

358360.LST

DECmate II ROM Listing



P?S

Name: DECmate II ROM Listing [from 358360.PAL]
Document: 358360.LST
File[s]: 358360.PDF
P?S Index: O>R,CDBI@sd0037HH0037SNL0037Khrshmf
Security Level: 2
Location: NY
Date: 26-Jun-2018
Medium: PDF

1	1	/	DECMATE II ROM CONTENTS	1
2	2			2
3	3	/	DECODED AND DISASSEMBLED BY CHARLES J. LASNER.	3
4	4			4
5	5	/	LAST EDIT: 02-DEC-1991 02:00:00 CJL	5
6	6			6
7	7	/	MAY BE ASSEMBLED WITH '/J' (PAL8 '/F') SWITCH SET.	7
8	8			8
9	9	/	THIS IS THE CODE USED IN THE DECMATE II PRIMARY CONTROL ROM. IT RESIDES IN	9
10	10	/	THREE 2716 PACKAGES KNOWN AS E113, E114, E115. THE THREE ROMS ARE ENCODED	10
11	11	/	INTO A 12-BIT IMAGE OF AN ENTIRE FIELD IN THE PARTICULAR ORGANIZATION	11
12	12	/	ILLUSTRATED BELOW. FOR ALL ROM CONTENTS: 0.X=BIT X IN THE SPACE 0000-3777,	12
13	13	/	4.X=BIT X IN THE SPACE 4000-7777.	13
14	14			14
15	15	/	E113: 0 1 2 3 4 5 6 7	15
16	16			16
17	17	/	BIT: 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7	17
18	18			18
19	19	/	E114: 0 1 2 3 4 5 6 7	19
20	20			20
21	21	/	BIT: 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7	21
22	22			22
23	23	/	E115: 0 1 2 3 4 5 6 7	23
24	24			24
25	25	/	BIT: 0.8 4.8 0.9 4.9 0.10 4.10 0.11 4.11	25
26	26			26
27	27	/	THIS IS THE ASSEMBLY OF THE RELEASED ROM SET KNOWN AS 358E2, 359E2, 360E2.	27
28	28			28
29	29	/	WHEN THE DECMATE II IS FIRST POWERED UP, ROM CONTROL IS ENABLED FOR ALL DIRECT	29
30	30	/	(IF) AND INDIRECT (DF) REFERENCES TO CP MEMORY, THUS THE CPU STARTS AT	30
31	31	/	ROM-BASED ADDRESS CP07777. THE ROM CODE MOVES A LOADING PROGRAM TO PAGE ZERO	31
32	32	/	OF THE CORRESPONDING RAM IN FIELD ZERO OF CP MEMORY, WHICH IS THEN STARTED.	32
33	33	/	TO ACCOMPLISH THE INITIAL MOVE OF THE LOADER, THE DF CONTROL IS CHANGED TO RAM	33
34	34	/	WHILE RETAINING IF CONTROL IN THE ROM.	34
35	35			35
36	36	/	THE LOADER MOVES THE BULK OF THE ROM CODE TO CP FIELD SEVEN AND THEN STARTS IT	36
37	37	/	THERE. NO CODE IS MOVED TO ANY ADDRESS LOWER THAN 0100, SINCE THE CODE IMAGE	37
38	38	/	IS STORED OFFSET BY 100. THIS MEANS THAT ROM LOCATION 0000 WINDS UP IN FIELD	38
39	39	/	7 LOCATION 0100 AND SO ON. DURING THE LOAD OPERATION, THE IF CONTROL IS	39
40	40	/	ALWAYS SET TO RAM BECAUSE AT THIS POINT THE LOADER IS RUNNING. THE DF IS	40
41	41	/	SWITCHED BETWEEN ROM AND RAM AS NECESSARY TO ACCOMPLISH THE LOAD OPERATION.	41
42	42	/	AFTER LOADING IS COMPLETE, THE ROM IS TOTALLY DISABLED.	42
43				43
44				44
45				45
46				46
47				47
48				48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1	44	/	DEFINITIONS.				1
