of read or write transfers. The block addresses are checked throughout and, in particular, a write transfer is not started until the leading block address of the tape block involved has been read and checked.

Hardware checking is provided on all transfers, and is acted upon by supervisor routines. A 24-bit check sum is formed and checked as each block is transferred to or from a tape, and a digit is set in the V-store if any failure is detected. Similarly a digit is set in the event of failure to transfer a full block of 512 words. These digits are tested by the block address interrupt routine on the conclusion of each transfer. Parity failure either on reading from core store or on formation of the parity during a transfer to core store causes the setting of interrupt flip-flops. If a tape fails to stop, this is detected by the block address interrupt routine as a particular case of block address failure. Failure to enter the block address routine (for example, through failure to read block markers) is detected by the timed interrupt routine at intervals of 100 milliseconds. Finally, failures of the tape mechanism, such as vacuum failure, set a separate interrupt flip-flop. The detection of any of these errors causes entry to tape monitor routines.

Organisation of Tape Operations

Magnetic tape operations are initiated by entrance to the tape supervisor routines in the fixed store from extracode instructions in an object program or, if the supervisor requires the tape operation for its own purposes, from supervisor extracode routines. From a table in the subsidiary store, the logical tape number used in a program is converted to the actual mechanism number, and the tape "order" is entered in a queue of such orders, in the subsidiary store, awaiting execution. A tape order may consist of the transfer of several blocks and any store blocks involved are "locked out" to prevent subsequent use before completion of the transfers; if any block is already involved in a transfer, the program initiating the request is halted. Similarly, the program is halted if the queue of tape instructions is already full. If the channels to which the deck can be connected are already occupied in a transfer or positioning, the tape supervisor returns control to the object program, which is then free to proceed. A program may thus request a number of tape transfers without being halted, allowing virtually the maximum possible overlap between the central computer and the tape mechanisms during execution of a program. Should a channel be available at the time a tape order is entered to the queue, the order is initiated at once by writing appropriate digits to the V-store, and by writing reserved tape transfer block numbers to the appropriate page address registers if the order involves a read or write transfer. The tape supervisor then returns control to the object program or supervisor routine.

One composite queue of tape orders is used for orders relating to all tape mechanisms and orders are extracted from the queue by S.E.R.'s entered from the block interrupt routine. On reading the penultimate block address involved in an operation (for example, the last leading block address in a forward transfer) the next operation for the channel is located, and if it involves the same mechanism as the current order, and tape motion in the same direction, the operation is "prepared" by calling any store block involved to core store. On reading the final block address and successfully concluding checks, the block address interrupt routine initiates the next operation immediately if one has been prepared, thus avoiding stopping the tape if possible. If no operation has been prepared, the interrupt routine stops the tape by setting a digit in the V-store, and a further "block address interruption" occurs when the tape is stopped and the channel can accept further orders. This interruption enters an S.E.R. which extracts the next order for the channel from the tape queue, and the cycle of events is repeated until no further order for this channel remains. As each transfer is concluded, any object program halted through reference to the store block is made free to proceed.

An exception to the above process is when a long movement (over 200 blocks) or a rewind is required. In this case, the movement is carried out at fast speed, with block address interruptions inhibited, and the channel may meanwhile be used to control another tape mechanism. The long movement is terminated by checking the elapsed time and at the appropriate moment, entering the tape supervisor from the timed interrupt routine. The mechanism is then brought back "on channel" and the speed is returned to normal. When reading of block addresses is correctly resumed, the search is continued in the normal manner.

The Title Block

The first block on each magnetic tape is reserved for use by the supervisor, and access to information in this block by an object program is through special instructions only. This block contains the title of the tape, or an indication that the tape is free. When magnetic tapes are required by the supervisor or by an object program, the supervisor prints instructions to the operator to load the named tape and to engage the mechanism on which it is loaded. The engage button of each mechanism is attached to a digit in the V-store, and these digits are scanned by the supervisor every one second. When a change to "engaged" status has been detected, the tape supervisor is entered to read the first block from the tape. The title is then checked against the expected title. In this way, the presence of the correct tape is verified, and furthermore the tape bearing the title becomes associated with a particular mechanism. Since the programmer assigns a logical tape number to the tape bearing a given title, this logical tape number used in extracode instructions can be converted by the supervisor to the actual mechanism number. Other supervisory information is included in the first block on each tape, including a system tape number and the number of blocks on the tape. Special supervisory routines allow Atlas to read tapes produced on the Orion computer, which uses the same tape mechanism but can write blocks of varying lengths on the tape. These tapes are distinguished on Atlas by a marker written in the title block.

Magnetic Tape Failures

All failures detected by the interrupt routines cause the block address interrupt routine to stop the tape at the end of the current block when possible, and then to enter tape monitor supervisory routines; if the tape cannot be stopped, it is disengaged and the tape monitor routines entered. These routines are S.E.R.'s designed to minimise the immediate effect on the central computer of isolated errors in the tape system, to inform maintenance engineers of any faults, and to diagnose as far as possible the source of a failure. As an example of the actions taken by monitor routines, suppose a check sum failure has been detected whilst reading a block from tape to core store. The tape monitor routines make up to two further attempts to read the block; if either succeeds, the normal tape supervisor is re-entered after informing the engineers. Repeated failure may be caused by the tape or the tape mechanism; to distinguish these, the tape is rewound and an attempt is made to read the first block. If this is successful, a tape error is indicated, and an attempt is made to read the suspect block with reduced bias level. Failure causes the mechanism to be disengaged and the program using the tape to be suspended. If the "recover read" is successful, the tape is copied to a free tape and the operator instructed to re-address the faulty tape, omitting the particular block which failed. If on rewinding the tape, the first block cannot be read successfully, failure in the tape mechanism is suspected and the operator is instructed to remount the tape on another mechanism. Other faults are monitored in a similar manner, and throughout, the operator and engineers are informed of any detected faults. Provision is made for the program using the tape to "trap" persistent tape errors and thereby to take action suitable to the particular problem, which may be more straight-forward and efficient than the standard supervisory action.

Addressing of new tapes and re-addressing of faulty tapes are carried out on the computer by supervisory routines called in by the operator. A tape mechanism is switched to "addressing mode", which prohibits transfers to and from the core store, permits writing from the computer to the reference tracks and to the block addresses on tape, and activates a timing mechanism to space the block addresses. When a new tape is addressed, addresses are written sequentially along the tape and the area between leading and trailing block addresses is checked by writing ones to all digit positions and detecting failures on reading back. Any block causing failure is erased and the tape spaced suitably. On completion, a special block address is written to indicate "end of tape" and the entire tape is then checked by reading backwards. Any failure causes entry to the re-addressing routine. Finally, the tape mechanism is returned to "normal" mode, a title block is written containing the number of blocks on tape, a tape number, and the title "Free", and the tape is made available for use. A tape containing faulty blocks is re-addressed, omitting such blocks, by entry to the re-addressing routine with a list of faulty blocks; the faulty blocks are erased and the remaining blocks are re-labelled sequentially, the tape being checked as when addressing a new tape.

SECTION 5. Peripheral Equipment Routines

5.1 General use of Peripheral Equipments

Peripheral Interruptions

A large number and variety of peripheral equipments may be attached to Atlas, but the amount of electronics associated with each equipment is kept to a minimum, and use is made of the high computing speed and interruption facilities to provide control of these equipments and large scale buffering.

Thus the TR5 paper tape readers, which operate at 300 characters per second, set an interrupt flip-flop whenever a new character appears (characters may be either 5 or 7 bits depending on which of the two alternative widths of tape is being read). Similarly, the paper tape punches, and the teleprinters which print information for the computer operators, cause an interruption whenever they are ready to receive a new character; these equipments operate at 110 and 10 characters per second respectively.

The card readers read 600 cards per minute column by column and interrupt the computer for every column. The card punches, at 100 cards per minute, punch by rows and interrupt for each row.

The Anelex printers have a print barrel, containing 64 different characters, rotating at 1000 revolutions per minute and there are 120 print positions spaced along the print barrel, a complete line being printed at a time. An interruption occurs when the Anelex buffer store is ready to receive further information. A single line feed can be completed in a quarter of a revolution and if only 48 consecutive characters on the print barrel are being used it is possible to print 1000 lines per minute. If all 64 characters are used 4 lines may be printed every 5 revolutions of the print barrel i.e. 800 lines per minute.

Information is received from, or sent to, these peripheral equipments via particular digit positions in the V store. For example, there are 7 such bits for each tape reader, and 80 for each card punch, together with a few more digits for control signals.

The majority of interruptions can be dealt with simply by the interrupt routine for the particular type of equipment. Thus the paper tape reader interrupt routine normally detects terminating characters and makes a parity check and, provided all is well, stores the character to await code conversion by the P.E.R. On output the characters are converted to the correct code by the P.E.R. before the interruption occurs.

The card routines however are complicated by the check reading stations; punching is checked one card cycle afterwards, and reading is checked one colum later. The interrupt routines apply these checks and in the event of failure a monitor SER is entered.

5.1 Continued

Attention by Operators

Whenever an equipment needs attention it is ''disengaged''from the computer. In this state, which is indicated by a light on the equipment and a corresponding bit in the V-store, it automatically stops and cannot be started by the computer.

The operator may engage or disengage an equipment by means of two buttons so labelled. The equipment may also be disengaged by the computer by writing to the appropriate V-store bit, but the computer cannot engage a peripheral.

The ''engage''and ''disengage''buttons do not themselves cause interruptions of the central computer. Instead, the ''engaged'' bits in the V-store are examined every second (this routine is activated by the clock interruption) and any change activates the appropriate S.E.R. In certain cases disengaging a device does not immediately inhibit its interruptions, so that if the operator disengages a card machine in mid-cycle to replenish the magazine or to empty the stacker, the cycle is completed correctly.

There are also other special controls for particular equipments, e.g. a run-out key on card machines, and a 5/7 channel tape width selector switch on punched tape readers.

Most devices have detectors that indicate when cards or paper are exhausted or running low. These have corresponding bits in the V-store that are read by the appropiate S.E.R. The paper tape readers however have no such detector, and the unlikely event of a punched tape passing completely through a reader (due to the absence of terminating characters) appears to the computer merely as a failure to encounter a further character within the normal time interval. This condition is detected by the one- second interrupt routine.

Store Organisation of Input and Output Information

In general, input information is converted to a standard 6-bit internal character code by the PER routine concerned and placed in the store 8 characters to a word. (Exceptions to this occur (a) in the case of card readers when they are reading cards not punched in a standard code, in which case the 12 bits from one column are simply copied into the store and occupy two character positions and (b) on reading 7-hole punched tape, when this is used to convey 7 information bits without a parity check. Such information is distinguished by warning characters, both on the input medium and in the store).

A certain amount of supervisor working space in the core store is set aside to receive this information from the interrupt routines, and is subdivided between the various input peripherals. The amount of this space depends on the number and type of perpherals attached; the first two Atlas computers will normally use one block (512 words). This block will be locked down in a page of the core store whenever any input peripheral is operating (i.e. most of the time).

5.1 Continued

As each input equipment fills its share of this block, the information is copied by an S.E.R. into another block devoted exclusively to that equipment. These copying operations are sufficiently rare so that the latter block need not remain in the core store in the meantime; in fact it is subjected to the same treatment as object programs by the drum transfer routine, and may well be put onto a drum and brought back again for the next copying operation. Thus only one page of core store is used full time during input operations, but nevertheless each input stream finds its way into a separate set of blocks in the store.

The page that is shared between input peripherals is sub-divided in such a way as to minimise the number of occasions on which information must be copied to other blocks; it turns out that the space for each equipment needs to be roughly proportional to the square root of its information rate.

Similarly, information intended for output is placed in a common output page, subdivided for the various output devices, and is taken from there by the interrupt routines as required. As soon as the information for a particular device is exhausted, a P.E.R. is activated to copy fresh information into the common output page. The P.E.R. converts the internal code character into the code used by the device and in the case of the card punch forms an image of the card as the information is required in rows of bits.

As for input the common output page is subdivided roughly in proportion to the square roots of the information rates.

5.2 Peripheral Input and Output

1. Brief Description

Peripheral input and output is done by a group of subroutines, some obeyed in interrupt control and some in extracode control. Between them they arrange to read characters from any input peripheral and to feed them as a string of 6-bit characters into the store. Transfer to the store is in two stages. First under interrupt control, usually one character at a time, into a small buffer locked down in the main core store. Second, when this buffer has been filled, the characters are transferred under extracode control into a normal main store block. Conversion to the internal code, if required, is done during the second stage. A binary mode of operation is also available and in this there is no conversion at the second stage, and the information is stored as consecutive 12-bit characters.

Output is merely the reverse procedure. Conversion to the code of the particular output peripheral is performed under extracode control while loading the small output buffer. Characters are then sent from this, one at a time, to the peripheral under interrupt control.

The system is intended to handle paper tape or cards on input, or paper tape, cards, teleprinter or line printer on output, without distinction between them.

2. Working Space

Each peripheral has three distinct requirements for working space. These are:-

(a) A private store in the subsidiary store for counters markers etc. (about 12 consecutive half words for each peripheral). Though expected to be in the subsidiary store, the routines will also function if this is locked down in the main core store.

(b) An input or output buffer of about $\frac{1}{8}$ to $\frac{1}{4}$ of a block per peripheral (minimum size: one half word for paper tape equipment, 80 half words for card reader). This is expected to be in the main store, and if there, must remain locked down for the whole of the period that the peripheral is operating. It must not be part of one of the main store blocks used for purpose (c) below. The buffer may alternatively be in subsidiary store, and in this case there is no restriction over sharing blocks between requirements (b) and (c)

(c) A block which could be anywhere on the drum, core or fixed stores to which the string of characters is finally sent on input, or from which it is drawn for output.

3. Location of the Routines

The routines are intended for loading into the fixed store. They may also be run in main or subsidiary store. If in main store the routines must not share one of the blocks used for purpose (c) above.

4. Subsidiary Store Address and Private Store

In addition to the working space, each slow peripheral equipment has a V-store address and a subsidiary store address. These are fixed in the sense that one is a built in feature of the computer and the other would require the fixed store programmes to be rewritten if it was to be changed.

The V-store address is the address of the V-store register containing the start, stop, disengaged digits etc.

The subsidiary store address is one half word in the subsidiary store, and it contains the address of a small group of consecutive words used as private store by this particular peripheral equipment.

As a cross-reference the first half word of private store of any peripheral will contain its V-store address. (less *6). The type of equipment, or its number within that type, can be determined by examining the digits of the V-store address.

Within the computer a peripheral is generally identified by the address of the beginning of its private store. This is not a fixed address and may well be changed from day to day.

There is a routine R501 which can be used at the beginning of a day to load the private store with the necessary initial constants

5. Input Buffer and Main Store

For each peripheral, the input buffer should, preferably, be large enough to accommodate all the characters read in during a period of about one second. The interrupting routine takes the characters from the reader and places them, one per half word, in the input buffer. When the buffer is filled, the reader is stopped and a Peripheral Extracode Routine is entered. If translation into the internal code has been specified, the P.E.R. performs the code conversion and packs the resulting 6-bit characters, 4 per half word. If the binary mode has been specified there is not code conversion and the characters are treated as being of 12 bits, and are packed 2 per half word. When the P.E.R. has exhausted the information in the buffer, the reader is started again to refill it. After several such cycles, the area specified in the main store block will eventually be filled and an exit is made to the return address requested by the Input Master Routine.

Owing to the fact that the number of word spaces in the store does not need to be a multiple of the number of words in the input buffer, there will generally be several characters left in the input buffer after a return has been made to the main programme. These will appear at the beginning of the next batch of characters when another request to read from the peripheral is made.

Other reasons for returning to the Input Master Routine during input are:

a) If a punching fault is detected.

b) If attention to the L.A.M. of the reader is overdue.

c) If a mechanical failure occurs on the reader.

d) If a sequence of three asterisks is detected.

e) If a paper tape has run out or a pack of cards has been completed.

These conditions are all detected at the time of the interrupting routines. The reader is stopped forthwith, and those characters which were read up to the time of stopping are transferred to the main store in the usual way. In this case there will be no characters left in the buffer when a return is made to the Input Master Routine.

6. Output Buffer

The output buffer for a peripheral should also, preferably, be large enough to accommodate as many characters as it can print in about one second.

The material for output may be in internal code or binary, but in either case the first half word must always be a separator.

The P.E.R. unpacks the characters, converts into the code of the peripheral if required and places them in the output buffer. When the buffer is full the peripheral is started and the contents of the buffer are sent to the peripheral by interrupting routines. When the buffer has been emptied, the P.E.R is called in again to fill it and the process is repeated. A return to the address specified by the Output Master Routine is finally made after the buffer has been emptied for the last time, and there are no further characters to print.

Other reasons for returning to the Output Master Routine during output are:

a) If attention to the L.A.M. of a punch is overdue.

b) If a mechanical failure occurs

c) If the supply of paper, paper tape, or cards runs out.

These conditions are detected during the period when the interrupting routines are sending information to the peripheral, and generally occur with the output buffer only partially emptied. A special entry to the routines permits output to be resumed at the point where it was left off. The normal entry should be used if the previous output is to be abandoned and a fresh start is to be made.

7. Engaging and Disengaging

7.1 Input Peripherals

The state of the peripheral, stopped or started, engaged or disengaged is examined by the One Second interruption routine, every second. When a reader is ready to be started an addrss of the Input Master Routine is inserted in the S.E.R. queue, and this routine, when it comes to be obeyed, must take the appropriate action of providing space in the main store and informing the operator etc. It should then cause an entry to R502 which starts the reader and transfers its information to the main store.

Once the Input Master Routine has been called to the S.E.R. queue the reader is reserved, and the One Second routine will give no further advice about the state of it, until it has been freed. In between reading blocks of material the reader can therefore be left standing idle, and there will be no spurious calls to start it from the One Second routine. When it has eventually been finished with, however, it is essential for the reader to be freed. A subroutine R504 is provided for doing this, Also, if the operator disengages the reader in the middle of reading, it automatically becomes free.

The Input Master Routine is called in when the reader is ready to be started, but the reason for having stopped last time is not given. This may have been because:

a) The last input run was successfully completed. The reader has now been reloaded and is waiting to be started.

or b) The operator pressed the disengaged button and has now re-engaged the reader. The previous input is to be continued.

c) The operator pressed the disengage button and has now reloaded the reader and engaged it again.

or d) The last input run ended in a fault, and the operator has now reloaded the reader and engaged it again.

The Input Master Routine controlling input must be able to recognise these possibilities, and know what to do with the material which is going to come in from the reader.

7.2 Output Peripheral

Output may be inititated, by use of the subroutine R503, whenever there is any information available for printing. No check is made during this subroutine of whether the peripheral is engaged or not (or whether there is even anything attached to that socket on the computer at all) and the Output Master Routine should therefore test the Disengaged digit before entering R503.

There is no way of preventing the operator disengaging the peripheral immediately after this test. If she does so and subsequently engages it again, the One Second routine will detect this amd immediately start up the printer to continue where it left off. A return will not be made to the requested exit address until all the character specified have been printed, or a fault occurs.

8. Subroutines Intended for use by other parts of the Supervisor

R501	Load private store of any peripheral
R502	Start reading from any input peripheral
R503	Start writing to any output peripheral
R504	Free any input peripheral
R508	Peripheral one second
R509	Find peripheral type

SECTION 6. The Operating System

6.1. Documents

Input

The fast computing speed of Atlas and the use of multiple input and output peripheral equipments enable the computer to handle a large quantity and variety of problems. These range from small jobs for which there is no data outside the program itself, to large jobs requiring several batches of data, possibly arriving on different media. Other input items may consist of amendments to programs, or requests to execute programs already supplied. Several such items may be submitted together on one deck of cards or length of punched tape. All must be properly identified for the computer.

To systematise this identification task, the concept of a "document" has been introduced. A document is a self-contained section of input information, presented to the computer consecutively through one input channel. Each document carries suitable identifying information (see below) and the supervisor keeps in the main store a list of the documents as they are accepted into the store by the input routines, and a list of jobs for which further documents are awaited.

A job may require several documents, and only when all these have been supplied can execution begin. The supervisor therefore checks the appearance of documents for each job; when they are complete the job scheduling routine is notified (see below).

Normally, the main core and drum store of the computer is unlikely to suffice to hold all the documents that are waiting to be used. The blocks of input information are therefore copied, as they are received, onto a magnetic tape belonging to the supervisor, called the "system input tape". Hence, if it becomes necessary for the supervisor to erase them from the main store, they can be recovered from the system input tape when the job is ready for execution.

The system input tape thus acts as a large scale buffer, and indeed it plays a similar part to that of the system input tape in more conventional systems. The differences here are that the tape is prepared by the computer itself instead of by off-line equipment, and that there is no tape-handling or manual supervision required after the input of the original documents - an important point in a system designed to handle many miscellaneous jobs.

This complete bufferage system for input documents is called the "input well". Documents awaiting further documents before they can be used are said to be in "input well A"; complete sets of documents for jobs form "input well B". Usually documents being accepted into input well B must be read from the system input tape back into the main store so that they are ready for execution; often however they will already be in input well A in the main store, so that only an adjustment of the block directory is required.

One result of this arrangement is that the same tape is being used both to write input blocks, in a consecutive sequence, and to read back previously written blocks to recover particular documents as they are required. The tape will therefore make frequent scans over a few feet of tape, although it will gradually progress forwards. The lengths of these scans are related to the main store space occupied by input well A. For example, so long as the scans do not exceed about 80 feet (130 blocks) the waiting time for writing fresh blocks will remain less than the time for input of three blocks from a card reader, so that comparatively little main store space need be occupied by input well A. To ensure that scans are kept down to a reasonable limit, any documents left on the system input tape for so long that they are approaching the limit of the scannable area are copied to the system dump tape (see below). If the number of these becomes large, the computer operators are warned to reduce the supply of documents through the input peripherals.

Output

The central computer can produce output at a much greater rate than the peripheral equipments can receive it, and an "output well" is used in a manner analogous to the input well. This well uses a "system output tape" to provide bulk buffering.

Output for all output peripherals is put onto the same tape, arranged in sections that are subdivided so that the contents of a section will occupy all currently operating peripherals for the same length of time. Thus if, for example, a burst of output is generated for a particular peripheral, it is spaced out on the system output tape, leaving spare blocks to be filled in later with output for other peripherals (this is possible because Atlas uses pre-addressed tape). In this way, the recovery of information from the tape into "output well B" as required by the various peripherals merely involves reading complete sections from the tape.

Again, there is a limit to the amount of information that can usefully be buffered on the output tape, due to the time required to scan back and forth between writing and reading regions, and this limit depends on the space available in the main store for output well B. An S.E.R. keeps a check on the amount of information remaining in output well B for each equipment, and relates this to the present scan distance to decide when to start to move the tape back for the next reading operation. If the amount of output being generated by object programs becomes too great some of it is put instead on the dump tape (see below) or a program is suspended.

The System Dump Tape

The system input and output tapes operate essentially as extensions of the main store of the computer. Broadly speaking, documents are read into the computer, programs are executed, and output is produced. The fact that the input and output usually spends some time on magnetic tape is, in a sense, incidental. This input and output buffering is, however, a continuous and specialised requirement, so that a particular way of using these tapes has been developed and special S.E.R.'s have been written to control them.

When demands on storage exceed the capacity of the main store and input and output tapes, a separate magnetic tape, the system dumptape, is used to hold information not required immediately. This tape may be called into use for a variety of reasons. Execution of a problem may be suspended and the problem recorded temporarily on the dump tape if other problems are required to fill the output well, or alternatively if its own output cannot be accommodated in the output well. Also, as already described, the output wells can "overflow" to the system dump tape. This tape is not used in a systematic manner, but is used to deal with emergencies. However, the system is such that, if necessary, the system input and output tapes can be dispensed with, thereby reducing the input and output wells and increasing the load on the system dump tape. In an extreme case, the system dump tape itself can be dispensed with, implying a further reduction in the efficiency of the system.

Headings and Titles

Every input document is preceded by its identifying information, mentioned above. This consists of two lines of printing, forming the heading and the title respectively.

The heading indicates which type of document follows. The most common headings are:

COMPILER followed by the name of a program language, which means that the document is a program in the stated language:

DATA which means that the document is data required by an object program; and

JOB which means that the document is a request for the computer to execute a job, and gives some relevant facts about it.

The last type of document is called a "job description". It gives, for example, a list of all other documents required for the job, a list of output streams produced, any magnetic tapes required, and upper limits to the storage space and computing time required. Many of these details are optional; for example if storage space and computing time are not quoted a standard allowance will be made.

For example, if a program operates on two data documents which it refers to as data 1 and data 2, the job description would contain:

INPUT

1 followed by the title of data 1

2 followed by the title of data 2

The program would appear in this list as data 0. Alternatively, a job description may be combined with a program, forming one composite document, and this will usually happen with small jobs.

Each output stream may be assigned to a particular peripheral or type of peripheral, or may be allowed to appear on any output equipment. The amount of output in each stream may also be specified. It is worth noting here that the organisation of the output well is such that it can readily accept two or more streams of output from a program destined for the same equipment, even though only one such equipment may exist. The streams are accumulated in the output well independently and are eventually output one after another.

For example, a description may include:

OUTPUT

1 LINE PRINTER 20 BLOCKS

2 CARDS

3 ANY

Each magnetic tape used by a program is identified by a number within the program, and the job description contains a list of these numbers with the title that appears in block 0 of each tape to identify it; for example:

TAPE

1 POTENTIAL FIELD CYLIND/204/TPU5

If a new tape is required, a free tape must be loaded, which the program may then adopt and give a new title. This is indicated thus:

TAPE FREE

2 MONTE CARLO RESULTS K49-REAC-OR4

The loading of tapes by operators is requested by the supervisor acting on the information in job descriptions.

Finally, the end of a document is indicated by

* * *

and if this is also the end of the punched tape or deck of cards it is followed by the letter Z. On reading this the computer disengages the equipment.

Logging and charging for Machine Time

As problems are completed, various items of information on the performance of the computing system are accumulated by the supervisor. Items such as the number of program changes and the number of drum transfers are accumulated and also, for each job, the number of instructions obeyed, the time spent on input and output, and the use made of magnetic tapes. These items are printed in batches to provide the operators with a record of computer performance, and they are also needed for assessing machine charges.

The method of calculating charges may well vary between different installations, but one desirable feature of any method is that the charge for running a program should not vary significantly from one run to another. One difficulty is that the number of drum transfers required in a program may vary considerably with the amount of core store which is being used at the same time for magnetic tape and peripheral transfers. One method of calculating the charge so as not to reflect this variation is to make no charge for drum transfers, but to base the charge for computing time on the number of instructions obeyed in a program. This, however, gives no incentive to a programmer to arrange a program so as to reduce its drum transfers, and more elaborate schemes may eventually be devised. The charge for using peripherals for input and output can be calculated from the amount of input and output. For magnetic tapes, the charge can be based on the length of time for which the tape mechanism is engaged, allowance being made for the time when the program is free to proceed but is held up by a program of higher priority. All this information is made available to the S.E.R. responsible for the costing of jobs.

Methods of Using the Operating System

The normal method of operating the computer is for documents to be loaded on any peripheral equipment in any order, although usually related documents will be loaded around the same time. The titles and job descriptions enable the supervisor program to assemble and execute complete programs, and the output is distributed on all the available peripherals. Usually programs are compiled and executed in the same order as the input is completed, but the supervisor may vary this depending on the load on different parts of the system. For example, a problem requiring magnetic tape mechanisms which are already in use may be by-passed in favour of a problem using an idle output peripheral; a problem which computes for a long time may be temporarily suspended in order to increase the load on the output peripherals. By these and similar methods, the S.E.R. responsible for scheduling attempts to maintain the fullest possible activity of the output peripherals, the magnetic tape mechanisms and the central computer.

Documents may also be supplied to the computer from magnetic tapes; these tapes may be either previous system input tapes or library tapes or tapes on which "standard", frequently used, programs are stored. Such documents are regarded as forming part of Input Well B and are read into main store when required. An alternative method of operating may be to use the computer to copy documents to a "private" magnetic tape, rather than to use the system input tape, and at a later time to supply the computer with a succession of jobs from this tape. Similarly, output may be accumulated on a private magnetic tape and later passed through the computer to one or more peripheral equipments. Routines forming part of the supervisor are available to carry out such standard "copying" operations.

Provision is also made for the chief operator to modify the system in various way; for example, priority may be given to a particular job, or a peripheral equipment may be removed from general use and allocated a particular task. An "isolated" operating station may, for example, be established by reserving a particular output equipment for use by problems loaded on a particular input equipment.

6.2 Major Routines of the Operating System

6.2.1 Co-ordination of the Operating System

The routines comprising the operating system control the initiation and termination of object programs, the passage of input and output information between peripheral equipments, tapes, and object programs, and the allocation of peripherals, tapes and store. The system forms a "program" of many branches, several of which can be active at any one time, although, of course, only one branch is actually obeyed by the central computer at any one time. Each major branch or routine is composed of a sequence of supervisor extracode routines (S.E.R.'S). The co-ordinator in fixed store organises the initiation of these routines, queing of halted routines etc. At any one time only one S.E.R. is being obeyed; others may be.

a) Inactive
b) in S.E.R. queues awaiting entry
c) halted in S.E.R. queues
d) effectively halted in drum, tape, or peripheral queues.

A diagram of the operating system is shown in Fig. A. Only the major subdivision into branches is shown, and the "normal" flow of control between them. Subdivision of each routine into a sequence of S.E.R.'s, only one of which is initiated at any one time for each branch, will be described later. In the interests of clarity, routines entered from several other routines are listed separately and are not included in the flow of control. The normal flow of control may be interrupted by timed routines, operators intervention, or hardware failure. The various ways these enter the normal flow of control are not indicated in the diagram.

The major routines shown in Fig. A are described briefly in the following section. The purpose of each routine is described, and inter-connection between the routines is indicated. In common with all branched programs, the inter-connection between branches which may be concurrently active does not take the form of a simple transfer of control. Suppose branch A wishes to call in branch B. If B is inactive, it may be activated to operate in parallel with branch A (that is, planted in an S.E.R. queue to be entered when branch A is concluded or halted). If branch B is already active but halted or awaiting entry, branch A must leave indicators in store to be acted upon by branch B at a suitable point in its cycle before it becomes inactive. This is the method used to activate one branch from another, and in this way, the number of branches concurrently active has a finite upper bound, even though one branch may be "called" at random by many other branches.

As a typical job passes through the system, it is acted on by the various routines shown in Fig. A. Input through peripherals is controlled by the Input Master: program and input data are stored on the system input tape by the Input Control Routine, which also assembles the complete job in store when required by the Active Schedule. It is planted in the Execute List by the Execute Scheduler, is processed and compiled, and then obeyed as an object program. Output is passed to the Output Scheduler, is written to the System output tape by Output Control, and is ultimately passed to the peripherals by the Output Master routine. When the job ends for any reason, it is subject to Post Processing (monitor, logging etc). Typically many jobs will be in transit through the system at any one time, being acted upon by the various routines which can operate concurrently.

The following are commonly used "subroutines" which are not included in the flow of control shown in Fig. A.

Drum Supervisor. This is used by all routines, since either program or data or both occupy main store blocks. S.E.R.'s may request drum transfers to or from core store, and will be halted in the drum queue until completion of the transfer. If the drum queue is full, the routines are first halted in the relevant S.E.R. queue.

Tape Supervisor. All routines handling magnetic tape use the tape supervisor to carry out basic tape operations. The tape supervisor may function concurrently with the routine calling it in (forming a separate branch) or the calling routine may be halted in the tape supervisor until the completion of a particular order. The calling routine may be halted in an S.E.R. queue if the tape queue is full.

Peripheral Supervisor. Routines handling input and output equipments use the peripheral supervisor to carry out basic transfers between peripherals and main store. The calling routine may be halted in the peripheral supervisor until completion of the required transfer or may function in parallel with the peripheral supervisor.

Operators output. This is a subroutine which initiates requests to print information on-line on the operators output. If the output is busy, the calling routine is halted; otherwise the request is passed to the peripheral supervisor, and the calling routine is re-entered.

Space Allocation. This is a subroutine entered when a new block of main store is required for any reason. It arranges the distribution of store blocks according to the priority of routine using blocks. The calling routine may be halted until store blocks become available, either through the natural loss of blocks by another routine or through the action of the space allocation routine in writing blocks to the system dump tape.

Tape Allocation. This routine is initiated whenever there is need for a tape unit or whenever a tape unit becomes free. It arranges the allocation of the available tape units, attempting to "look ahead" as far as possible in order to minimise the effect on the computing system of operator handling time. This routine forms a separate branch of the operating system, being activated to run concurrently with other branches; when halted it occupies a base in the Slow S.E.R. Queue. Entries and exits are listed in the description of other major routines of the operating system.

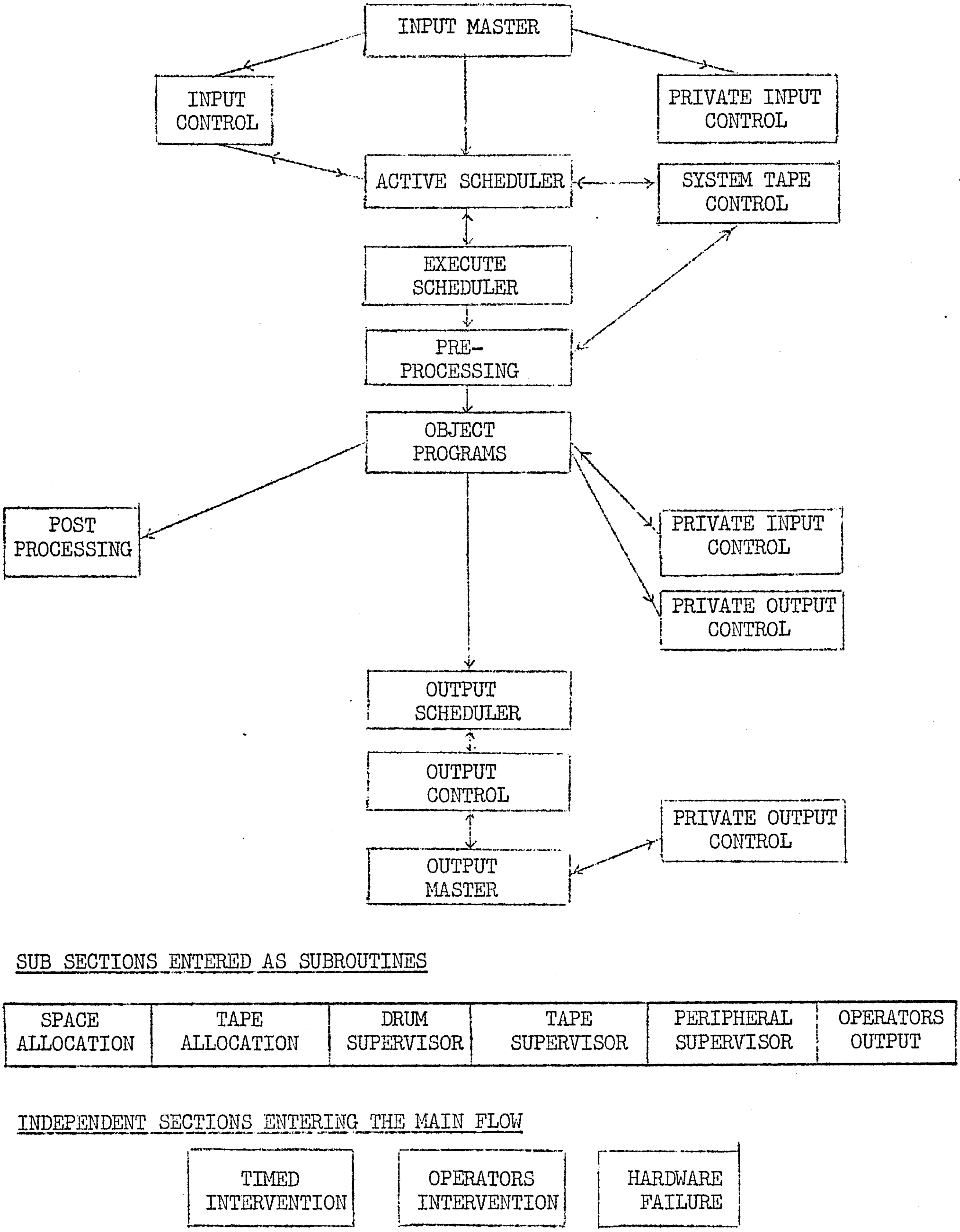


Fig. A. Flow Diagram of the Operating System.

6.2.2 Outline of Major Routines of the Operating System

1) Input Master

Purpose To control the assembly of information from input peripherals to blocks in main store. Information is collected in separate blocks for separate peripherals. When one block is filled, it is linked in store with previous blocks from the same peripheral and the Input Control Routine is activated if necessary to write the block to the System Input Tape. Alternatively, the Private Input Control is activated to write the block to a private tape. The headings and titles of all documents are decoded and lists are compiled and updated of incomplete jobs, unattached documents, and complete jobs (whose peripheral input is complete). A list is also maintained of documents required from other tapes, - previous system input tapes or private tapes. When a job is added to the complete job list, the Active Scheduler is activated to transfer the job to the Execute List if appropriate. If the job requires magnetic tapes, the Tape Allocation Routine is notified, in order to load free tape units whenever possible.

The peripheral supervisor enters the Input Master Routine on any failure of punching or equipment, and the Input Master Routine is responsible for dealing with the failure and controlling restarts. The routine also detects and acts on operators' requests.

Connections with Other Routines

Entered from:	One Second Routine	:	Equipment engaged
	Peripheral Supervisor	:	Block filled, end of document, failure detected.
Exit to :	Peripheral Supervisor	:	Read to end of document or till block filled.
	Input control	:	Block available for system input tape.
	Private input control	:	Block available for private input tape.
	Active scheduler	:	Complete job available.
	Tape allocation	:	Complete job available needing tapes.
Lists :	Complete job list		
	Incomplete job list		
	Unassigned document list		
	System document list		

Organisation The Input Master Routine comprises a set of S.E.R.'s forming an extension of the peripheral input S.E.R.'s. The routine can deal with any number of peripherals operating in parallel. When halted, the routine occupies the base in the Slow S.E.R. queue reserved for a peripheral, and no further S.E.R.'s can be entered from the peripheral interrupt routine until the Input Master Routine is terminated (usually by re-entering the peripheral supervisor). Other routines are activated "in parallel" and are not obeyed until the Input Master Routine is completed or halted; these other routines do not, therefore, hold up peripheral equipments.

2) Input Control

Purpose To control transfers to and from the system input tape. The Input Control Routine is initially activated when the System Input Tape is stationary in the writing position. Firstly any blocks made available by the Input Master are written to tape, each block containing within itself the position on tape of the next block in the input stream. When writing is completed, the tape may be scanned back and documents read off. Documents may be required for jobs waiting in the active list; job descriptions may be required to obtain titles of magnetic tapes required; complete jobs or documents may have to be read from the input tape in order to reduce the area to be scanned. The Input Control routine itself selects these latter documents; the Active Schedules and Tape Allocation routines request the other documents. During the backward motion, job lists are read to store (which are compiled originally by the Input Master routine). These lists contain the location on tape of all documents required, and also a list of tapes required. The latter are made available to the Tape Allocation Routine. The former are collected in a list of required documents. During the forward sweep, required documents are read from the tape into store; jobs completely assembled are marked as such on the Active List, and the Active Schedule is activated if necessary. When writing position is reached again, the entire process is repeated. The Input Control Routine becomes inactive when the tape is in writing position and neither writing nor reading is requested.

Connections with Other Routines

Entered from:	Input Master	:	Block to be written to System Input Tape.
	Active Schedule	:	Job to be assembled in store, job description required.
	Tape Allocation	:	Job list to be read from System Input Tape.
Exit to :	Active Schedule	:	Job collected.
			Job incomplete.
			Job description available.
	Tape Allocation	:	List of tapes available.
Lists :	Documents required for or read to active list.		

Organisation The Input Control Routine forms a sequence of S.E.R.'s based on the System Input Tape, and occupies the corresponding base in the Tape S.E.R. queue. Routines which call it are arranged to activate it only when the Input Tape is idle; otherwise information is left in the Active List or in the Input stream linking which the Input Control routine scans before becoming inactive. The routine may be activated by other routines, entered, for example, from timed interruptions or operators' intervention.

3) System Tape Control

Purpose To read documents to store from previous system tapes, to form part of the input well, as indicated by job descriptions. The routine is initiated by the tape allocation routine when a tape unit is free, and scans the system document list prepared by the Input Master Routine. Documents are read into store in the order in which they occur on tape. When no more documents are required from further along the tape, further documents are read if required after rewinding the tape to the earliest required position. The routine may become inactive when no further documents are required or when insufficient store is available. When input for a job is completed, the job is marked in the Complete Job list. The System Tape Control routine may be restarted by the Active Schedules in order to prepare a specific job.

Connections with Other Routines

Entered from	:	Tape Allocation Routine	:	Mount tape and search System Document list.
		Active Schedules	:	Read documents of a given job.
Exit to	:	Active Schedules	:	Job prepared.
		Tape Allocation Routine	:	Tape no longer required.