2	45						2
3	46	/	OPERATE INSTRUCTIONS.				3
4	47						4
5	48	7014	R3L=	7014	/ROTATE AC LEFT THREE WITHOUT AFFECTING LINK		5
6	49						6
7	50	/	COMBINED OPERATE INSTRUCTIONS.				7
8	51						8
9	52	7201	NL0001=	CLA IAC	/LOAD AC WITH 0001		9
10	53	7326	NL0002=	CLA CLL CML RTL	/LOAD AC WITH 0002		10
11	54	7305	NL002A=	CLA CLL IAC RAL	/LOAD AC WITH 0002 ON 8/I OR NEWER		11
12	55	7325	NL0003=	CLA CLL CML IAC RAL	/LOAD AC WITH 0003		12
13	56	7307	NL0004=	CLA CLL IAC RTL	/LOAD AC WITH 0004		13
14	57	7327	NL0006=	CLA CLL CML IAC RTL	/LOAD AC WITH 0006		14
15	58	7215	NL0010=	CLA IAC R3L	/LOAD AC WITH 0010		15
16	59	7203	NL0100=	CLA IAC BSW	/LOAD AC WITH 0100		16
17	60	7332	NL2000=	CLA CLL CML RTR	/LOAD AC WITH 2000		17
18	61	7350	NL3777=	CLA CLL CMA RAR	/LOAD AC WITH 3777		18
19	62	7330	NL4000=	CLA CLL CML RAR	/LOAD AC WITH 4000		19
20	63	7352	NL5777=	CLA CLL CMA RTR	/LOAD AC WITH 5777		20
21	64	7333	NL6000=	CLA CLL CML IAC RTR	/LOAD AC WITH 6000		21
22	65	7346	NL7775=	CLA CLL CMA RTL	/LOAD AC WITH 7775		22
23	66	7344	NL7776=	CLA CLL CMA RAL	/LOAD AC WITH 7776		23
24	67	7240	NL7777=	CLA CMA	/LOAD AC WITH 7777		24
25	68						25
26	69	/	PROCESSOR I/O INSTRUCTIONS.				26
27	70						27
28	71	6266	CPD=	6266	/FORCE DF TO REFER TO MAIN MEMORY		28
29	72	6256	GCF=	6256	/GET CURRENT FIELDS		29
30	73	6004	PEX=	6004	/EXIT FROM CP TO MAIN MEMORY		30
31	74	6003	PG0=	6003	/RESET HLTFLG FLIP-FLOP		31
32	75	6236	PRQ3=	6236	/TYPE 3 PANEL REQUEST		32
33	76	6000	PRS=	6000	/READ PANEL STATUS		33
34	77	6207	RSP1=	6207	/READ FIRST STACK POINTER		34
35	78	6227	RSP2=	6227	/READ SECOND STACK POINTER		35
36	79	6276	SPD=	6276	/FORCE DF TO REFER TO CP MEMORY		36
37	80	6246	WSR=	6246	/WRITE SWITCH REGISTER (ROM CONTROL), CLEAR AC		37
38	81						38
39	82	/	INTERNAL I/O INSTRUCTIONS.				39
40	83						40
41	84	0003	KEY=	03	/KEYBOARD DEVICE CODE		41
42	85	0004	TTY=	04	/SCREEN DEVICE CODE		42
43	86						43
44	87	/	VIDEO INTERRUPT INSTRUCTIONS.				44
45	88						45
46	89	0006	VIDINT=	06	/VIDEO INTERRUPT DEVICE CODE		46
47	90						47
48	91	6060	VFL=	VIDINT^10+6000	/SET VIDEO INTERRUPT FLAG		48
49	92	6061	VSF=	VIDINT^10+6001	/SKIP ON, CLEAR VIDEO INTERRUPT FLAG		49
50	93	6062	VCL=	VIDINT^10+6002	/NOP (CLEAR THE AC?)		50
51	94	6063	VNOP1=	VIDINT^10+6003	/(NOP?)		51
52	95	6064	VDUMM1=	VIDINT^10+6004	/(NOP?)		52
53	96	6065	VIE=	VIDINT^10+6005	/WRITE INTERRUPT ENABLE PER AC[11]		53
54	97	6066	VDUMM2=	VIDINT^10+6006	/(NOP?)		54
55	98	6067	VDUMM3=	VIDINT^10+6007	/(NOP?)		55
56							56
57							57
58							58