Organisation The System Tape Control forms a series of S.E.R.'s based on the entry in the Tape S.E.R. queue for the particular tape being controlled. The routine can operate in parallel, controlling a number of system tapes.

4) Private Input Control

Purpose The Private Input Control is used to copy information from the input well to a private magnetic tape, and to read from a private tape into an object program (by simulation of a peripheral equipment). Blocks are written to a tape as they become available from the Input Master Routine. When reading tape into an object program, a two-block buffer is used, one attached to the program, the other filled from the private tape. Since a private tape will be loaded with one document at a time, occupying successive blocks, the rate at which an object program can demand information can be met with this size of buffer.

Connection with Other Routines

Entered from	:	Tape Allocation Routine	:	Mount private tape for read/write.
		Input Master	:	Write block to tape.
		Object Program	:	Read next block from tape.
Exit to	:	Object Program	:	Block available.

Organisation This is a short sequence of S.E.R.'s which plant orders in the tape queue. They are based either on object programs or on relevant private tape. Once information is made available in store, it is read to a program by peripheral extracode orders.

5) Active Scheduler

Purpose To select jobs from the complete job list for assembly in main store prior to execution. If the actual assembly involves input from magnetic tapes, this assembly is carried out by the Input Control Routines, activated by the Active Scheduler where necessary, and they re-enter the Active Scheduler when a job is completely assembled. The scheduler is entered when a new job is entered to the complete job list, and when a vacancy occurs on the active list (the list of programs prepared for execution). The routine aims at maintaining a back log of available programs sufficient to maintain full activity during one swing of the system input tape (around 16 secs.). If there is a vacancy on the active list, a required job is first selected (e.g. the requirement may be for a tape, peripheral or computer limited job). Once requirements are met, the active list is filled with jobs in order on the complete job list which are not prevented from running through their use of reserved drum bands, use of a large proportion of store, use of tape units, etc. When a job has been assembled by the Input Control Routines, the active scheduler is entered which in turn enters the Execute Scheduler in order to consider the job for immediate execution.

Connections with Other Routines

Entered from	:	Input Master	:	New job available.
			:	Job collected.
			:	Job incomplete .
			:	Job description available.
		Execute Schedule	:	Supply a job.
		System Tape Control	:	System documents collected.
		Tape Allocation	:	Tape units available/mounted.
Exit to	:	Execute Schedule	:	Job available.
		Input Control	:	Job to be assembled in store, job description required.
		System Tape Control	:	Assemble documents of a given job.
		Tape Allocation	:	Acquire tape units for a given job.

Organisation The Active Schedulers form a sequence of S.E.R.'s with their own reserved base in the Slow S.E.R. Queue, which is "booked" when the routine is halted for drum transfers or full drum queue.

6) Execute Scheduler

Purpose To transfer jobs from the Active list to the execute list in subsidiary store and to begin execution by entering the pre-processing routine. The scheduler must select the job to be entered and control its priority in the execute list. The routine is entered whenever a new entry of an assembled job appears on the active list, whenever a vacancy occurs in the execute list (via the Post-Processing Routine), or when the supply to a peripheral output equipment is nearly exhausted. Operator requests for action (such as high priority for a job) effectively enter the routine via entries in the active list when an adjustment of the execute list is required.

Connections with Other Routines

Entered from	:	Active Scheduler	:	Job ready for execution
		Post Processing	:	Vacancy in the execute list
		Output Schedule	:	Output back log low
Exit to	:	Active Scheduler	:	Vacancy created in the active list
		Preprocessing	:	Execute a given job

Organisation This routine has a separate base reserved for it in the Slow S.E.R. Queue, which it occupies when halted for drum transfers. Routines entering it either activate it if it is idle or note the request in store, where it is detected by the execute scheduler.

7) Preprocessing

Purpose After entry of a job into the execute list, this routine decodes the job description and eventually calls in the relevant compiler. The compiler may return to the supervisor routine to read more "job description" and eventaully compilation is concluded. The preprocessing routine scans the parameters of the job amongst which are the execute switch, entry address and location of the program. If the execute switch is set to "execute" blocks used by the compiler are lost; drum sectors are acquired when necessary, store is reserved and the object program is entered under main or extracode control, as preset by the compiler. If any recording of new documents is to be carried out, this is a function of this routine.

Connections with Other Routines

Entered from	:	Execute Scheduler	:	Jobs to be executed
		Compilers and other processors	:	Continue processing or execute
Exit to	:	Compilers etc.	:	Process the selected input stream
		Main program	:	Execute
		End program	:	Do not execute
		System Tape Control	:	Read library documents

Organisation This routine comprises a series of S.E.R.'s and pseudo main programs, using main and extracode control. It forms a logical preliminary to the object program and obeys the rules of an object program, except that during its operation, the "Process switch" of the program is set, permitting exit to extracode control where required.

8) Post Processing

Purpose To conclude the execution phase of a job (or the compiling phase if execution has not been requested). This routine comprises Program monitor routines and "End Program" extracode. Its function is to monitor and "print" where appropriate, print costing information, update the central log, dump the program where appropriate, instruct the operator to disengage and label magnetic tapes, close all output streams, and "lose" all main store blocks connected with the job, including outstanding blocks of input well. It then enters the Execute Scheduler to seek for a replacement, and also activates the Tape Allocation Routine if tape units are made free.

Connections with Other Routines

Entered from	:	Monitor	:	Program "fault" detected
		End Program	:	End execution of program
		Preprocessor	:	Omit "execution" of program
Exit to	:	Execute scheduler	:	Vacancy in execute list
		Output scheduler	:	Output streams closed
		Tape allocation routine	:	Tape units free

Organisation Like the preprocessing routine, this routine is a combination of S.E.R.'s based on the main program and pseudo programs using main or extracode control, with special exit to extracode control when required. Parts of it are entered from the Space Allocation Routine (to dump a program), and by hardware monitor routines to dump a program and/or terminate it.

9) Output Scheduler

Purpose To control the passage of output documents to the output control (and hence to the output equipments) and to maintain a list of the location in store of each output stream. The routine is entered from object programs as the first block of each output stream is completed, in order to maintain a list of output documents, and is entered again on "Break output", when a stream can be sent to output control. The routine is also called by Output Control routine when the supply to a peripheral equipment is exhausted, and is entered by the Space Allocation Routine when the output well of incomplete documents is to be emptied. When a peripheral reaches emergency (low back log) the Output Scheduler enters the Execute Schedule to request another job.

Connections with Other Routines

Entered from	:	Object Programs	:	Output stream started, ended
		Output Control	:	Peripheral supply exhausted
		Space Allocation	:	Output well to be cleared

Exit to	:	Output Control	:	Add output document to an output stream
		Execute Scheduler	:	Emergency on a peripheral

List : Output documents, incomplete and complete - location in store or on dump tape or on system output tape, length, and job number and stream number.

Organisation This routine forms a separate series of S.E.R.'s, with its own base in the Slow S.E.R. queue. A part of the routine is based on the object program initiating output.

10) Output Control

Purpose To control transfers to and from the system output tape. The Output Control Routine is initially activated when the system output tape is stationary in the writing position. Any blocks made available by the Output Scheduler are written on tape, each block containing within itself the position on tape of the next block in the output stream. When writing is completed, blocks may require to be read to the Output Master Routine; the tape is scanned back and the back logs in store are added to. The tape is then moved forwards to the writing position, and the cycle is repeated. The Output Control Routine becomes inactive when the tape is in the writing position and neither writing nor reading is requested.

Connections with Other Routines

Entered from	:	Output scheduler	:	Add to output stream
		Output Master	:	Supply more output
Exit to	:	Output Master	:	Start output
		Output Scheduler	:	Supply more output

Organisation The Output Control Routine forms a sequence of S.E.R.'s based on the System Output Tape, and occupies the corresponding base in the Tape S.E.R. queue. Routines calling it are arranged to activate it only when the Output Tape is idle; otherwise information is left in store which the routine scans before becoming inactive.

11) Private Output Control

Purpose To control transfers from an object program to a private magnetic tape via peripheral extracodes, and to control reading of information from a private or system tape to peripheral output equipments. Blocks are written to tape as they become available from an object program. Blocks are called from a private tape to form part of the output well when called by the Output Master Routine; a two block buffer for each equipment will be used, since the private tape is not involved in both reading and writing operations.

Connection with Other Routines

Entered from	:	Tape Allocation Routine	:	Mount tape for read/write
		Object Programs	:	Write block to tape
		Output Master Routine	:	Read block from tape

Exit to	:	Output Master Routine	:	Block available for printing

Organisation This routine forms a sequence of S.E.R.'s based on the tape being controlled. The tapes are used for reading or writing, though several streams may be written to or read from the tape at any one time.

12) Output Master

Purpose To control the passage of information from main store to the output peripherals, excluding on-line operators output devices. Blocks are passed to the peripheral supervisor for output; if the back log of blocks for any output device thereby becomes too low (the limit is a present parameter of the system) the Output Control Routine is activated to read more blocks from the System Output Tape. If the Output Control is already active, no action is taken as it will ultimately replenish the output back log. If a private tape is supplying a peripheral, the Private Output Control Routine is activated to read another block. The peripheral supervisor returns to Output Master on completion of a block or part block, and on equipment failure; the Output Master supplies another block if possible, and deals with failures and restarts, including "paper low" warnings. If ultimately no more output remains for a peripheral, the peripheral is stopped, and may be subsequently restarted by activation of the Output Master Routine by Output Control Routines.

Connection with Other Routines

Entered from	:	One Second Routine	:	Output peripheral engaged
		Peripheral supervisor	:	Output completed, failure of equipment
		Output Control	:	Start output
		Private Output Control	:	Start output
Exit to	:	Peripheral supervisor	:	Output a block or part of block
		Output Control	:	Supply more output
		Private Output Control	:	Supply more output

Organisation The Output Master Routine comprises a set of S.E.R.'s forming an extension of peripheral output S.E.R.'s. The routine can deal with any number of peripherals operating in parallel. When halted the routine occupies the base in the Slow S.E.R. queue reserved for a peripheral, and no further S.E.R.'s can be entered from the peripheral interrupt routine until the Output Master is terminated (usually by re-entering the peripheral supervisor).

7.1 Purpose of the Monitor Program

The monitor program is a set of routines in fixed store and main store which deals in a general manner with the effect on the course of an object program of detectable errors. It is primarily designed to deal with faults caused by the object program (program faults), but it is also entered following the detection of computer failures of failures in on-line peripheral equipments, such as magnetic tapes, which affect the functioning of the program. The monitor program is common to all types of program faults, the different faults being distiguished on entry by a marker or counter in a B line. The program investigates whether the fault has been "trapped" by the program, and if so enters the trap; it is also possible for the program to request private monitor action, in which case the program is re-entered, either before or after the standard monitor printing. On conclusion of monitor printing, the "End Program" sequence is entered.

7.2 Types of Faults

7.2.1 Program faults

a) Faults detected by hardware

These results in the setting of a look-at-me, line 1 of the central computer V store, and include exponent overflow, division overflow, use of an unassigned function, and sacred violation. The program causing these interrupts must be in control at the time of the interruption; a common interrupt routine deals with all these faults, extinguishing the appropriate look-at-me and setting a digit in B91 corresponding to the type of fault. It is assumed that the error has been caused directly by the current object program and not by failure in supervisor routines. Multiple faults can be dealt with by setting appropriate digits in B91. One (common) SER is entered to the slow SER queue to continue the analysis.

b) Faults detected by S.E.R.

Faulty use of store and peripherals are detected by S.E.R. entered from extracode instructions in the object program. Only one such fault is detected at any one time. It is recorded in B91 as a counter without altering any fault already recorded of an interrupt type, and the same S.E.R. is entered as that initiated by interrupt faults. This S.E.R. forces the current object program controls to cause entry to routine in extracode when all currently active S.E.R's are concluded.

Faulty use of store blocks, namely reference to an illegal block number or exceeding store allocation, may be detected when the program is not in control of store but has been resumed "in supervisor" following a fault. In such a case, the monitor program causes re-entry to itself when the program is resumed in store control, when B91 is set as usual.

Over-running time is also detected by S.E.R's. Over-running of computing time (exceeding either a local counter or the overall counter set in the job description)is detected when page timers are updated, and can **only** occur when the program is in control of store. Suitable digits are set in B91, as if this fault has been detected by hardware. Over-running tape waiting time is recorded in a similar way.

Other faults may be detected when the program is halted for some reason and is not in control of store. One such fault is when the drum routines obey the extracode "Read to page P"; if page P is locked down when the instruction is obeyed at the top of the drum queue, the transfer cannot proceed. The block involved is unlocked, without completion of the transfer, and the usual S.E.R. monitor sequence is entered if the program involved is in control of store. If it is not in control of store, the monitor sequence is not entered until the program is resumed. It is assumed that, when using this extracode, the programmer has made due allowance for this occurrence and is prepared to be interrupted at any succeeding time.

"Off line" faults may be detected in the use of on-line peripheral equipment e.g. the program may reach the end of a magnetic tape. This fault is not detected until the transfer is actually obeyed, which may be some time after the instruction was given by the program. To enable the programmer to deal with such a fault and resume the program, the contents of extracode working registers are specially preserved before the monitor sequence is entered, and only one such monitor reason is dealt with at once. The method used by the programmer to resume after such a fault is described below.

c) Faults detected by Extracode

Faults such as those in arguments of functions are detected directly by extracodes. Only one such fault is detected at once, and the extracodes set a suitable counter in B91 and jump directly to the monitor sequence. In the ways described above, all programs errors enter a common extracode sequence with B91 holding a record of all faults detected "in parallel".

7.2.2 Computer Faults

After a computer fault has been detected and dealt with, and it is desired to restart on object program, the monitor sequence is again entered to cause interruption to the present flow of control and to delete or restart the program. Again the monitor sequence is entered in extracode control with suitable fault records in B91. A similar course is followed after a failure in magnetic tapes which may cause the program to be abandoned. As in the case of program failures in use of tapes, only one such fault is monitored at any one time. In all cases of computer failure, current extracode working registers are preserved before entry to the monitor sequence.

7.3 Trapping of Faults

The programmer had facility to "trap" individual errors and so cause immediate exit from the monitor sequence. The programmer provides a trapping vector, and informs the supervisor of its location by means of extracode instructions 1110. Full word n of this table contains trap information for an error of type n ($0 \leq n \leq 15$). Half word n holds the address to jump to main control; the previous value of main control is stored in the B register specified in digit 8-2 of half word 0.4 n of the table. When the trap is entered B91 hold the fault information as described in Section 7.2, B92, 93,121 are altered but all other registers are unaltered. Not all errors can be trapped; only those are included which the programmer might reasonably be expected to deal with before resuming the program, and the occurrence of which may be a useful means of avoiding extra checking in the program. For example, overflow of a local timer may prove a convenient end of an iterative loop. Faults which the programmer might be expected to avoid (such as sacred violation) cannot be trapped; faults arising from violation of the original job description, such as overall time exceeded, cannot be trapped.

Trapping may be avoided by specifying a negative address in extracode 1110; unless specified otherwise, the supervisor assumes no trapping. In order to trap some errors but not others, the programmer may fill any unwanted entry n, in the trapping table with a negative jump address in half word 0.0n. Trapping of program errors is treated by the supervisor as a "normal" procedure and entry to a trap permits the program to continue normally. If multiple errors are detected (B91 contains a record of more than one error on entry to the extracode monitor sequence), trapping is ignored if a fault is not in the group for which trapping is allowed. If all faults can be trapped, the highest priority fault is inspected and exit is made to the trap or to continue monitor, according to the setting of the relevant entry in the trapping table. It is the programmers responsibility to deal with multiple faults of which only one is trapped.

Computer failure can be trapped; the monitor routine arranges to queue these up, allowing one to be trapped at once, and an extracode "Exit from trap" must be used after entry to such a routine. Similarly when off line program error (e.g. tapes) are trapped. If it is not used, no further information will be given on computer or tape faults. This extracode can specify the following actions, according to the address S+bm, as follows

S+bm	Action required
-ve	Monitor, printing "Monitor entered"
0	Restart
+ve, odd	Resume at S
+ve, even	Recover working registers and resume at current extracode, after setting B127 to S.

If the program is not in trap, all cause exit to monitor; similarly if the program is in monitor. Restart is only permissible after a computer failure, not after an off line program fault. If any other similar faults are awaiting attention, they are dealt with before resumption.

7.4 Standard and Private Monitor Printing

7.4.1 Entry to private monitor

If a fault is not trapped, the monitor program regards the program as effectively terminated and proceeds to diagnostic printing. This consists of

a) one line describing the fault.

b) standard post mortem printing.

The programmer may supply a private monitor sequence, using extracode 1112 to specify the starting address, and this is entered in main control before (a), after (a), or after (b), according to digits 1,0 of the starting address

Digits 1,0 =	0	1	Enter before (a)	
	0	0	Enter after (a)	(the normal case)
	1	0	Enter after (b)	

When entered before (a), B91 contains the record of faults, B92 contains the current value of main control, and B93, 121 only are altered. In certain cases such as page lock down, where a page number is to be specified, this is found in B119 on entry. When entered after (a) or (b), the contents of B96, 97 are also altered.

Once private monitor has been entered, it will never be re-entered for any subsequent fault; any subsequent fault may be trapped, but if not trapped will cause standard monitor printing. This is necessary in order to avoid endless loops of errors in the event of faults in the private monitor sequence itself. Examples of the application of this rule are when overall computing time, execution time, or output time are exceeded; the monitor routines add standard amounts to the check values to allow for monitor action, and if these exceeded by a private monitor routine, they are incremented again, but cause entry to standard monitor printing.

7.4.2 Standard Monitor

The standard monitor printing routines are in main store, and are called from the drum and copied to form a part of the object program when required; for this and other purpose, one spare block is always retained with each program. The routines operate under main control, but the "process switch" is set before entry, permitting exit to extracode when required.

A description of each fault is printed on a separate line, using output stream 0 of the program. The messages are stored as packed characters and are of variable length; only characters common to all output equipments are used. Before any such printing, program branching is terminated if it was in use.

If no private monitor printing has been requested, a standard post mortem is printed. This consists of the following information:

Line 1: Heading ORDER followed by value of main control less 1. If this value is in private store, the description UNALLOCATED is printed. If the store location has not been defined, this description is also printed. Otherwise, the contents of the specified store are printed as

Function, Ba, Bm, S

The function is printed in octal form; Ba, Bm as decimal integers to three figures; the "full word" part of S as a decimal integer, signed, followed by a "point" and the last octal digit, unless this is zero, when it is omitted. Following this, the contents of Ba, Bm are printed as signed decimal integers followed by the last octal digit; this print is omitted if the B line is B0.

Line 2: Repeat of line 1 for main control
These two lines described the instructions most likely to have caused faults. Extracode faults are caused by order M-1; block addressing faults may have been caused by order M, resulting in non-equivalence.

Line 3, 4, 5.....
Value of B lines 1 to n (n a preset parameter, 10 suggested upper limit). These are printed 4 to a line in the form

B 3 = signed decimal (. octal digit)

the octal digit being omitted if zero.

Line 6: Heading "ACCUMULATOR" followed by single length accumulator, unstandardised, as a signed fraction. This followed by "/" and the octal exponent. If exponent overflow has occured, the overflow digit is ignored, but the sign digit is preserved for printing.

Line 7,....:

If magnetic tapes are in use, their positions are listed on separate lines as

TAPE n AT (block number)/(word number)

the word number being omitted if variable tape operations are not in use. This concludes the standard monitor printing; private monitor printing is then entered if called for, otherwise the printing is augmented by the "End Program" sequence. This prints the number of instructions obeyed, and the accumulated time of use of magnetic tapes. The quantity of output on each stream is printed at the end of each stream; the quantity and location of input is printed at the start of the program, again on output stream 0.

7.5 Table of Faults and Numbering

Fault	Detected by*	Mark of count in B91	Trap number if any
Local time	S	Dig. 18	0
DO	I	Dig. 17	1
EO	I	Dig. 9	2
Page locked down	S	Dig. 22	3
No. of blocks	S	2.0	4
Square root	E	2.4	5
Log	E	3.0	6
Trig function	E	3.4	7
Inverse function	E	4.0	8
Input ended	S	4.4	9
End of tape	S	5.0	10
Variable string error	E	5.4	11
Unassigned function	I	Dig. 19	
SVI	I	Dig. 15	
SVO	I	Dig. 13	
Illegal block number	S	9.6	
Band not reserved	S	10.2	
Computing time	S	Dig. 21	
Execution time	S	Dig. 20	
Input not defined	S	11.6	
Output not defined	S	12.2	
Output exceeded	S	12.6	
Tape not defined	S	13.2	
Illegal search	S	13.6	
No selected tape	S	14.2	
No mode defined	S	14.6	
Mark in read mode	E	15.2	
Tape failures	S	6.0	12
Computer failures	S	6.4	13

* I = Interrupt
E = Extracode
S = S.E.R.

Note: The above ordering is provisional and subject to amendment if more faults are to be trapped.

7.6 Summary of entrances to the Monitor Program

1)	Interrupt faults:	Enter R700 at (1) with B123 = line 1, C.C.V. Store. Exit to Enter Supervisor or to scan interrupts.
2)	S.E.R. faults:	Enter R700 at (2) with B100 holding fault record, B102 holding the return address after recording the fault, and B126 odd
	or	If return address is to exit from Supervisor. Enter R700 at (14) with B100 holding the fault record. B126 odd
		Note that these entrances can only be made when the program causing the fault is known to be in control of store.
3)	Extracode faults:	Enter R701 at (2) in extracode control with B91 holding the fault record.
4)	"Off-line faults"	Detected by SER: Enter R709 at (2) with B100 holding the fault record, together with digit 0 = 1, B106 holding the program number and B110 the return address after recording the fault. This entry is used for both program and computer faults.
5)	On line computer faults (e.g. parity)	As entry (4). Both these entries cause R700 to be entered when the program is ultimately resumed "out of supervisor" and due care must be taken to ensure that the program will be thus resumed and is not "permanently" halted.

SECTION 8. The Engineers Tests

8.1 The Engineers Initial Tests and Octal Input

Most of the engineers test routines will be stored on isolated sectors of the magnetic drum. Further versions of them will also exist on magnetic tape and paper tape, the latter mainly for the initial commissioning of a machine. Additional tests will be in the Fixed Store to ensure that if they are being obeyed correctly sufficient of the machine is working to read more tests either from the magnetic drum or from paper tape. For the latter purpose an Octal Input routine is included in the Fixed Store. This routine occupies 64 registers, uses only the flip-flop B-registers as working space and requires ten B and five Test instructions.

Entry to the Initial Tests is by pressing the Engineers Interrupt button on the Engineers console. The effect of this is:

a) to set a digit in the Central Computer V-store (digit 27, line 5) and to switch the Engineers tape reader on

b) to put the address of the first instruction in the Initial Tests (2560 in the Fixed Store, Octal 40050000) in Interrupt Control

c) to set the I/ME digit to I. No record is kept of the previous state of this digit and hence it is not normally possible to resume any programs that were being carried out at the time the botton was pressed.

The Initial Tests use the display lights on the console (B120) as an output device. They also read the handswitches on the console and hence these can be used to indicate whether further tests are to be carried out or control returned to the Supervisor.

In addition to the 512 words allocated for the Initial Tests and Octal Input a further 256 words are available for a Drum Transfer test.

SECTION 9. Detailed Specifications of Arithmetic Extracodes 1200 - 1777

9.1 Test Instructions 1200 - 1237, 1712, 1736 and 1737

In this section more detailed specifications are given of the arithmetic extracode functions.

In the table the extracode function is given on the left, followed by a description. On the right, four columns give the following information, in order:-

a) The total number of orders obeyed. This includes the extracode order itself. In some cases a range or simple formula is given.

b) The number of registers used in the fixed store. This column is subdivided to show those in the jump table and those in the main area of the store.

c) The B-registers used. This list does not include B-registers 119, 121, 122, 126 as these are always used.

d) Interconnected routines. In many cases, routines are so interleaved that an arbitrary decision has to be made as to which registers belong to which extracode.

Function	Description	Orders Obeyed	Registers J.T.	Registers Main	B-registers	Connected Routines
1200	ba' = n if AO set, clear AO. If the accumulator overflow is set, place integer n in Ba. Clear the overflow setting.	9	1	7	91	1201
1201	ba' = n if AO not set, clear AO If the accumulator overflow is not set, place n in Ba. Clear the overflow setting.	7	1	-	91	1200
1202	ba' = n if m $\neq$ 0 or -nl If the more significant half of the accumulator is neither zero nor all ones, place n in Ba.	11	1	9	91,124	
1206	ba' = n if m.s. char. in g=0. If the most significant six bit character in the logical accumulator is zero, place n in Ba.	4	3	0	91	
1216	ba' = n if bm > 0 If the contents of Bm are greater than zero, place n in Ba.	4-6	1	4		1217
1217	ba' = n if bm $\leq$ 0 If the contents of Bm are less than or equal to zero, place n in Ba.	3-5	1	4		1216

Function	Description	Orders Obeyed	Registers J.T.	Registers Main	B-registers	Connected Routines
1223	ba' = n if Bc = 1 If B-carry is set, place n in Ba.	4	3	0	91	
1226	ba' = n if bt > 0 If the B-test register contents are greater than zero, place n in Ba.	4-6	1	0		1227
1227	ba' = n if bt ≤ 0 If the B-test register contents are less than or equal to zero, place n in Ba.	3-5	5	0		1226
1234	c'=c+2 if am approx. equal to s. am' = am, l'=0 If the contents of the most significant half of the accumulator are approximately equal to the contents of S, and am is non zero and standardised, main control is stepped on by two. Approximate equality is defined by $\left\lvert \frac{am-s}{am} \right\rvert < C(ba)$, i.e. the modulus of (am-s divided by am) is compared with a number whose address is held in Ba. If am is zero, the test is ignored. If am is non standard, an interrupt occurs on the division. am is preserved but l is lost.	11	1	9		1235
1235	c'=c+2 if am not approx. equal to s. am' = am, l' = 0 Main control is increased by two if am is not approximately equal to s i.e. $\left\lvert \frac{am-s}{am} \right\rvert \geq C(ba)$ Other details as 1234, except that if am = 0, then am is not approx. = s.	12	1	1		1234
1236	ba' = n if am > 0 If the contents of Am are greater than zero, n is placed in Ba.	4-6	1	0		1237
1237	ba' = n if am ≤ 0 If the contents of Am are less than or equal to zero, n is placed in Ba.	3-5	5	0		1236

Function	Description	Orders Obeyed	Registers J.T.	Registers Main	B-registers	Connected Routines
1712	c+1) ( ) c c+2)as am(=)s ; am'=am, l'=0 c+3) ( ) Main control is stepped on by one, two, or three, depending on am being greater, equal to or less than the contents of A. The contents of Am are preserved, but the contents of L are lost.	7	1	5		
1736	c'=c+2 if am s If the modulus of am is greater than or equal to s, then main control is increased by two. The contents of Am are preserved, the contents of L are lost.	9	1	6		1737
1737	c'=c+2 if am s am'=am, l'=0 If the modulus of am is less than s, main control is increased by two. am is preserved, l is not.	8	1	1		1736

Function	Description	Orders Obeyed	Registers J.T / Main	B-Registers	Connected Routines
1727	c' = c +1, 2 or 3 as am >, = or < s. am' = am, al = 0 Main control is increased by one, two or three, depending on am being greater, equal to, or less than s. am is preserved, al is cleared.	7	1 / 5	-	-
1736	c' = c + 2 if $\|am\| \geqslant s$ am' = am, al' = 0 Main control is increased by two if the modulus of am is greater than or equal to s. am is preserved, al is cleared.	9	1 / 6	-	1737
1737	c' = c + 2 if $\|am\| < s$ am' = am, al' = 0 Main control is increased by two if the modulus of am is less than s. am is preserved, al is cleared.	8	1 / 1	-	1736

9.2 Character data processing

In 1250 and 1251 S is taken as a character address

Function	Description	Orders obeyed	Registers J.T / Main	B-registers	Connected Routine
1250	ba' = s The 6-bit character of address s is placed in digits 0-5 of Ba; the rest of Ba is cleared	7 - 10	1 / 8	91	-
1251	s' = ba The least significant six bits in Ba are placed in character address s; the rest of s is unchanged.	11 - 18	1 / 12	91,92,93	-
1252	Unpack n packed characters Unpack n characters, packed from character address C(ba) and place in digits 0-5 of successive half words starting from C (ba*), clearing digits 6-23 of these words. ba and ba* are unchanged.	<16+7n	1 / 25	91-95	1253
1253	Pack n characters Take n characters, stored in digits 0-5 of successive half words starting at C(ba*), and pack them into locations starting at character address C(ba). ba and ba* are unchanged	26+5n	1 / 17	91-95	1252

9.3 B-register Operations

Where relevant, and unless otherwise stated, the point is one octal place up from the least significant end.

Function	Description	Orders Obeyed	Registers J.T / Main	B-registers	Connected Routines
1300	ba' = int. pt. s, am' = frac. pt. s. The integral part of s is placed in Ba; the fractional part of s is placed in Am, standardised.	10	1 / 7		1301
1301	ba' = int. pt. am, am' = frac. pt. The integral part of am is placed in Ba; the fractional part is left in Am, standardised.	9	1 / 0		1300
1302	ba' = ba.n ba is multiplied by n and the result placed in Ba. The octal fraction, if any, is rounded towards zero to the nearest eighth.	23-24	1 / 24	95,97	1302-4 1312-4
1303	ba' = -ba.n ba is negatively multiplied by n and the result placed in Ba, rounded as in 1302	22-23	1 / -	95,97	"
1304	ba' = int. pt. (ba ÷ n), 697' = remainder. The integer result of dividing ba by n is placed in Ba, and the remainder in B97. The remainder has the sign of the dividend, so the quotient is rounded towards zero.	25-28	1 / 9	95,97	"
1312	ba' = ba.n for 24 bit integers ba is multiplied by n and the result placed in Ba.	23-24	1 / -	95,97	"
1313	ba' = -ba.n for 24 bit integers ba is negatively multiplied by n and the result placed in Ba.	22-23	1 / -	95,97	"

Function	Description	Orders Obeyed	Registers J.T / Main	B-registers	Connected Routines
1314	ba' = ba ÷ n for 24 bits integers; b97' = remainder The integer result of dividing ba by n is placed at the foot of ba and the remainder in B97. The remainder has the sign of the dividend, so the quotient is rounded towards zero.	25-28	1 / -	95,97	1302-4 1312-4

In the six shift extra codes following the octal fraction of n is ignored.

Function	Description	Orders Obeyed	Registers J.T / Main	B-registers	Connected Routines
1340	ba' = ba.2^{-n} arithmetically	11-23	1 / 38	91,92	1340-5
1341	ba' = ba.2^{n} arithmetically ba is shifted right or left n places. If n is negative ba is shifted n places in the opposite direction. Right shifts are unrounded and negative sign digits are propagated.	10-21	1 / -	91,92	"
1342	ba' = ba circularly shifted right n places	10-20	1 / 9	91,92	"
1343	ba' = ba circularly shifted left n places If $\|n\|$ is not less than 24, the correct answer is still obtained but extra orders will be obeyed to reduce n until it is within range. If n is negative ba is shifted in the opposite direction.	9-19	1 / -	91,92	"
1344	ba' = ba logically shifted right n places	12-23	1 / 13	91,92	"
1345	ba' = ba logically shifted left n places ba is regarded as a positive number so a negative sign digit is not propagated on right shifts. No rounding takes place. If n is negative, ba is shifted in the opposite direction. If $\|n\| \geqslant 24$, ba' = 0.	11-22	1 / -	91,92	"

Function	Description	Orders Obeyed	Registers J.T / Main	B-registers	Connected Routines
1347	s' = s v ba The result of OR ing s with ba is placed in S.	5	4 / 0	91	-
1356	bt' = ba ≢ s The B-test register is set by the result of non-equivalenting ba with s	5	1 / 0	-	1357
1357	bt' = ba ≢ n The B-test register is set by the result of non-equivalenting ba with n	4	3 / 0	-	-
1364	ba' = (ba & $\bar{n}$) (bm & n) b119' = (ba ≢ bm) & n The digits of bm are copied into ba where there are ones in n and the digits of ba are unchanged where there are zeros in n.	6	5 / 0	-	-
1376	bt' = ba & s The B-test register is set by the result of collating ba with s	5	1 / 0	-	-
1377	bt' = ba & n The B-test register is set by the result of collating ba with n	4	3 / 0	-	1377

SECTION 10. Detailed Specifications of the Magnetic Tape, Drum and Peripheral Extracode Routines

10.1 The Magnetic Tape Extracodes 1001 - 1047

Block Transfers

In the following instructions the parameter K ($0 \leq K \leq 7$) is used as a counter, but if K = 0 the count is set as 1.

1001 Search for section S on tape B and stop just before it

1002 Read the next K sections from tape B into store blocks P, P+1, ..., P+K-1

1003 Read the previous K sections from tape B into store blocks P+K-1, ..., P+1, P.

1004 Write store blocks P, P+1, ..., P+K-1 on to the next K sections on Tape B.

1005 Move tape B forwards K sections.

1006 Move tape B backwards K sections.

Organisational Instructions

1010 Mount

Allocate the number B to the tape whose title is stored in locations S, S+1, etc. If this tape is not already available instruct the operator to mount it on channel K of this program.

1011 Mount Free

Select a free tape on channel K of this program; if such a tape is not already available instruct the operator to mount one. Write on Section 0 of this tape the title stored in locations S, S+1, etc. and allocate it as tape B of this program.

(If K = 0 in instructions 1010 and 1011 it will be assumed that any channel may be used. Instructions to mount tapes may also be given on a steering tape or card).

1012 Mount the next reel of file Ba and allocate the number n to it.

The tape title has been previously identified in the job description as being a unit in file Ba. Allocate the program number n to it and if this tape is not available instruct the operator to mount it. The tape may be renamed, by extracode 1022, to have the label Ba when processing of the tape currently referred to as Ba is completed.

When forming a new file, a new FREE tape is loaded which the programmer may title by use of extracode 1014.

1013 Receive Tape (from another program).

1014 Write Title

Write on section 0 of tape B the title stored in S, S+1, etc. Also search for section 1.

1015 Read Title

Read the title of tape B from section 0 to locations S, S+1, etc. Also search for section 1.

1016 Unload (Preserve for later use)

Rewind tape B, disengage the tape mechanism. Instruct the operator to remove the tape, ensure that the correct title is written on the reel and store it for later use.

1017 Free Tape (Not required again)

Erase the title on tape B and return the tape to the Supervisor program for general use. (Tapes may also be freed by means of a steering tape).

1020 Release Tape (Pass it to another program)

Delete tape B from the allocation of this program and make it available for another program without freeing or disengaging it.

If $n \neq 0$ in instructions 14 to 16 above, the number of tape mechanisms reserved for the program is reduced by one.

1021 Release Mechanisms

Reduce by S the number of tape mechanisms reserved for use by the program.

1022 Re-allocate

Allocate the number S to the tape which was previously referred to in this program as b_W.

1023 How long?

h'_W = number of 512 word sections available on tape B (excluding section 0). In full word address position of specified half word.

1024 Where am I? (See also under Variable Length Instructions)

After block transfer orders

$s' = A/\text{Zero}$

where A = Address of next section on tape B, going forwards. In full word address position of first half word.

Variable Length Instructions

Start Instructions

1030 Start Reading Forwards

Select tape B to be read forwards using variable length transfers, starting at the next word on the tape. Henceforth ensure that at least 512K words are held in the buffer awaiting transfer. The buffer is in blocks P, P+1,...., P+K+1.

1031 Start Reading Backwards

Select tape B to be read backwards using variable length transfers, starting at the previous word on the tape. Henceforth ensure that at least 512K words are held in the buffer awaiting transfer. The buffer is in blocksP, P+1, .. P+K+1 (or P+K if Q = 0).

1032 Start Writing Forwards

Select tape B to be written forwards, using variable length transfers, starting at the next word on the tape. Up to K+1 buffer blocks are used as required. A Marker Q is written before the first word of information ($1 \leq Q \leq 7$) The buffer is in blocks P, P+1, ..., P+K.

1033 Select

Select tape B for succeeding variable length operations, in the mode previously specified for that tape.

1034 Start Reading Forwards from Fixed Blocks

1035 Start Reading Backwards from Fixed Blocks

Instructions 1034 and 1035 are the same as 1030 and 1031 except that they initiate variable length reading for tapes which have been written by block transfers, and are therefore not divided into strings.

1036 ba' = Selected Magnetic Tape

Set Ba to contain the program number of the magnetic tape currently selected for variable length operations. This extracode may be used if it is desired to select a different magnetic tape for variable length transfer (e.g. in a sub-routine) and then re-select the original one. If no tape is currently selected, ba is set to a negative value.

1037 s' = mode of magnetic tape Ba

Put n in the store line S to indicate the present mode of use of tape Ba where

n = 0 for variable length read forwards transfers using strings
n = 1 for variable length read backwards transfers using strings
n = 2 for variable length write transfers using strings
n = 3 for fixed block transfers
n = 4 for variable length read forwards transfers from fixed blocks
n = 5 for variables length read backwards transfers from fixed blocks

Transfer Instructions

1040 Transfer

Transfer b_w words between store addresses starting at S and the selected tape, in the mode (reading forwards, reading backwards or writing) appropriate to that tape. The transfer is terminated on a marker b_k.

On Writing, b_w words from locations S, S+1, ... S + b_w - 1 are written to the next b_w locations on the selected tape. A marker b_k is written on tape after them.

On Reading, the transfer continues until b_w words of information have been read or until a marker $\geqslant b_k$ is encountered, whichever is the sooner

b_w' = The number of words of information actually read.

b_k' = 0 if no marker $\geqslant b_k$ was encountered.

= m if a marker m ($\geqslant b_k$) terminated the transfer or came immediately after word b_w.

When reading forwards the next b_w' words are read from tape to store locations S, s + 1, ... S + b_w' - 1.

When reading backwards the previous b_w' words are read from tape to store locations

$$S + b_w - 1, \quad S + b_w - 2, \; ..... \; S + b_w - b_w'$$

If b = 0 when reading, the transfer continues until the first marker is encountered, as though b_w were equal to b_w'. When reading backwards this means that b_w words are read to store locations

$$S + b_w' - 1, \quad S + b_w' - 2, \; ..... \; S.$$

1041 Skip

Skip b_w words, terminating on a marker b_k.

Skip operates in the same way as transfer except that no words are transferred.

When in a writing mode b_w addresses on tape are skipped and a marker b_k is written after them. Note, however, that the previous contents of these addresses, whether information or marker, are <u>not</u> preserved on tape, except when complete 512 word tape section are skipped.

When in a reading mode the skip continues until b_w words of information have been passed or until a marker b_k is encountered, whichever is the sooner.

b_w' = The number of words of information actually skipped.

b_k' = 0 if no marker $\geqslant b_k$ was encountered.

= m if a marker m ($\geqslant b_k$) terminated the transfer or came immediately after word b_w.

Note that skip is much less efficient than search for positioning the tape, and should not be used for skipping more than a few sections along the tape.

1042 Mark

Available only when in writing mode.

Writes a marker Q after the last word on the selected tape. This marker replaces any marker which was previously on the tape at this point ($1 \leqslant Q \leqslant 7$)

After writing a string on tape it may be discovered that the end of a group has been reached. The mark instruction may then be used to change the marker at the end of the string. It may be used again if it is later found that the end of an even higher order group has been reached.

1043 Stop

Stop variable length operations on tape B.

After variable length operations for a given tape have been completed a stop instruction may be given. It will release the buffer blocks associated with those operations. After writing operations it will cause the last part section to be written immediately from the buffer to magnetic tape, but this will also be done by any of the operations:

start, search, unload, release tape, block transfer.

1024 Where am I? (See also under Organisational Instructions)

After variable length transfers

$s' = A/W$

where W = Address within the section of the current marker on tape B or, if not on a marker, of the next word going forwards.

A = Address of the section containing word W.

1044 Word Search

Search for word W, section A of tape B, where $s = A/W$.

1046 Read the next block on the Orion tape Ba into store blocks P, P+1, ... P+K.

A check is made that tape B is an Orion tape (first digit in block is a zero, - read when the tape was first mounted) and the next block, reading forwards is read into store blocks P, P+1, ... P+K. A non-equivalence interrupt occurs and the program is monitored if insufficient pages are reserved for the transfer. The maximum length of transfer permitted is 4096 words but no automatic indication is given of how many words are read from the tape.

1047 Read the previous block on Orion tape Ba to store blocks P+K, P+K-1, ...P.

This is similar to 1046 except that the first word read from the tape is stored in address 511 of block P+K, the second word is stored in address 510 of block P+K, etc.

INPUT-OUTPUT EXTRACODES 1050-1067

1. End of line character.

The supervisor stores input and output behind the scenes as six-bit characters in records (which correspond to lines of printing). The last six bit character in every complete record is the end of line character. It is interpreted according to the table given below.

Input from paper tape or cards can give rise to only four of the possible end of line characters.

These are:

CR	on	5 hole tape	(octal code 20)
LF	on	5 hole tape	(octal code 01)
NL	on	7 hole tape, or End of Card	(octal code 21)
Paper throw	on	7 hole tape.	(octal code 40)

The output peripherals have, between them, the same facilities and the Anelex line printer also has Paper Throwing with homing on channels 0 - 7.

Owing to the way the different peripherals operate it is not always possible to obey correctly the end of line instructions intended for another peripheral. In these cases a compromise is made according to the following rules:

a) Line feeding. The number of line feeds ($0 \leq n \leq 15$) is always performed correctly.

b) Carriage return is performed (if requested) only if it can be done while still retaining the correct number of line feeds.
Carriage return is performed (even though not requested) if this is necessary in order to achieve the correct number of line feeds.

c) Paper throwing. The channel number is interpreted modulo m, where m is the number of homing channels available on the printer. If no paper throwing facility exists on the printer, the peripheral routine initiates one line feed instead.

d) Repeated spaces or back spaces are not inserted by the supervisor in any attempt to position the carriage correctly.

End of line characters.

Code (octal)	Effect
00 to 17	n line feeds without carriage return $0 \leq n \leq 15$
20 to 37	n line feeds with carriage return $0 \leq n \leq 15$
40 to 47	Paper throw without carriage return. Home on channels 0 to 7.
50 to 57	Paper throw with carriage return. Home on channels 0 to 7.
60 to 77	Spare

The octal code 00 (zero line feeds without carriage return) has no effect on the printer carriage, and is used to end binary records. Codes 60 to 77 also have no effect on the printer carriage.

Input Extracodes

1050 Select Input n.

All succeding 'Read' orders (until a new input is selected) refer to the data called Input n in the Job description.

If Input n was not specified in the Job description, use of this extracode causes an exit to the monitor routine.

If 'Read' orders are used without previously selecting an Input, then Input 0 is assumed.

1051 Find selected input

ba' = number of currently selected input.

1052 Find peripheral equipment number.

ba' = V-store address of the peripheral equipment used for the currently selected input.
= 0 if this input originated as output from another programme.

1053 Test Binary / Internal code.

If the next character to be read from the currently selected input stream is a binary character, sets ba' = n.

If the next character is in internal code, ba is unaltered.

If there are no characters remaining on the currently selected input stream an exit is made to the monitor routine.

1054 Read next character to Ba / Jump to n at end of Record.

Reads the next 6-bit character from the currently selected input, and places it at the least significant end of ba. With internal code input this will transfer one internal code character. With binary input, where the information is stored in 12-bit quarter words, the first use of the extracode will read the 6 m.s. bits of the binary character. The next use of the extracode will read the 6 l.s. bits.

Normally c' = c + 1 , but if the end of a record has just been exceeded, c' = n, and Ba contains the carriage control character in bits 5 - 0.

If all the characters on the currently selected input stream have already been read, this extracode causes an exit to the monitor routine.

1056 Read ba characters to S

Reads the next ba characters from the currently selected input and places them in store locations beginning at S. (Bits 0, 1 of S are ignored). The information is packed, four 6-bit characters per half word. Bit 23 of ba is ignored.

If the end of the record is not reached, ba is unaltered on exit except bit 23 which is set = 1.

If the end of record is reached, ba is set:
bit 23 = 0
bits 22 - 0 = number of characters actually read.
The last character is the carriage control character.

If all the characters in the currently selected input stream have already been read, this extracode causes an exit to the monitor routine.

1057 Read next record to S

Reads the next complete record from the currently selected input and places it in store locations beginning at S. Bits 0, 1 of S are ignored. The record is packed, four 6-bit characters per half word, and the last character is the carriage control character.
On exit Ba contains:

bit 23 = 0.
bits 22 - 0 = character count

If all the records on the currently selected input have already been read, then use of this extracode causes an exit to the monitor routine.

Note that extracodes 1056 and 1057 will run very much faster if no characters have previously been read from the record, or if the number of characters which have previously been read from the record is any other multiple of 4.

Output Extracodes

1060 Select output n.

All succeeding 'write' orders (until a new output is selected) are to the peripheral called Output n in the Job description. Bit 0 = 1 if binary characters. Bit 0 = 0 if writing internal code characters.

If output n was not specified in the Job description use of this extracode causes an exit to be made to the monitor routine.

If 'write' orders are used without previously selecting an output, then output 0 is assumed.

If a change is made from binary to internal code or vice versa, in the middle of a record, the previous part-record is terminated with a zero control character.

1061 Find selected output

ba' = number of currently selected output (bit 0 as in 1060).

1062 Find peripheral equipment type.

ba' = V-store address of equipment number 0 of the peripheral type currently selected for output
= 0 if the currently selected output is to any peripheral.

1064 Write character n.

Writes the character occupying the 6 least significant address bits to the currently selected output. If the internal code mode has been selected this will write one internal code character. If binary, the extracode must be used twice to write the m.s. and l.s. halves of each 12-bit binary character.

1065 End this record.

Writes the carriage control character occupying the 6 least significant address bits to the currently selected output, and terminates the record.

1066 Write ba characters from S

Writes the ba characters beginning at the address S to the currently selected output (bits 0, 1 of S ignored). The characters must be packed, four 6-bit characters per half word.

Ba to contain on entry:

bit 23 = 0 if the record is to be ended
= 1 if the record is not to be ended

bits 22 - 0 = character count

If the record is to be ended, the last character is taken to be the carriage control character

1067 Write a record from S.

Writes the record beginning at the half word address S to the currently selected output (bits 0, 1 of S ignored). Ba to contain character count in bits 22-0 (bit 23 ignored)

The record should be packed, four 6-bit characters per half word, and the last character is the carriage control character.

Note that extracodes 1066 and 1067 run very much faster if no characters have previously been sent to the record, or if the number of characters which have previously been sent to the record is any other multiple of 4.

SECTION 11. Detailed Specification of the Organisational Extracodes

11.1 General Organisational Extracodes 1100 - 1137

Subroutine Entry

1100 Enter subroutine at s; ba' = c + 1

1101 Enter subroutine at S; ba' = c + 1

1102 Enter subroutine at bm; ba' = c + 1

Branch Instructions

1104 Start branch B at S.

1105 Kill branch B. If B = 64 kill current branch

1106 Halt current branch if B is active

1107 Assign dump for B branches at S.

Trap Setting

1110 Set trap / normal mode

1111 Trap

Monitor Routines

1112 Set Monitor jump to n.

This instruction implies the programmer has a private monitor routine which he wishes to be obeyed if any machine or program error is detected. This routine is obeyed on Main control starting at the instruction in address (n+bm).

1113 Set Restart address to n.

This instruction gives the programmer the facility of being able to restart his program at some intermediate point without having to return to the beginning after, for example, a machine fault. The programmer must organise the storing of any information necessary for a restart before specifying this order, and his program starting at address (n+bm), obeyed on Main control, must contain the necessary instructions to return to his restart conditions. The effect of this extracode is to replace the initial entry address by the restart entry address.

1114 Dump on tape B on "End Program".

This extracode specifies that the programmer requires all the information connected with his program to be stored on the specified tape when instruction 1177 is obeyed. Recording starts at the next block on tape B.

1115 Dump on tape B if program monitored.

If the program is monitored because of a program or machine fault being detected, the program is dumped on the specified tape and not on the standard dump tape.

1116 Do not dump if program monitored.

If the program is monitored because of a program fault being detected, the program is not dumped.

1117 End Program.

Inform the supervisor that the program is ended.

Miscellaneous transfers

1120 ba' = clock

1121 ba' = date

1122 ba' = instruction counter

1123 set instruction counter = $n.2^{10}$

1124 v6' = n

1125 ba' = v6

1127 ba' = v6 & n.

Searches

1130 Iterative substitution: if m.s. but of S is 0 ba' = s, exit; otherwise test $C(S - 2^{20})$, etc.

1131 Table search

11.2 Compiler and Supervisor Extracodes 1140 - 1157

1140 Read parameter ba to s.

Various parameters are connected with each program e.g. expected computing time, the number of drums required etc., and these parameters are listed by the Supervisor for each program. This extracode transfers the parameter identified by ba to the store location specified by the S digits.

1141 Write parameter ba from s.

Copy the contents of store location S to the parameter list location specified by the number in Ba. This extracode is used by compilers to amend job descriptions where necessary.

1142 Interpret the directive at S and return control to ba.

This is an extracode used by the compiler routines. It provides a general means of returning control to the supervisor following a stage of compilation. A section of "Job Description" stored in internal characters in location S onwards is decoded by the supervisor. If it is description (e.g. specification of time) control is returned to the address given in Ba after recording the relevant parameter. If it is also imperative, suitable action is taken. As an example, the "Job Description" ***Z or DATA cause an exit from compilation to execution of the target program according to preset parameters in the parameters list.

1143 Call System document s to be input stream ba.

This extracode is used by compilers to call library routines etc. recorded on a system library tape in the form of an input or output stream. The contents of store S onwards identify the document, which can subsequently be read by selecting the input stream ba.

1144 Call System document s to store blocks ba onwards.

This extracode is used by compilers to call library routines etc. recorded on a system tape in binary form (e.g. a binary program, a dump etc.). The contents of store S onwards identify the document, which is read into store from block ba onwards.

1147 Call in compiler n

The appropriate compiler is attached to the program, overwriting any blocks already in use whose block labels are used by the compiler. Control returns to the instruction following this extracode.

1150 Assign ba blocks, labels n to n +ba-1, to overflow.

This extracode enables a program or compiler to temporarily hand blocks n to n +ba-1 to the supervisor, which may write them to the system dump tape. Subsequent use of these labels in the program causes new blocks to be assigned. The block labels are retained in the "overflow" region and additions to this region must bear distinct block labels. If ba = 0, one block is transferred.

1151 Set up ba blocks, labels n to n +ba-1, from overflow.

This extracode recalls blocks previously written to the overflow region by use of 1155. Any existing blocks having these labels are overwritten. If ba = 0, one block is recalled. If these blocks do not exist in the overflow region, the program is monitored.

1156 Enter Extracode control at n if the In Supervisor switch is set.

This routine is used by various Supervisor Extracode Routines which are obeyed on main control. The effect is to transfer to Extracode control in the normal way and to obey the instruction in location (S+bm). If the "In supervisor" switch is not set, this extracode is treated as an illegal instruction and the main store program which specified it is monitored.

1157 Enter Extracode control at n if the "Process Switch " is set.

Similar to 1156 except that the condition for successfully obeying this routine is that the "Process" switch for the current main program must be set.

11.3 Store Extracode 1160 - 1177

Drum Transfers

It is envisaged that most programmers will use the core and drum store as a one-level store. However, in certain circumstances it may be useful to be able to specify that a given block of information should be either in core store or that it can be written to the drum store. For this purpose the drum transfer extracodes listed below are provided. Should a programmer using drum extracodes make a "mistake" (e.g. he writes block b to the drum just before he uses it again) the one-level "automatic" transfers take over and rectify the "mistake", although there may be some time wasted waiting for drum transfers.

All drum transfers are initiated via the drum queue. For the one-level store, one page of core store is kept empty i.e. with no useful information in it. The sequence of events started by a non-equivalence interrupt is as follows:-

(a) A supervisor extracode routine (the drum transfer routine) is entered. This SER inserts the item "read block P" in the drum queue. The object program is halted.

(b) From the drum queue, the drum transfer to read the required block to an empty page is initiated by writing to the appropriate lines of the drum V-store.

(c) The learning program is entered to chose a page to write to the drum.

(d) When the read transfer is complete the drum transfer to write the contents of the chosen page to the next empty sector of the drum is initiated. (if the chosen page is already empty the next transfer in the drum queue is initiated).

(e) When the write-to-drum transfer is complete the next transfer in the drum queue is initiated.

An attempt is made to keep instructions on different pairs of stacks of the core store from operands. (This decreases the core store access time and so increases the rate of obeying instructions). The core store is divided into two sections; one section is for instructions, the other is for operands. On the Manchester University Atlas these two sections are of equal size, each 16 pages (8096 words). When there are empty pages in both sections the required Block is read to the appropriate section; if the non-equivalence is caused by a request for an instruction the block is read to the instructions section of the core store, and if for an operand to the operands section. When there are empty pages in only one section of the core store the block is read to an empty page even if it is the "wrong" section. When the learning program is choosing pages to write to the drum it attempts to rectify these "errors" and get instructions in the instruction section of the core store and operands in the operands section of the core store.

Store Extracodes

1160 Head block P (4 registers of fixed store)

(Address digit 23 = 0 if operands section preferred
1 if instructions section preferred

digits 22-12 = P
remaining digits irrelevant)

This is the same as an "automatic" drum transfer to read block P to the core store which is started by a non-equivalence interrupt. However the object program is not halted.

When the drum transfer is inserted in the drum queue (see (a) above) control may be returned to the object program even before the read drum transfer is initiated. The original drum copy is lost i.e. the sector originally containing block P is indicated as empty in the directories. If the block is already in core store, nothing is done.

1161 Duplicate read (72 registers)

(Address digits 22-12 = P_1
remaining digits irrelevant

digit 23 of ba = 0 if operands
1 if instructions

digits 22-12 of ba = P_2
remaining digits irrelevant)

If block P_1 is on the drum a read transfer is performed to form P_2 in the appropriate section of the core store. A page is chosen by the learning program and written to the drum so that one page of core store is kept empty. If the block P_1 is in the core store, a write-to-drum transfer is initiated to form block P_2 on the next empty sector and then the block labels P_1 and P_2 are interchanged so that the duplicate block P_2 is now in core store as required. If the block P_1 is in the core store and the drum store is full, block P_1 is copied to an empty page via the accumulator to form block P_2. If block P_1 is not allocated, block P_2 is lost and nothing further is done. In every case the original block P_2 (if any) is lost.

1162 Read (K+1) blocks (80 registers)

(Address digit 23 = 0 if operands, 1 if instructions

digits 22-12 = P
digits 2 - 0 = K
remaining digits irrelevant
digits 10-0 of ba = D (digits 10-8 = cabinet, digits 7 - 6 = drum, digits 5-3 = band, digits 2-0 = θ)
remaining digits irrelevant).

A multiple block drum transfer is performed to read K blocks from sector D onwards to form block P, (P+1)........ (P+K-1) in the core store. There are only 6 sectors round a band of the drum and all θ's including the initial θ written in the address, are interpreted modulo 6.

e.g. if the starting θ = 5, (cabinet, drum and band 0) K = 3 and P = 25

Then sector 0.5 is read to block 25
sector 0.0 is read to block 26
and sector 0.1 is read to block 27

(n.b. sectors 0.0 and 0.1 <u>not</u> sectors 1.0 and 1.1)

(K+2) separate items are inserted in the drum queue as follows:-

"lock down block P"
"lock down block P + 1"
.
.
.
"Lock down block P + K"
"multiple block read"

1163 Read to page p (40 registers)

(Address digits 22 - 12 = P, block label
remaining digits irrelevant

digits 11-3 of ba = p. page number
remaining digits irrelevant)

(a) This item is inserted in the drum queue.

(b) From the drum queue, the contents of page p are copied to an empty page via the accumulator. The drum transfer to read block P to page p is initiated.

(c) The learning program is entered to chose a page to write to the drum.

(d) The chosen page is written to the drum.

If when (b) is begun, page p is locked down, a trap is entered i.e. jump to an address specified in a previous "trap" extracode. If page p is empty the copy part of (b) above is omitted. If block P is already in the core store the contents of page p is copied to an empty page and then block P is copied to page p via the accumulator. The original sector or page occupied by block P is made empty. If block P is already in page p nothing is done.

1164 Rename (39 registers)

(Address digits 22-12 = P_1
remaining digits irrelevant

digits 22-12 of ba = P_2
remaining digits irrelevant)

Block P_1 is renamed block P_2. If the block is in the core store the page address register is changed appropriately. The original block P_2 (if any) is lost. If block P_1 is not allocated, block P_2 is lost and nothing further is done.

1165 Store allocation = n blocks

This enables the number of main store blocks allocated to a program to be changed during the execution of this program. (The store allocation is also set by the JOB tape).

1167 Clear blocks (10 registers)

(Address digits 23 = 0 if clear blocks required
= 1 if clear blocks not required

remaining digits irrelevant)

Whenever a new block of main store is referred to, an empty page of core store is allocated and is then cleared to floating point zero by a loop of accumulator instructions if the "clear blocks" switch is set. This extracode sets or resets this switch. The switch is set initially to clear all blocks.

1170 Write block P (50 registers)

(Address digits 22-12 = P
remaining digits irrelevant)

Block P is written to the next empty sector and the page of core store originally occupied by block P is made empty. If the drum store is full block P is released from core store as in the extracode "release block P". If block P is already on the drum or is not allocated, nothing is done.

1171 Duplicate write (8 registers)

(Address digit 23 = 1 if operands, 0 if instructions
digits 22-12 = P_1
remaining digits irrelevant
digits 22-12 of ba = P_2
remaining digits irrelevant)

Block P_1 is written to the next empty sector to form block P_2 on the drum. If the drum store is full, block P_1 is copied to an empty page to form block P_2. If block P_1 is on the drum, a read drum transfer is performed to form block P_2 and the labels are interchanged so that block P_2 is on the drum as required. If block P_1 is not allocated, block P_2 is lost and nothing further is done.

1172 Write (K+1) blocks (80 registers)

(Address digits 22-12 = P
digits 2-0 = K
remaining digits irrelevant

digits 10-1 of ba = D
remaining digits irrelevant)

A multiple block transfer is performed to write K blocks to sectors D onwards from blocks P, (P+1)......(P+K-1) in the core store. (K+2) separate items are inserted in the drum queue as follows:-

"lock down block P"
"lock down block P+1"
.
.
"lock down block P+K
"multiple-block write"

All Θ's are interpreted modulo 6 as for the extracode "read (K+1) blocks".

1173 Release block P

(Address digits 22-12 = P
remaining digits irrelevant)

The page timers are set so that the learning program will choose this page to write to the drum. If block P is already on the drum or is not allocated, nothing is done.

This extracode is performed without entering the drum queue so that block P may be written to the drum earlier using this extracode than using the "write to drum" extracode which waits its turn in the drum queue.

1174 ba' = number of pages available

(digits 11-3 = number of pages
remaining digits zero)

This gives an estimate of the number of pages available to this program. No guarantee is given that this number of pages will be permanently available to this program.

1175 ba' = number of blocks available

(digits 13-3 = number of blocks
remaining digits zero)

This gives an estimate of the number of main store blocks available to the program and includes its present allocation of blocks.

1176 Lose sector D (20 registers)

(address digits 10-0 = D
remaining digits irrelevant)

Sector D, which has been received in the JOB tape, is made empty and becomes available to the one-level store.

1177 Lose block P (31 registers)

(address digits 22-12 = P
remaining digits irrelevant)

The sector or page occupied by block P is made empty. If block P is already not allocated, nothing is done.

12.1 CENTRAL MACHINE FAULTS.

Machine faults which are specifically detected during operation of the Supervisor program cause monitor action. The following notes describe the monitor action taken at present by the Secondary Supervisor and Full Supervisor, and indicate the machine registers which are preserved and may be of value in diagnosing the fault.

FIXED STORE PARITIES

Loop stop on Interrupt Control. Unless the fault occurred with interrupts inhibited, the probable area of store concerned can be deduced from the M/E digit (line 4 * 6) and B127 or 126, all of which are unaltered.

CORE STORE, WORKING STORE PARITIES

Loop on Interrupt Control if Handswitch 1 is set to 1. Otherwise the PAR's are set to the physical page numbers and the Core Store and Working Store are searched for lines of incorrect parity. The addresses of any such lines are printed on the teleprinter. When this search is complete all Magnetic Tape Decks are disengaged. Unless the Handswitches are even an automatic entry is made to the Fixed Store routine normally entered by Engineer's Interrupt and '6.6'. The Supervisor will be read from Tape when the appropriate Deck is engaged.

NON-EQUIVALENCE ON INTERRUPT CONTROL

The value of interrupt control is inevitably lost. A fixed store loop of two or three seconds is obeyed to allow peripherals to stop, and

f 20000000

is printed on the teleprinter. After this the PAR's are cleared and a loop stop is entered. If the teleprinter is manually disengaged, the PAR's will be unaltered. Line 34 * 6001 is cleared.

NON-EQUIVALENCE TAPES OR DRUM

As above only

f 00000040

is printed on the teleprinter.
The address requested is preserved in line 34 * 6001. If the teleprinter is disengaged or the machine is stopped manually during the 2-3 sec. loop, the PAR's will be unaffected.

Line 34 * 6001 should read
*3670 for drum transfer
*37 A 0 or * 37A7 for tape transfer, channel A.
The P.A.R. should be set appropriately.

DRUM PARITY, DCA, DBI, DRI, DCF

The transfer is restarted until it is successfully completed, the idling loop being the most probable background activity. After seven successive failures printing is initiated. This may occur at once or if the output routine itself is on the drum, when the transfer is successfully completed. The print comprises, for example

f drum parity r bdfh

Item 3 may read PARITY, DCA, DBI, DCF
Item 4 may read r (read transfer) or w (write transfer)
Item 5 comprises pairs of octal digits

ab	=	sector	(a = 0 b = 0 to 5)
cd	=	band	(c = 0 d = 0 to 7)
ef	=	drum	(e = 0 f = 0 to 3)
gh	=	cabinet	(g = 0 h = 0)

12.1 Continued
DRUM TRANSFER INCOMPLETE AFTER 1 TO 2 SECONDS

Action as for non-equivalence on interrupt control, with print

f 0002000A

where a -> h define the drum sector requested as above

A = 2 (read transfer)
6 (write transfer)

MACHINE TESTS IN FINAL SUPERVISOR

The Final Supervisor incorporates a test of the machine which is entered at intervals of approximately 5 minutes. It last for about 3 seconds during which time all peripherals and magnetic tapes will pause.

Should any fault be detected an indication of the type of fault is printed on teleprinter 0. For example

MACHINE TESTS FAILED 00002064

The octal number, which is also displayed in B120, gives the subtests which have failed according to the list below. After printing the test stops in a hoot loop. To restart, the Supervisor should be recalled from magnetic tape.

Digit and Lamp	Test
00000001	Instruction Counter will not store correctly.
00000002	Instruction Counter interrupting at the wrong time.
00000004	Instruction Counter failing to interrupt.
00000010	B-Store Switching Test failing.
00000020	Accumulator Test 12 Subtest 1 failing.
00000040	" 12 " 2 "
00000100	" 12 " 3 "
00000200	" 12 " 4 "
00000400	" 12 " 5 "
00001000	" 12 " 6 "
00002000	" 12 " 7 "
00004000	" 16 " 1 "
00010000	" 16 " 2 "
00020000	" 16 " 3 "
00040000	" 17 " 1 "
00100000	" 17 " 2 "
00200000	" 17 " 3 "
00400000	" 17 " 4 "

Instruction Counter Test.

This checks that it is possible to write to and read from the Instruction Counter, using several patterns. A check is also made that it will count from zero to 2048 and that it interrupts at the correct time.

B-Store Test.

This checks Switching between all pairs of B-Lines from B1 - B99.

Accumulator Test.

This consists of Tests 12, 16 and 17 of the Standard Accumulator Tests adapted to run on Extracode Control. Further details of these tests may be obtained from the Accumulator Tests Description.

12.2 PERIPHERAL FAULTS.

The following headings are printed on the teleprinter after the corresponding peripheral fault has occurred. The necessary Engineer or Operator action is given below together with an indication of whether the program is ended or continued.

DISABLED followed by peripheral type.
- a) The program will have been ended.
- b) The peripheral requires Engineer attention as it is broken.

OVERDUE followed by peripheral type.
- a) The program will have been ended.
- b) The peripheral requires Engineer attention.

CARDS OUT CP or CR (card punch or card reader)
- a) Refill the input hopper with the next batch of cards.
- b) Re-engage the reader or punch.
- c) The program should procede in a normal fashion.

CHECK FAIL CR
- a) Replace rejected card at bottom of input hopper.
- b) Re-engage reader.
- c) Program should procede normally.
- d) This procedure may be repeated as many times as required. If the card will not be accepted the program must be thrown off, manually.

CHECK FAIL CR3 (Creed 3000)
- a) The program will have been ended and the punch will be disengaged.
- b) The Creed requires Engineer attention before re-use.

CHECK FAIL CP
- a) The card has been attempted 5 times without success.
- b) The punch is disengaged and requires Engineers attention before proceding.
- c) The program will have been ended.

LOW - ANELEX, TT(teletype), TP(teleprinter) or CR3
- a) The equipment will be stopped and disengaged.
- b) Reload with paper or tape as required.
- c) Re-engage equipment and program will continue.

PARITY TR5,TR7 or CR
- a) A punching error has been detected on tape or card
- b) The program will have been ended.

12.3 MAGNETIC TAPE FAULTS

E TYPE FAULTS - REPEATED 7 TIMES
E 1 DECK N
0011 0022
(leading block address fault on deck N, the present block address on tape is given in octal by the first four digits, here it is block 9, and the requested block address is given in octal by the bottom four digits, here it is block 18)
E 2 DECK N
0011 0022
(Trailing block address fault on deck N, etc)
E 3 DECK N
0011 0022
(checksum failure on deck N, etc)
E 4 DECK N
0011 0022
(not 512 word transfer on deck N, etc)
E 5 DECK N
0011 0022
(deck N failure)

F TYPE FAULTS

F 1 DECK N
(failed to align tape to stop before present block address = expected block address after 7 attempts)
F 2 DECK N
(failed to stop when expected to stop, present block address has changed since the stop order was given)
F 3 DECK N
(stop bit not set in tape command register when expected to stop)
F 4 DECK N
(direction and read bias not set correctly in tape command register, tape not started)
F 5 DECK N
(deck failure, interrupt cannot be reset immediately)
F 6 DECK N
F 7 DECK N
(write bit not reset on tape command register after write transfer)
F 8 DECK N
(read bit not set or reset accordingly before or after a read transfer)
F 9 DECK N
(failed to clear error after 7 attempts)

If the transfer is not successful after repeated attempts the program is monitored for a deck fault.

12.4 JOB DESCRIPTION FAULTS

READER N	JOB DOCUMENT FAULT (any fault in job description not covered by other print outs, N is the peripheral identifier, see section 12.8)
READER N	TITLE TOO LONG (more than 80 characters in the title - this count includes newlines and runout characters etc)
READER N	EXCESS STORE OR TAPE (the programmer has requested too much store or too many tape decks)
READER N	INCORRECT FORMAT (this is followed by the line of incorrect format)
READER N	EXCESS DOCUMENTS (the job description specifies more than 15 input or 15 output documents)
READER N	NO INPUT OR OUTPUT STATEMENT (no input or output heading)

12.5 PROGRAM RESULTS

An example of the format of a program's output with explanation is given below

00.00.01 / 20.11.63. 16.50.49.

(the digits before the slash indicate the number given to the output document by the supervisor and have no significance to the programmer. The next number is the date followed by the time when the program first produced the output)

OUTPUT 0

(the results which follow belong to the program's output stream 0, a program may have up to 16 output documents)

BEATLE SURVEY

(title of job)

(then follows the program's output belonging to this stream)

(if the program is monitored the supervisor then gives some information about the fault, this may be followed by the program's own private fault print. The layout of the supervisor monitor print is as follows

INPUT ENDED

(this is the reason why the program was monitored, a list of program faults detected by the supervisor is given in section 11.4 of the programming manual CS 348)

```
(Instr.  X-2,  F, Ba, Bm, S     Ba = ,  Bm = .
 Instr.  X-1,  F, Ba, Bm, S     Ba = ,  Bm = .
 Instr.  X     F, Ba, Bm, S     Ba = ,  Bm = .
```

where X is the address given in B127. On the same line that X is printed are given the instruction in this address and the contents of the non-zero B lines specified in the instruction. Also printed on the two previous lines is similar information for the instructions in the two preceeding addresses. If b $\geq$ 100 b=0 is printed. It should be noted that although the contents of B lines 1 to 99 are printed B91 to 93, and 96 and 97 are destroyed by the supervisor)

INSTRUCTION 990 450

(the program had obeyed 990 instruction counter interrupts when it ended, 450 of there being during compiling)

STORE 32 / 24

(the program reserved 32 blocks but only 24 had been used when the program ended)

INPUT 0 10 BLOCKS

(input stream 0 contained 10 blocks, this line is repeated for all input streams)

OUTPUT 0 ANY 5 BLOCKS

(output stream 0 contained 5 blocks to be output on ''any'' peripheral, this line is repeated for all output streams)

END OUTPUT 5 BLOCKS

(this is used to terminate all output documents, there were 5 blocks of output in this stream)

12.6 LOG

The log comes out on punch 0 and a typical example with explanation is given below

CALL SUPERVISOR 63007600

(this is printed at the start of the day when the supervisor is brought into store from magnetic tape, 63007600 is the supervisor checksum)

16.50.47 20.11.63

(the first number is the time in hours, minutes and seconds when execution of the program began and the second number is the date)

BEATLE SURVEY

(title of job)

10 B

(10 is the number of blocks of input and B is the input medium, B is TR5 0, C is TR5 1, D is TR5 2, E is TR5 3, F is TR7 and G is card reader; this is repeated for all input documents)

5 BH

(5B is the number of blocks of ouput and H is the output type:, a amount of output may also be given in records and this is indicated by AR, Types of output medium are: BB blocks of seven hole, RC records on Anelex BD blocks on five hole, RE records on cards, BF blocks on magnetic tape, RG records on any, BH blocks on any; with the ''any'' stream if the number of records is greater than 32B then the log specifies records of output. This is repeated for all output documents)

2 155 16

(2 is the number of magnetic tape decks reserved by the program,
155 is the number of blocks transferred to or from magnetic tape,
16 is the number of seconds spent waiting for tape transfers)

32 24 2 812 990 371

(32 is the number of store blocks reserved by the program
24 is the number of store blocks in use when program ends
2 is the number of the compiler as per compiler directory
812 is the number of instruction counter interrupts at end of compiling
990 is the number of instruction counter interrupts at end of program
371 is the number of drum transfers

note that if the number of instruction counter interrupts at end of compiling is zero than the program has monitored during compiling and then IC interrupts compiling = IC interrupts at end of program.
The types of compilers are 2 - IIC, 4 - ABL, 6 - EMA, 8 - AA, 10 - MAC 12 - CC and 14 - HARTRAN)

16.51.07

(this is the time the program finished execution)

12.7.1 WORKING STORE LAYOUT FOR MUSE

0-22.4	APPLY TO JOB IN CURRENT CONTROL
0	EXTRACODE WORKING SPACE
5	PROCESSED JOB DESCRIPTION ADDRESS
5.4	SELECTED TAPE (tape currently selected for variable tape transfers)
6	MAIN PROGRAM CONTROLS (6: extracode control of job, 6.4: $\frac{1}{2}$ V4 - V6, when supervisor entered, record of M/E digit and B test register)
7	INPUT / OUTPUT RECORD (location in block directory of the current block of the currently selected input / output)
8-14	INPUT EXTRACODE WORKING SPACE (ALTERNATE HALF WORDS)
8.4-14.4	OUTPUT EXTRACODE WORKING SPACE (ALTERNATE HALF WORDS)
8	NUMBER OF CURRENT INPUT STREAM (number set up in job description)
8.4	NUMBER OF CURRENT OUTPUT STREAM (even if internal code and odd if binary),
9	HALF WORD CURRENTLY BEING UNPACKED (the next character due to be passed to the main program is in bits 23-18)
9.4	HALF WORD CURRENTLY BEING PACKED (the next available character space is in bits 23-18, the last available space contains 4.0 until it is overwritten by the last character)
10	ADDRESS IN R595 (bits 2-0 indicates character to be passed to main program next)
10.4	OUTPUT WELL ADDRESS (address of separator at begining of current record)
11	ADDRESS IN R595 (even if internal code, odd if binary)
11.4	RECORD COUNT (current count of complete records output fron job)
12	INPUT PERIPHERAL V STORE ADDRESS
12.4	OUTPUT PERIPHERAL V STORE ADDRESS
13	ADDRESS NEXT HALF WORD IN INPUT WELL
13.4	ADDRESS NEXT HALF WORD IN OUTPUT WELL
14	END ADDRESS OF THE CURRENT RECORD
14.4	END ADDRESS OF SPACE IN THE OUTPUT WELL
15	TRAP ADDRESS
15.4	PRIVATE MONITOR ADDRESS
16	BLOCK LIMIT
16.4	BLOCK MONITOR (extracode 1135 parmeters, jump to n when block $\geq$ ba defined, 16 bits 22-12 = ba, 16.4 = n)
17	PROGRAM BRANCH INDICATOR (if negative no program branching, otherwise contains page number of program branch block)
17.4	CURRENT DUMP DESCRIPTION (normally zero, extracode 1116 ''do not dump'' sets this negative, extracode 1115 bits 14-9 = n, 8 bits 8-2 = Ba)
18	WAIT TIME (time spent waiting for tape transfers, bits 23-1 in 0.1 seconds)
18.4	LIMIT WAIT TIME (execution time - computing time, as specified in job description)

12.7.1 Continued

19	UNIT TIME (temporary dump of instruction counter timer whilst in supervisor)
19.4	CHECK TIMER (maximum computing time on instruction counter interrupts
20	OVER FLOW CHECK
20.4	OVER ALL TIMER (overflow check, sets counter L (local) < T (total) 19.4 = L , 20 = T-L)
21	NUMBER OF DRUM TRANSFERS
21.4	NUMBER OF TAPE TRANSFERS
22	NUMBER OF RESERVED TAPES
22.4	COMPILIER NUMBER
23-44.4	PERIPHERAL PRIVATE STORE ADDRESSES
23	TR5 0
23.4	TELETYPE 0
24	TR5 1
24.4	TELETYPE 1
25	TR5 2
25.4	TELETYPE 2
26	TR5 3
26.4	TELETYPE 3
27	0.1
27.4	0.1
28	0.1
28.4	0.1
29	0.1
29.4	0.1
30	0.1
30.4	GRAPHICAL OUTPUT 0
31	TR5 8 (number 8 value occurs if a peripheral look at ne has zero on its V line, when this occurs a fixed store routine is entered which does nothing)
31.4	TELETYPE 8
32	TR7 0
32.4	CREED 3000 0
33	CARD READER 0
33.4	CARD PUNCH 0
34	CARD READER 1
34.4	X.RAY
35	0.1
35.4	0.1
36	TELEPRINTER 0
36.4	ANELEX PRINTER 0
37	TELEPRINTER 1
37.4	ANELEX PRINTER 1
38	0.1
38.4	GRAPHICAL OUTPUT 8
39	0.1
39.4	0.1
40	TR7 8
40.4	CREED 3000 8
41	CARD READER 8
41.4	CARD PUNCH 8

12.7.1 Continued

42		0.1
42.4		0.1
43		0.1
43.4		0.1
44		TELEPRINTER 8
44.4		ANELEX PRINTER 8
45-247.4		PRIVATE PERIPHERAL STORES address relative to first half word
0		V STORE ADDRESS OF PERIPHERAL (less * 6)
0.4		return address (when input / output finished contains reason why return has been made
	0	requested length of input / output completed
	0.4	end sequence *** on paper tape
	1	disabled
	1.4	no tape in TR5 or TR7
	2	overdue
	2.4	check fail card punch
	3	paper required for, anelex, teletype, teleprinter, creed 3000
	3.4	no cards in card punch
	4	punching error detected by TR5, TR7 or card reader
	4.4	no cards in card reader
	5	overflow - anelex
	5.4	punching 8,7 on first column of card
	6	tape out TR7
	6.4	check failed creed 3000
	7	check failed card reader a description of the fault is normally printed out on the teleprinter)
1		STARTING ADDRESS (of input / output buffer less * 7)
1.4		END ADDRESS (of input / output buffer less * 7)
2		CURRENT ADDRESS (with input contains address of next available store space, with output contains address of next character - reserved block number)
2.4		END ADDRESS OF STORE (reserved block number less * 7)
3		CURRENT BUFFER ADDRESS FOR INTERNAL ROUTINES (less * 7)
3.4		CURRENT INPUT / OUTPUT BUFFER FOR EXTRACODE ROUTINES (less * 7)
4		INPUT / OUTPUT RECORD (with input contains address (less * 7) reserved for next

12.7.1 Continued

separator, with output contains number of characters remaining in current record)

4.4	INFORMATION ON STATE OF PERIPHERAL (see note on 0.4)
5	CODE CONVERSION INFORMATION
5.4	TEST (with input used to test for ***, with output contains carriage control character at end of record)
6	FAULT INFORMATION (on input contains information on action required when a punching fault detected, not used with output)
6.4	OUTPUT STREAM NUMBER (bits 8-3 = stream number, bits 23-9 count of current number of output blocks)

248	END OF PROGRAM EXTRACODE
250	TAPE FAULTS (bits 0-7 = channels 0-7 respectively, the appropriate bit is set to or 1 if transfer not completed after 0.1 to 0.2 seconds)
250.4	CRITERION FOR PROGRAM CHANGE (change if wait > n sectors)
251	COUNT OF PROGRAM CHANGES
251.4	TAPES FOR ACTION (bits 0-7 = channels 0-7, set to 1 during one second interrupt if deck or status changes)
252	PROGRAM CHANGE LOOP
256	0.1 SECOND CLOCK (bits 23 to 1)
256.4-272	BAND DIRECTORY (consecutive half-words, entry m for band m contains in bits 20-14 program number, bits 10-3 program band number)
272.4-280	CURRENT ORDER (word i = current order on tape channel i)
280.4-288	NEXT ORDER (word i = next order on tape channel i)
288.4-321	DECK DIRECTORY
321.4-312.4	DECK ALLOCATION (alternate half-words, one per deck with bits

23	=	1 if entry in use, 0 if not in use
22	=	1 if this entry assigned to deck 0 if not yet assigned
21	=	1 if deck out of action, 0 if all right
20-14	=	logical tape number for main program
13	=	1 if title has been changed
12	=	1 if Orion tape
11	=	1 if special action before title
10	=	1 if reserved
9	=	1 if long job, 0 if short
8-2	=	program number, 0 for supervisor tapes
1	=	1 if re-engage expected
0	=	1 if tape available)

12.7.1 Continued

313-320 CHECK TIME
(alternate half-words one per deck, contains n in
n second search, otherwise all 1's)

320.4-352 TAPE QUEUE

352.4-368 SPARE

368.4 CURRENT PROGRAM NUMBER IN STORE CONTROL
(this may be different from current program number in
supervisor control)

369 PROGRAM CHANGE MARKER
(non-zero if state of execute list has changed since last
on main control)

369.4 NON-EQUIVALENCE MARKER
(odd if non-equivalence has just occurred in main
program)

370 SUPERVISOR EXIT TRAP

370.4 PROGRAM SCAN EXIT
(variable exit at 1/202)

371-374.4 SWITCH DIRECTORY
(halfword per main program with bits
23 = 1 if branching
22-20 = location of dump in dump block
19 = 1 if compiling
18 spare
17-15 = program type, 1 tape, 2 common, 3 computing
14 = 1 if subject to dynamic time sharing ie
reallocation in execution list depending on use of tapes
after every four seconds)
13 = 1 if program has its' own clock ie its' timers
are updated only when it is in control
13-7 = instruction counter for scheduling
6-4 = priority, 0 top, 1 2 & 3 normal, 4 low
3-2 = spare
1-0 = monitor description

375-378.4 MONITOR DIRECTORY
(halfword per main program with bits
23 = 1 if in off-line trap
22 = 1 if page lockdown
21-12 = off line monitor decription
11-2 = page number if lockdown trap
1 = 0/1 if master process
0 = 1 if waiting off line trap

379-402.4 MAIN PROGRAM SHORT DUMPS
(3 words per program containing B100-B104 and SER return
address when program is halted in supervisor)
(note that there is no main program 0 and therefore
the supervisor uses these half-words for itself as follows
375 = number of free blocks in machine
379 = number of free entries in drum queue, if ≤ 0
drum queue is full
383 = address of last entry in drum queue
bit 0 = 0 if drum queue is empty
387 = address of first entry in drum queue, ie entry
in progress
391 = current SER dump address
395 = current SER entry
399 = current SER base)

12.7 Continued

403-427 TAPE SER QUEUE
427.4-467.4 SLOW SER QUEUE
468-579.4 BLOCK DIRECTORY
(consecutive half words, one per main store block
bits 23,0 = 00 unused entry, = 10 vacated entry
= 01 block in core store, = 11 block on drum
22-12 = block label
11-1 = sector number if block on drum
or 11-3 = page number if block in core store
newly defined blocks have sector number 2047, supervisor blocks are at the beginning of the block directory and main program blocks work back from 479.4,
the starting address for a programs block area is contained in the program store directory)

580-691.4 BLOCK STATUS DIRECTORY
(consecutive half-words one per main store block,
entry m corresponds to entry m in the block directory,
for main programs 23-3 = T, the last idletime, in instruction interrupts, when this block was on the drum
2-0 = lock out status
0 = not locked out
1 = drum transfer to cores
2 = drum transfer to drum
3 = transfer from peripheral, leave in cores
4 = transfer from peripheral, write to drum
5 = trasnsfer to peripheral, leave in cores
6 = transfer to peripheral, write to drum
7 = transfer to peripheral, lose
The status directory is also used to link blocks in the input well and when a job is awaiting execution)

692 PAGE DIRECTORY
(alternate half-words, entry p corresponding to page p with bits
23 = 1 if page empty, 0 if occupied
22 = 1 if page locked down, 0 otherwise
20-14 = program number of program owning the block, 0 for supervisor
13,1 = spare
12-2 = location in block directory, relative to start, of the block occupying page p
0 = 0 if page locked out, 1 otherwise)

692.4-723.4 CONTENTS OF PARS
(alternate halfwords, entry p corresponding to page p with bits
23 = 1 if locked out
22-12 = block label

12.7.1 Continued

```
724-755.4       PAGE TIMERS
756-915.4       DRUM QUEUE
                (each queue entry occupies five halfwords and begins
                at 841.4 working backwards with word
                0              =     information on return/sector number/duplicate
                                     block/page number
                0.4, bits 23-3 =     return address
                     2-1       =     SER queue, 10 top, 01 tape, 11 slow
                     0         =   0
                1, bit   23    =   1 if clear new block
                     22        =   1 if lockdown required
                     21        =   1 if don't change timer
                     20-14     =     program number
                     12-2      =     entry in block directory relative to start
                                     of area reserved for program P
                     13,1      =     spare
                     0          =  1 if operand, 0 if instruction
                1.4            =     address if drum transfer routine, the entry
                                     of the routine to deal with the next stage o
                                     this item
                2, bits 23-2=        address of next queue entry relative to start
                                     last item contains address of itself
                     1         =     spare
                     0         =   1 if item in this space, 0 if empty)
916-927.4       EMPTY SECTOR TABLE
                (entry of drum d and angle θ is in half-word 3d+θ/2
                bits 7-0 indicate the state of the relevant sectors on bands 7-0
                and are 1 if empty, 0 if used:, bits 23-8 are zero)
928             LOCAL TEST TIME
928.4           INCREMENT CLOCK
929-932.4       PROGRAM TIMER DIRECTORY
                (added to learning program from instruction counter interrupts
933-936.4       PROGRAM WAIT DIRECTORY
                (time spent waiting for current tape transfer)
937-940.4       LEADING HALT MARKERS
                (successive half-words, one for each standard reason for
                halt, full drum queue, full tape queue etc containing
                bits 23-21 = number of SERs halted for this reason in top
                             priority SER queue
                     20-14 = first main program halted for this reason
                     13-8  = number of SER's halted for this reason in tape
                             SER queue
                     7-0   = number of SER's halted for this reason in slow
                             SER queue)
941-947         HALT POSITIONS IN SER QUEUE
                (alternate half-words)
941.4           1 SECOND CLOCK
                (in bits 23-2)
942.4           DRUM CHECK
                (half-word to check completion of a drum transfer,
                set to zero when transfer started)
943.4           TOP SER QUEUE and IN SUPERVISOR SWITCH.
948             LEADING PROGRAM NUMBER
                (zero if no programs on execute list)
948.4           CURRENT PROGRAM NUMBER
                (in supervisor control)
949-952.4       PROGRAM STATUS DIRECTORY
                (half-word per main program with bits
                23       =   1 if halted for block
                22-12    =     reason for halt
                11       =   1 if tapes used
                10       =   1 if clearway job ie don't change execution of
                                this program till it is held up
```

12.7 |Continued

9 = 0/1, full recover/in supervisor only recover
8-2 = link to next job
1 = 1 if in supervisor
0 = 1 if free, 0 if halted)

953-956.4 PROGRAM STORE DIRECTORY
(half-word per main program with bits
23-13 = number of blocks reserved for this program -1
12-2 = start block area relative to start of block directory
1 = 1 if in processing made
0 = 0 if clear run blocks)

957 BLOCK LOCATION TABLE
(working space for B203 holding directory positions of blocks, available for use by other routines)

961 TIMED/TEST CRITERION

962 TIMED/TEST ENTRIES
(1 second routine scans 961, if < 0.1 second clock SER is put into slow queue with entry address 962, if >, 0.1 second clock do nothing)

963 SCHEDULER TIME
(time when dynamic scheduler is next to be entered, time in second units 0.1 in 23-1)

963.4 RETURN ADDRESS TO RESERVE BAND

964 DUMP TAPE POSITION
(digits 13-0 contain block address of next block available for writing on dump tape)

964.4 DUMP TAPE RECORD
(digits 23-3 contain channel of dump tape, digit 0 = 1 if tape on, 0 if tape off)

965.4 HIGHEST PAGE TO BE CONSIDERED
(digits 23-4 contain limit of pages for learning program to consider, usually 31)

966 ORION TAPE CHECK

966.4 PREVENT PREPARATION NEXT
(if zero can prepare magnetic tape orders in advance, otherwise do not prepare them)

967 STRING DIRECTORY 1
(variable tape extracode directory)

975 F1 F2
(digit i = channel i, F1 = 1 if end of block address interrupt expected, F2 = 1 is set by 0.1 second interrupt when F1 = 1 and monitor if F1 = F2 = 1. F1 and F2 cleared at end of block interrupt)

976 DISENGAGE RECORD
(copy of tape disengage digits, line 16*6003)

976.4 ADDRESS OF HIGHEST PAGE FOR INSTRUCTIONS

977 SPACES IN TAPE QUEUE
(digits 23-2 contain the number of vacant entries in magnetic tape queue)

977.4 DUMP FOR EXTRACODE CONTROL
(if parity occurs when in supervisor, B126 and 108-110 dumped and subsequently used by error routines. 977 is also used by input/output extracodes to record B126)

978 IN SUPERVISOR DUMP FOR B110

978.4 IN SUPERVISOR DUMP FOR B108

979 IN SUPERVISOR DUMP FOR B109

978 USE DIGIT TABLE
(half-word per page, bit 0 set to 1 if page used since last instruction counter interrupt then half-word moved up one place)

995 USE DIGITS AT LAST INTERRUPT

996 DRUM WORKING SPACE

997 WORKING SPACE

998 WORKING SPACE

12.7 |Continued

999	OPERATORS OUTPUT TELEPRINTER 0 (address less*7 working area of print routine)
999.4	OPERATORS OUTPUT TELEPRINTER 1
1000	STRING DIRECTORY 2 (V store tape directory contained in alternate half-words)
1000.4	F3 (alternate half words, contains 0,1, second clock when tape started in inter-block gap or on trailing block address when moving onto leading block address, 0 if not this condition. Checked by 1 second routine and monitor if F3 < time - 6 seconds) (alternate half-words)
1008	RECORD OF DUMP BLOCK (block on dump tape to which subsidiary store dumped on machine failure)
1009	SCHEDULER KEY WORD (bit 23 = 1 if scheduler active)
1009.4	FAULT COUNTER (digits 23-0 contain counter if number of interrupt type faults during monitoring procedure)
1010	MAIN FAULT RECORD (digits set for various types of machine fault)
1010.4	DRUM EXIT RECORD
1011	DRUM LOCATION RECORD (on drum fail records θ, b,d, & c using 2 octal digits each)
1011.4	DRUM FAULT RECORD (digits set for various types of drum fault)
1012	TOTAL NUMBER OF MAIN STORE BLOCKS AVAILABLE
1012.4	MISCELLANEOUS MARKER
1015.4	INTERRUPT JUMPS (alternate half-words)

12.7.2 WORKING STORE LAYOUT FOR LONDON AND NIRNS

0-22.4 APPLY TO JOB IN CURRENT CONTROL

0-4.4 EXTRACODE WORKING SPACE (99/900)
5 PROCESSED JOB DESCRIPTION ADDRESS (4/261)
5.4 SELECTED TAPE (3/489)
6-6.4 MAIN PROGRAM CONTROLS (6/201)
(6: extracode control of job, 6.4: $\frac{1}{2}$V4-V6, when supervisor entered, record of M/E and B test register)
7-7.4 INPUT / OUTPUT RECORD (12/227)
(location in block directory of the current block of the currently selected input / output)
8-14 INPUT EXTRACODES WORKING SPACE (70/595) - (76/595)
(alternate half words)
8.4-14.4 OUTPUT EXTRACODES WORKING SPACE (70/596) - (76/596)
(alternate half words)
8 NUMBER OF CURRENT INPUT STREAM (70/595)
(number set up in job description)
8.4 NUMBER OF CURRENT OUTPUT STREAM (70/596)
(even if internal code, and odd if binary)
9 HALF WORD CURRENTLY BEING UPACED (71595)
(the next character due to be passed to the main program is in bits 23 - 18)
9.4 HALF WORD CURRENTLY BEING PACKED (71/596)
(the next available character space is in bits 23 - 18, the last available space contains 4.0 until it is overwritten by the last character.)
10 ADDRESS IN R595 (72/595)
(bits 2 - 0 indicate next character to be passed to main program)
10.4 OUTPUT WELL ADDRESS (72/596)
(address of separator at beginning of current record.
11 ADDRESS IN R595 (73/595)
(even if internal code, odd if binary)
11.4 RECORD COUNT (73/596)
(current count of complete records output from job
12 INPUT PERIPHERAL V-STORE ADDRESS (74/595)
12.4 OUTPUT PERIPHERAL V-STORE ADDRESS (74/596)
13 ADDRESS OF NEXT HALF WORD IN INPUT WELL (75/595)
13.4 ADDRESS OF NEXT HALF WORD IN OUTPUT WELL (75/596)
14 END ADDRESS OF THE CURRENT RECORD (76/596)
14.4 END ADDRESS OF SPACE IN THE OUTPUT WELL (76/596)

WORKING STORE LAYOUT FOR LONDON AND NIRNS

15 TRAP ADDRESS (7/700)

15.4 PRIVATE MONITOR ADDRESS (4/702)

16 BLOCK LIMIT (19/203)

16.4 BLOCK MONITOR (31/203)
(extracode 1135 parameters; jump to n when block $\geq$ ba where word 16 bits 22-12 = ba, word 16.4 = n)

17 PROGRAM BRANCH INDICATOR (7/202)
(if negative no program branching, otherwise contains page number of program branch block)

17.4 CURRENT DUMP DESCRIPTION (44/711)
(normally zero, extracode 1116 'do not dump' sets this negative, extracode 1115 sets bits 14-9 = n

18 WAIT TIME (9/227)
(time spent waiting for tape transfers, bits 23-1 in 0.1 seconds)

18.4 LIMIT WAIT TIME (10/227)
(execution time - computing time, as specified in job description)

19 UNIT TIMER (18/226)
(temporary dump of instruction counter timer whilst in supervisor)

19.4 CHECK TIMER (21/303)
(maximum computing time on instruction counter interrupts)

20 OVERFLOW CHECK (7/704)

20.4 OVERALL TIMER (8/704)
(overflow check, sets counter L (local) <T (total) 19.4 = L, 20 = T-L)

21 NUMBER OF DRUM TRANSFERS (45/314)

21.4 NUMBER OF TAPE TRANSFERS (66/400)

22 NUMBER OF RESERVED TAPES (46/498)

22.4 COMPILER NUMBER (13/290)

23 ORION TAPE CHECK (4/242)

23.4 SPACES IN TAPE QUEUE (61/400)
(digits 23-2 contain the number of vacant entries in magnetic tape queue)

24-31.4 CURRENT ORDER (51/400)
(word i = current order on tape channel i)

32-39.4 FOLLOWING ORDER (52/400)
(word i = next order on tape channel i)

40-47.4 ALPHA (54/400)

48-55.4 BETA (55/400)

56-63.4 CHECK TIME (56/400)
(alternate half words, one per deck; contains n in n seconds search, otherwise all 1's)
(<u>NOTE</u> This is insufficient for 16 decks and it is probable that the private store of tape readers 0 and 1 i.e. 671 - 680.4 will be moved to the main store and this space used).

64-79.4 DECK ALLOCATION (58/400) or (5/221)
(alternate half-words, one per deck with bits :

WORKING STORE LAYOUT FOR LONDON AND NIRNS

	23 = 1	if entry in use, 0 otherwise
	22 = 1	if this entry assigned to deck
	= 0	if not yet assigned
	21 = 1	if deck out of action, 0 if alright
	20-14 =	logical tape number for main program
	13 = 1	if title has been changed
	12 = 1	if Orion tape
	11 = 1	if special action before title
	10 = 1	if reserved
	9 = 1	if long job, 0 if short
	8-2 =	program number, 0 if supervisor tape
	1 = 1	if re-engage expected
	0 = 1	if tape available)

56.4-63.4 F3 (59/400)
(alternate half words, contains 0.1 second clock when tape started in inter-block gap or on trailing block address when moving onto leading block address:, 0 if not this condition. Checked by 1 second routine and monitor if F3 < time - 6 secs.)

64.4-79.4 STRING DIRECTORY 2 (77/400)
(V-store tape directory contained in alternate half-words)

80-95.4 DECK DIRECTORY (53/400)

96-111.4 STRING DIRECTORY 1 (76/400)
(variable tape extracode directory)

112-143.4 TAPE QUEUE (60/400)

144-144.4 F1F2 (67/400)
(digit i = channel i, F1=1 if end of block address interrupt expected, F2 = 1 is set by 0.1 second interrupt when F1 = 1 and monitor if F1 = F2 = 1. F1 and F2 cleared at end of block interrupt)

145 DISENGAGE RECORD (68/400)
(copy of tape disengage digits, line 16*6003)

145.4 SPARE
(extension of 68/400 if 32 decks)

146 PREVENT PREPARATION NEXT (69/400)
(if zero can prepare magnetic tape orders in advance, otherwise do not prepare them

146.4 SCHEDULER KEYWORD (3/660)
(bit 23 = 1 if scheduler active)

147-147.4 RECORD OF DUMP BLOCK (16/223)
(block on dump tape to which subsidiary store dumped on machine failure)

148 DUMP TAPE RECORD (11/244)
(digits 23-3 contain channel of dump tape, digit 0 = 1 if tape on, 0 if tape off)

WORKING STORE LAYOUT FOR LONDON AND NIRNS

148.4 DUMP TAPE POSITION (12/244)
(digits 13-0 contain block address of next block available for writing on dump tape)

149-152.4 PROGRAM STORE DIRECTORY (4/203)
(half word per main program with bits
23-13 = number of blocks reserved for this program - 1
12-2 = start of block area relative to start of block directory
1 = 1 if in processing mode
0 = 0 if clear run blocks)

153 LEADING PROGRAM NUMBER (9/204)(9/207)
(zero if no programs on execute list)

153.4 CURRENT PROGRAM NUMBER (5/203)
(in supervisor control)

154-157.4 PROGRAM STATUS DIRECTORY (9/204)
(half-word per main program with bits
23 = 1 if halted for block
22-12 = reason for halt
11 = 1 if tapes used
10 = 1 if clearway job, i.e. don't change execution of this program till it is held up.
9 = 0/1 full recover / in supervisor only recover
8-2 = link to next job
1 = 1 if in supervisor
0 = 1 if free, 0 if halted

158 NUMBER OF FREE BLOCKS (10/205)

158.4-161.4 PROGRAM MONITOR DIRECTORY 0.4(8/227)
(half-word per main program with bits
23 = 1 if in off-line trap
22 = 1 if page locked down
21-12 = off line monitor description
11-2 = page number if lockdown trap
1 = 0/1 if master process
0 = 1 if waiting off-line trap)

162-165.4 PROGRAM SWITCH DIRECTORY (15/204)
(half-word per main program with bits
23 = 1 if branching
22-20 = location of dump in dump block
19 = 1 if compiling
18 = spare
17-15 = program type; 1 tape, 2 common, 3 computing
14 = 1 if subject to dynamic time sharing, i.e. reallocation in execution list depending on use of tapes after every four seconds
13 = 1 if program has its' own clock, i.e. its' timers are updated only when it is in control.

WORKING STORE LAYOUT FOR LONDON AND NIRNS

	13-7 =	instruction counter for scheduling
	6-4 =	priority; 0 top, 1, 2 and 3 normal, 4 low
	3-2 =	spare
	1-0 =	monitor description)
166		COUNT OF INSTRUCTIVE COUNTER INTERRUPTS (2/301)
166.4-169.4		PROGRAM TIMER DIRECTORY (20/303) (added to learning program from instruction counter interrupts)
170-173.4		PROGRAM WAIT DIRECTORY (13/205) (time spent waiting for current tape transfers
174-197.4		MAIN PROGRAM SHORT DUMPS (7/204) (3 words per program containing B100-104 and SER return address when program is halted in supervisor) Note that there is no main program 0 and therefore the supervisor uses these half-words for itself as follows:
174		NUMBER OF FREE ENTRIES IN DRUM QUEUE (4/315)
178		LAST ENTRY IN DRUM QUEUE (3/315) (bit 0 = 0 if drum queue is empty)
182		FIRST ENTRY IN DRUM QUEUE (2/315) (entry in progress)
186		CURRENT SER DUMP ADDRESS (12/213)
190		CURRENT SER ENTRY ADDRESS (5/201)
194		CURRENT SER BASE (7/201)
198		CURRENT PROGRAM IN STORE CONTROL (9/205) (this may be different from current program number in supervisor control)
198.4		PROGRAM CHANGE MARKER (6/202) (non-zero if state of execute list has changed since last on main control)
199		NON-EQUIVALENCE MARKER (4/204) (odd if non-equivalence has just occurred in main program)
200		PROGRAM SCAN EXIT (5/202) (variable exit at (1/202))
200.4		LENGTH USE DIGITS (4/303)
201-204.4		LEADING HALT MARKERS (2/213) (successive half-words, one for each standard reason for halt, full drum queue, full tape queue, etc. containing bits 23-21 = number of SER's halted for this reason in top priority SER queue 20-14 = first main program halted for this reason 13-8 = number if SER's halted for this reason in tape SER queue 7-0 = number of SER's halted for this reason in slow SER queue)
205-207		HALT POSITIONS IN SER QUEUES (3/213) (alternate half-words)

WORKING STORE LAYOUT FOR LONDON AND NIRNS

```
205.4       LOCAL TEST TIME  (12/222)
206.4       INCREMENT CLOCK  (12/229)
207.4       1 SEC. CLOCK  (6/229)
            (in bits 23-2)
208         SCHEDULER TIME  (14/230)
            (time when dynamic scheduler is next to be entered, time in 0.1
             sec. units in digits 23-1)
208.4       0.1 SECOND CLOCK  (5/229)
            (bits 23-1)

209-212.4   PROGRAM CHANGE LOOP  (11/226)
213-213.4   TAPES FOR ACTION  (8/230)
            (bits 0-7 = channels 0-7, set to 1 during one second interrupt if
             deck or status changes)
214-214.4   TAPE FAULTS  (9/229)
            (bits 0-7 = channels 0-7 respectively, the appropriate bit is set to
             0 or 1 if transfer not completed after 0.1 - 0.2 seconds)
215-216.4   OPERATORS OUTPUT CHANNELS  (3/220)
            (address less *7 of working area for each output)
217-228.4   TABLE OF EMPTY SECTORS  (2/313) = -96.4(15/229)
            (entry of drum d and angle θ is in half-word 3d + θ/2 bits 7-0
             indicate the state of the relevant sectors on bands 7-0 and are
             1 if empty, 0 if used. Bits 23-8 = 0)
229-308.4   DRUM QUEUE  (20/315)
            (each queue entry occupies 5 half-words and begins at 306.4 working
             backwards with word
            0  =  information on return / sector number / duplicate block / page
                  number
            0.4 bits 23-3 = return address
                      2-1 = SER Queue; 10 top, 01 tape, 11 slow
                       0  = 0
              1 bits 23   = 1  if clear new block
                     22   = 1  if lockdown required
                     21   = 1  if don't change timer
                  20-14   =    program number P
                     12-2 =    entry in block directory relative to start of
                               area reserved for program P
                     13,1      spare
                        0 = 1  if operand, 0 if instruction
           1.4 = address if drum transfer routine, the entry of the routine
                 to deal with the next stage of this item.
             2 bits 23-2 = address of next queue entry relative to start of
                           queue, last item contains address of itself.
                     1    =    spare
                     0    = 1  if entry in this space, 0 if empty)
```

WORKING STORE LAYOUT FOR LONDON AND NIRNS

309-313	TOP SER QUEUE + IS SWITCH (10/201)
313.4	MISCELLANEOUS MARKER (15/229) = 96.4(2/313)
314-317.4	BLOCK LOCATION TABLE (35/203) (working space for R203 et al. holding directory positions of blocks, available for use by other routines)
318-318.4	DRUM WORKING SPACE (25/314)
319-319.4	WORKING SPACE (7/203)
320-320.4	WORKING SPACE (8/203)
321-345	TAPE SER QUEUE (9/201)
345.4	DUMP FOR E-CONTROL (16/241) (if parity occurs when in supervisor, B126 and 108-110 are dumped and subsequently used by error routines. Also used by input / output extracodes to record B126)
346	IN SUPERVISOR DUMP FOR B108 (11/243)
346.4	IN SUPERVISOR DUMP FOR B109 (12/243)
347	IN SUPERVISOR DUMP FOR B110 (13/243)
347.4	RETURN ADDRESS TO RESERVE BAND (4/334)
348-363.4	BAND DIRECTORY (6/321) (consecutive half-words, entry m for band m contains in bits 20-14, program number; bits 10-3 program band no.)
364	CRITERION FOR PROGRAM CHANGE (15/223) (change if wait > n sectors)
364.4	COUNT OF PROGRAM CHANGES (16/227)
365-365.4	TIMED / TEST CRITERION (11/230)
366-366.4	TIMED / TEST ENTRIES (12/230)
367-510.4	BLOCK DIRECTORY (2/203) (For 48K core store and 4 drums) (consecutive half words, one per main store block. bits 23,0 = 00 unused entry, = 10 vacated entry, = 01 block in core store, = 11 block on drum 22-12 = block label 11-1 = sector number if block on drum or 11-3 = page number if block in core store Newly defined blocks have sector number 2047, supervisor blocks are at the beginning of the block directory and main program blocks work backwards from 378.4. The starting address for a program's block area is contained in the program store directory)
511-513	RESTART AND TAPE (98/198) (half words 0 - 3 not used)
511	SPARE
511.4	LONG JOB RECORD (1/999)
512	NUMBER OF MAIN STORE BLOCKS AVAILABLE (8/334)
512.4	NUMBER OF JOBS INPUT (2/999)
513.4-515.4	SPARE

WORKING STORE LAYOUT FOR LONDON AND NIRNS

516-523.4 PRIVATE STORE OF ANELEX 0

(together with the private stores of TR5's 0 and 1 this the only peripheral private store contained in the subsidiary store. The remainder are to be found in block *3663 which is permanently locked down in core store

Contents of the various half-words relative to the start of the private store are

0 V-STORE ADDRESS OF PERIPHERAL
(less *6)

0.4 RETURN ADDRESS
(when input / output finished contains reason why return has been made

0	requested length of input / output completed	
0.4	ending sequence *** on paper tape	
1	disabled	
1.4	no tape in TR5, TR7	
2	overdue	
2.4	check fail card punch	
3	paper required for anelex, teletype, teleprinter	
3.4	no cards in card punch	
4	punching error detected by TR5 or card reader	
4.4	no cards in card reader	
5	overflow - anelex	
5.4	punching 7/8 on first column of card	
6	tape out TR7	)not relevant on
6.4	check failed Creed 3000	)LONDON and NIRNS
7	check failed card reader	

A description of the fault is generally printed out on the operators output

1 STARTING ADDRESS
(of input / output buffer less *7)

1.4 END ADDRESS
(of input / output buffer less *7)

2 CURRENT ADDRESS
(for input contains address of next available store space for output contains address address of next character - reserved block number)

2.4 END ADDRESS OF STORE
(reserved block number less *7)

3 CURRENT BUFFER ADDRESS FOR INTERNAL ROUTINES
(less *7)

3.4 CURRENT INPUT / OUTPUT BUFFER FOR EXTRACODES
(less *7)

4 INPUT / OUTPUT RECORD
(with input contains address less *7 reserved for next separator, with output contains number of characters remaining in current record)

WORKING STORE LAYOUT FOR LONDON AND NIRNS

separator, with output contains number of characters remaining in current record)

4.4 INFORMATION ON STATE OF PERIPHERAL
(see note on 0.4)

5 CODE CONVERSION INFORMATION

5.4 TEST
(with input used to test for ***, with output contains carriage control character at end of record)

6 FAULT INFORMATION
(on input contains information on action required when a punching fault detected, not used with output)

6.4 OUTPUT STREAM NUMBER
(bits 8-3 = stream number, bits 23-9 count of current number of output blocks)

524-525 SPARE

525.4-526 SYSTEM TAPES (4/999)

526.4 INPUT WELL COUNT (3/999)

527-670.4 BLOCK STATUS DIRECTORY (3/203)
(consecutive half-words, one per main store block, entry m corresponds to entry m in the block directory, for main programs bits 23-3 = T, the last idle time, in instruction counter interrupts, when this block was on the drum. bits 2-0 = lock-out status

0	=	not locked
1	=	drum transfer to cores
2	=	drum transfer to drum
3	=	transfer from peripheral, leave in cores
4	=	transfer from peripheral, write to drum
5	=	transfer to peripheral, leave in cores
6	=	transfer to peripheral, write to drum
7	=	transfer to periheral, lose

The status directory is also used to link blocks in the input well, and also when a job is awaiting execution)

671-678.4 PRIVATE STORE OF TR5 0) Those may be replaced by the Check
679-686.4 PRIVATE STORE OF TR5 1) Time entries, at present 56-63-4

687-782 PAGE DIRECTORY (6/203)
(alternate half words, entry p corresponding to page p with

bits	23	=	1	if page empty, 0 if occupied
	22	=	1	if page locked down, 0 otherwise
	20-14	=		program number of program owning the block
		=	0	for supervisor block
	13,1			spare
	12-2	=		location in block directory, relative to the start, of the block occupying page p
	0	=	0	if page locked out, 1 otherwise)

WORKING STORE LAYOUT FOR LONDON AND NIRNS

687.4-782.4	CONTENTS OF PAR's (2/312) (alternate half words, entry p corresponding to page p with bits 23 = 1 if locked out 22-12 = block label (i.e. PAR setting)
783-810.4	PERIPHERAL SUBSIDIARY STORE TABLE (5/599)
783	TR5 0
783.4	Teletype 0
784	TR5 1
784.4	Teletype 1
785	TR5 2
785.4	Teletype 2
786	TR5 3
786.4	Teletype 3
787	0.1
787.4	0.1
788	0.1
788.4	0.1
789	0.1
789.4	0.1
790	0.1
790.4	Graphical output 0
791	TR5 8 (number 8 value occurs if a peripheral look at me has zero in its V-store line, when this occurs a fixed store routine which does nothing is entered
791.4	Teletype 8
792	TR7 0
792.4	Creed 3000 0
793	Card reader 0
793.4	Card punch 0
794	Card reader 1
794.4	0.1
795	0.1
795.4	0.1
796	Teleprinter 0
796.4	Anelex 0
797	Teleprinter 1
797.4	Anelex 1
798	0.1
798.4	Graphical output 8
799	0.1
799.4	0.1
800	TR7 8
800.4	Creed 3000 8
801	Card reader 8
801.4	Card punch 8

WORKING STORE LAYOUT FOR LONDON AND NIRNS

802	0.1
802.4	0.1
803	0.1
803.4	0.1
804	Teleprinter 8
804.4	Anelex 8
811	DRUM FAULT RECORD (5/340) (digits set for various types of drum fault
811.4	DRUM LOCATION RECORD (11/340) (on drum fail records 0-, b, d, and C using 2 ocatal digits each)
812	DRUM EXIT RECORD (12/340)
812.4	MAIN FAULT RECORD (7/241) (digits set for various types of machine fault)
813	FAULT COUNTER (8/241) (digits 23-0 contain counter of number of interrupt type faults during monitor procedure)
813.4	ADDRESS OF HIGHEST PAGE FOR INSTRUCTIONS (48/314)
814-814.4	SPARE
815-910.4	PAGE TIMERS (8/303)
911-959.4	SLOW SER QUEUE (8/201)
960-1010.4	USE DIGITS (6/303) (half word per page, bit 0 set to 1 if page used since last instruction counter interrupt then half-word moved up one place)
1011-1013.4	USE DIGITS AT LAST INTERRUPT (7/303)
1014	USED TO ENTER 1054 EXTRACODE
1014.4	NUMBER OF JOBS COMPLETED (16/655)
1015	1060 EXTRACODE
1015.4-1023.4	INTERRUPT JUMPS (alternate half words)
1016	1117 EXTRACODE
1017	SPARE
1018	HIGHEST PAGE TO BE CONSIDERED (11/304)
1019	1050 EXTRACODE
1020	1064 EXTRACODE
1021	1065 EXTRACODE
1022	1066 EXTRACODE
1023	1067 EXTRACODE

1.5.64

12.8 PERIPHERAL IDENTIFIERS

Each peripheral has an octal identifier which is digits 11-3 of its V-store address.

160	TR5 0
161	TR5 1
162	TR5 2
163	TR5 3
200	TELETYPE 0
201	TELETYPE 1
202	TELETYPE 2
203	TELETYPE 3
101	ANELEX
040	TR7 0
140	CREED 3000 0
000	CARD READER 0
220	CARD PUNCH
260	TELEPRINTER 0
261	TELEPRINTER 1

12.9 OPERATOR REQUESTS

All requests must be read in on 160 (operators input) unless operators input is transferred to some other device.

Tabs and Spaces are ignored everywhere except in job titles.
Erases, Backspaces, set and case changes are ignored everywhere.

Tape Units : referred to as T0, T1 etc. but can only be specified in requests 23, 24, 25 (note : if deck already engaged deck allocation directory is correctly set but no printing will take place until supervisor has finished with deck)

XR00 : GIVE JOB TOP PRIORITY

XR01 : GIVE JOB HIGH PRIORITY

XR02 : GIVE JOB NORMAL PRIORITY

XR03 : GIVE JOB LOW PRIORITY

```
[e.g.  XR01
       (Title)
       ***Z  ]
```

XT20 : TRANSFER OPERATORS INPUT

XT21 : TRANSFER OPERATORS OUTPUT

XT22 : TRANSFER TAPE OPERATORS OUTPUT

```
[e.g.  XT20       XT21
       163        200
       ***Z       ***Z  ]
```

XR23 : REMOVE PERIPHERAL FROM SYSTEM

XR24 : STOP AND TRANSFER PERIPHERAL

XR25 : RECONNECT PERIPHERAL INTO SYSTEM

```
[e.g.  XR23       XR25       XR23
       161        160        T7
       ***Z       ***Z       ***Z  ]
```

XR40 : PERMANENTLY LOCK PERIPHERALS

XR41 : LINK

XR42 : LOCK

XR43 : MAKE SEMI REMOTE

XR44 : ASSOCIATE

XR45 : MAKE NORMAL

```
[e.g.  XR40       XR45
       160/201    162/101
       ***Z       ***Z  ]
```

XR46 : GIVE STATE OF PERIPHERALS

12.9 Continued

```
XR60-63   :      CHANGE SUPERVISOR PARAMETERS.
                     Parameters specified in various ways.
XR60        i)   REMOVE CORE STORE  :
a/b
                 a and b determine pages of store to
                 be removed  (don't try and remove page
                 containing *3411)  a need not be > b
XR62       ii)   REPLACE CORE STORE  :
a/b

XR61      iii)   REMOVE DRUM STORE
a/b/cdefgh
                 where a is drum number    (0,1, 2 or 3)
                       b is band number    (0 - 7)
                    c-h are sector numbers ( all in 0 - 5)
                 c-h need not all be specified.
                 Alternatively :
                 a         ->  remove drum
                 a/b       ->  remove band
         [but NOT  AD a/bc  remove bands]

XR63        iv)  REPLACE DRUM STORE
a/b/cdefgh

         [Examples
                 XR60             remove pages 3 - 7 inc.
                 7/3
                 ***Z

                 XR61
                 2/1/024                  remove sector 0,2 4 on
                 ***Z             band 1 of drum 2

                 XR63             replace drum 0.
                 0
                 ***Z

XR64        :    BATCH COMPILE
                 ie  keep compiler in store

XR65        :    END BATCH COMPILING

XR66        :    ACCEPT EXTRACODE PROGRAM

        [Examples
                 XR64             batch compile with ABL
                 ABL
                 ***Z

                 XR66             next program on operators'
                 ***Z             input is allowed to
                                  use extracode control ]
```

12.9 Continued

ACTION ARISING FROM OPERATOR REQUESTS.

All operator requests initiate some printing on the main operators output, which in general is the teleprinter 260. For most requests this printing is immediate but in some cases it may be delayed, for example it may be necessary to read from magnetic tape to obtain the necessary information. With one exception requests are only accepteable on the operators input, which is defined to be reader 160 at the start of day. The magnetic tape operators output is defined to be teleprinter 261.

FAULT PRINTING.

Whenever a fault is detected in an operator request the heading

OPERATOR REQUEST FN

is printed, where N is the fault number. If the fault has occurred in the first line of the request this line is then printed direct from the input buffer; the fault itself will lie in the request identifier but it the operator has put any comment after the identifier then this also will be printed. If the fault has occurred in the second line then after the heading has been printed the request identifier is printed on a new line in its reconstructed form. This will then be followed on a new line by the faulty second line printed direct from the input buffer.

Due to the restricted character set on the teleprinter certain characters that are available in the flexowriter code are not printable on the teleprinter. However, all the information necessary to implement any request is available both in flexowriter and teleprinter codes. All run-out characters then these will appear on the teleprinter as square brackets

Multiple request faults are not detected, ie once a fault has been found fault printing is initiated, and further faults that may present are ignored.

Faults common to all requests are as follows:

F0 : Buffer exceeded. All requests must be less than 512 characters in length; this includes shift, set change and spurious run out characters.

F1 : Character missing in first line of request e.g. only one octal digit in request number. The complete unreconstructed first line (together with any comment that may be present) of the requst is printed.

F2 : Request not read in on operators input, or, Unassigned request, or,
XR or XT followed by wrong request number.

The meanings of the remaining fault numbers 3 - 7 are dependent on the request numbers.

Fault printing for these faults consists of 3 lines; the first is the heading, the second the reconstructed request identifier (if any comment is present htis is not printed), and the third is the unreconstructed second line.

Requests 00 - 03 (job title requests)

F3 : title too long. A title must not contain more than 80 characters.

If the job title is not recognized the following is printed:

XR 0 -
(title)
TITLE NOT RECOGNIZED.

Requests 20 - 22 (dealing with operators input , output channels)

F3 : format error
F6 : wrong type of peripheral specified in request
F4 : unassigned peripheral
F5 : specified peripheral out of use

Requests 23 - 25 (single peripheral requests)

F4 : unassigned peripheral
F3 : format error
F6 : peripheral not in correct state for request
F5 : magnetic tape deck number $\geq$ 8

Requests 40 - 45 (linking peripherals requests)

F3 : format error
F4 : unassigned peripheral
F5 : peripheral out of use
F6 : either both input or both output peripherals specified
F7 : output device stated first in request
F7 : output device already waiting to be linked.

PRINTING ARISING FROM SUCCESSFUL REQUESTS.

Requests to change job priorities produce the following response:

JOB TITLE
JOB GIVEN *** PRIORITY

where *** is top, high, normal or low according to whether the request number was 00, 01, 02 or 03 respectively.

XR 04 : STARE OF JOB
JOP TITLE
JOB ON X LIST, Y PRIORITY
where X is either execute, active or job and Y is either top, normal or low.
If the job is on the execute list then a further line
INSTRUCTIONS OBEYED Z
is also printed. Z is the current number of instruction counter interrupts for this program.

XT 20 : OPERATORS INPUT TRANSFERRED X, where X is the peripheral number

XR 21 : OPERATORS OUTPUT TRANSFERRED TO X

XR 22 : TAPE OPERATORS OUTPUT TRANSFERRED TO X

XR 23 : ABC FREED

XR 24 : ABC STOPPED AND TRANSFERRED

XR 25 : ABC RECONNECTED

If a magnetic tape deck is specified in XR23 or XR24 the response will be

DECK N OUT OF ACTION

and for XR25

DECK N RECONNECTED.

XR 40 : PERIPHERALS ABC AND DEF PERMANENTLY LOCKED, where ABC and DEF are the peripheral numbers.

XR 41 : PERIPHERALS ABC AND DEF LINKED

XR 42 : PERIPHERALS ABC AND DEF LICKED

XR 43 : PERIPHERALS ABC AND DEF MADE SEMI REMOTE

XR 44 : PERIPHERALS ABC AND DEF ASSOCIATED

XR 45 : PERIPHERALS ABC AND DEF NORMALISED

XR 46 : produces a variable amount of printing according to the state of the peripheral system.

The first line to be printed is always

STATE OF PERIPHERALS

Following this, the single line

OPERATORS INPUT AND OUTPUT ARE ABC DEF GHI

where ABC is the operators input, DEF is the main operators output and GHI is the magnetic tape operators output.

Following this, each on a new line are statements

ABC DESABLED

for each relevant peripheral which requires attention.
Next there follow statements of the form

ABC AND DEF ***

where xxx may be PERMANENTLY LOCKED
LINKED
LOCKED
SEMI REMOTE
ASSOCIATED.

The final line to be printed is always

REMAINING PERIPHERALS NORMAL

and this line is printed irrespective of the fact that there may be no 'remaining' peripherals.

XR 47 : produces a variable amount of output according to the state of the supervisor parameters.
The first line printed is always
STATE OF SUPERVISOR PARAMETERS
The next line(s) are optional; they contain up to 16 decimal numbers corresponding to the page numbers of the core store that are not in use. The lines are preceeded by CORE PAGES and terminated by U/S.
Next follows drum information, each drum number is printed in the form DRUM n (n=0-3) whether or not there are any useless sectors on it. If there are not more than 5 sectors out of use on a band then a string of items bn 5n terminated by U/S is printed. If the whole band is out of use then the single item bn U/S is printed.
The final line printed is
TOTAL STORE AVAILABLE n BLOCKS where n is the decimal total of all usable drum sectors and core store pages. <u>N.B.</u> This does not mean that the main programmer can use up to n blocks since no adjustment is made to take account of the space used by the Supervisor.

12.9 continued

XR 60	:	
XR 61	:	
XR 62	:	all print 'SUPERVISOR PARAMETERS ACCEPTED'
XR 63	:	For XR61, it the specified drum area includes a band reserved by main program, then REMOVE DRUM STORAGE dn bm RESERVED is printed,
XR 64	:	COMPILER n NOT RECOGNIZED where n is compiler name or COMPILER n AVAILABLE
XR 65	:	COMPILER n NOT RECOGNIZED or COMPILER n DELETED FROM STPRE
XR 66	:	EXTRACODE PROGRAM WILL BE ACCEPTED ON X where Xis the peripheral number

12.10 INFORMATION FOR OPERATORS.

STARTING THE SYSTEM

To commence operating with the supervisor the following steps should be taken.

1. Mount the ''final supervisor'' tape and the relevant system tapes, if any are to be used, on working decks. System tapes are : system input tape, system output tape, system dump tape and a combined system input and output tape. Notice will be given beforehand what system tapes are to be used, normally it will be the one combined input and output tape.

2. Do a reset then engage the final supervisor deck and all output peripherals.

3. Set handkeys, 1,2, 4 and 5 them press engineers interrupt and the single button.

4. When the tape begins to move clear the handkeys then press handkey seven followed by handkey zero. This should cause the supervisor to be read read from magnetic tape into main store and the idling loop to be entered. The engineer in charge should be consulted if any difficulty is encountered.

5. Engage decks on which system tapes are mounted. This will cause the teleprinter to print out what system tapes are mounted. Once this is checked to be correct the machine is ready to receive programs. Before starting jobs however any necessary operator requests, removing faulty decks or linking certain peripherals etc., should be read in.

RESTARTS

When a breakdown occurs the above procedure should be repeated except that in step 4 only handkey zero should be pressed. This recovers the jobs which have not yet been executed and any output not yet completed. Jobs whose input is incomplete when the failure occurs have to be read in again. Handkey seven has the effect of losing all previous jobs and their output if the computer is operating without any system tapes a restart should never be attempted ie handkey seven in step 4 should always be set.

JOBS REQUIRING MAGNETIC TAPES

The supervisor controls the assembly of magnetic tapes for jobs and prints out on the teleprinter in the tape room instructions regarding the mounting and dismounting of tapes on decks. Once the tape has been mounted the deck should be engaged and write permit allowed if the supervisor indicated a write ring was needed on this tape. The supervisor then checks that the correct tape has been mounted and if not indicates this on the teleprinter, in this case the wrong tape should be dismounted and the correct one mounted. It is possible for a jobs magnetic tapes to be premounted and engaged without any directives from the supervisor. The supervisor finds on job assembly that the tapes are already mounted and the job proceeds directly to the execution phase. Normally however, it is better to await instructions from the supervisor.

TAPE ADDRESSING

The tape addressing parameters should be read in and the tape to be addressed mounted on deck 7. The deck should then be modified (see engineers if necessary) and the deck engaged with write permit allowed. All information regarding tape addressing is given on the teleprinter in the tape room. When one tape has finished addressing the procedure can be repeated for any further tapes. When tape addressing is complete it is most important that the deck be unmodified.

12.10 continued

The tape addressing parameter tape reads

JOB
TAPE ADDRESSING
COMPILER TAD
A N
***Z

N is the number of blocks to be addressed and will usually be 5000.

TELEPRINTER INFORMATION

All information printed on the teleprinter should be noted by the operators and appropriate action taken, details of this information is given elsewhere in the handbook. Information relating to machine faults should be passed on to the engineer in charge.