1	100	/	VIDEO CONTROLLER INSTRUCTION.	1
2	101			2
3	102	0012	VIDDEV= 12 /VIDEO CONTROLLER DEVICE CODE	3
4	103			4
5	104	6120	APTFL= VIDDEV^10+6000 /SET APT INTERRUPT FLAG	5
6	105	6121	APTSKP= VIDDEV^10+6001 /SKIP ON, CLEAR APT INTERRUPT FLAG	6
7	106	6122	LSCREG= VIDDEV^10+6002 /LOAD VIDEO REGISTER SELECT	7
8	107	6123	CGLOAD= VIDDEV^10+6003 /LOAD CHARACTER GENERATOR ADDRESS/DATA	8
9	108	6124	VLOAD= VIDDEV^10+6004 /LOAD SELECTED REGISTER FROM AC[4-11]	9
10	109	6125	APTIE= VIDDEV^10+6005 /WRITE INTERRUPT ENABLE PER AC[11]	10
11	110	6126	VIDCON= VIDDEV^10+6006 /AC TO VIDEO CONTROL	11
12	111	6127	VREAD= VIDDEV^10+6007 /READ SELECTED REGISTER	12
13	112			13
14	113	/	PRINTER INTERFACE INSTRUCTIONS.	14
15	114			15
16	115	0032	PRICON= 32 /PRINTER INPUT DEVICE CODE	16
17	116			17
18	117	6320	PRIFL= PRICON^10+6000 /SET PRINTER INPUT FLAG	18
19	118	6321	PRISKP= PRICON^10+6001 /SKIP ON, CLEAR PRINTER INPUT FLAG	19
20	119	6322	PRICLR= PRICON^10+6002 /CLEAR THE AC	20
21	120	6323	PRINO1= PRICON^10+6003 /NOP	21
22	121	6324	PRIRS= PRICON^10+6004 /OR INPUT BUFFER WITH AC	22
23	122	6325	PRIIE= PRICON^10+6005 /WRITE INTERRUPT ENABLE PER AC[11]	23
24	123	6326	PRIRB= PRICON^10+6006 /LOAD INPUT BUFFER INTO AC	24
25	124	6327	PRINO2= PRICON^10+6007 /NOP	25
26	125			26
27	126	0033	PROCON= 33 /PRINTER OUTPUT DEVICE CODE	27
28	127			28
29	128	6330	PROFL= PROCON^10+6000 /SET PRINTER OUTPUT FLAG	29
30	129	6331	PROSKP= PROCON^10+6001 /SKIP ON, CLEAR PRINTER OUTPUT FLAG	30
31	130	6332	PRONO1= PROCON^10+6002 /NOP	31
32	131	6333	PRSB= PROCON^10+6003 /LOAD BAUD RATE PER AC[8-11]	32
33	132	6334	PROPC= PROCON^10+6004 /OUTPUT CHARACTER FROM AC[4-11]	33
34	133	6335	PROIE= PROCON^10+6005 /WRITE INTERRUPT ENABLE PER AC[11]	34
35	134	6336	PROLS= PROCON^10+6006 /OUTPUT CHARACTER FROM AC[4-11], CLEAR AC	35
36	135	6337	PRONO2= PROCON^10+6007 /NOP	36
37	136			37
38	137	/	RD51D I/O INSTRUCTIONS.	38
39	138			39
40	139	6700	RDNOP= 6700 /RESERVED INSTRUCTION	40
41	140	6701	RDSR= 6701 /SKIP ON, CLEAR DATA REQUEST FLAG	41
42	141	6702	RDSC= 6702 /SEND COMMAND PER AC	42
43	142	6703	RDSD= 6703 /SKIP ON, CLEAR DONE FLAG	43
44	143	6704	RDTD= 6704 /TRANSFER DATA TO/FROM AC	44
45	144	6705	RDWE= 6705 /WRITE INTERRUPT ENABLE PER AC[11]	45
46	145	6706	RDSE= 6706 /SKIP ON, CLEAR ERROR FLAG	46
47	146	6707	RDTEST= 6707 /RESERVED TEST INSTRUCTION THAT CLEARS THE AC	47
48	147		/ONLY IF THE RD51D IS INSTALLED.	48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1	149	/	COMMUNICATIONS PORT DEFINITIONS.		1
2	150				2
3	151	0030	PORTIN= 30	/COMMUNICATIONS PORT INPUT DEVICE CODE	3
4	152				4
5	153	6300	IFL= PORTIN^10+6000	/SET COMMUNICATIONS PORT INPUT/OUTPUT FLAG	5
6	154	6301	ISF= PORTIN^10+6001	/SKIP ON, CLEAR PORT INPUT/OUTPUT FLAG	6
7	155	6302	ICF= PORTIN^10+6002	/NOP (CLEAR THE AC?)	7
8	156	6303	INOP1= PORTIN^10+6003	/(NOP?)	8
9	157	6304	IRS= PORTIN^10+6004	/READ COMMUNICATIONS PORT RECEIVE BUFFER	9
10	158	6305	IIE= PORTIN^10+6005	/PORT I/O INTERRUPT ENABLE PER AC[11]	10
11	159	6306	IRB= PORTIN^10+6006	/READ COMMUNICATIONS PORT RECEIVE BUFFER	11
12	160	6307	INOP2= PORTIN^10+6007	/(NOP?)	12
13	161				13
14	162	0031	PORTOUT=31	/COMMUNICATIONS PORT OUTPUT DEVICE CODE	14
15	163				15
16	164	6310	DUMBFL= PORTOUT^10+6000	/SET COMMUNICATIONS PORT DUMMY FLAG	16
17	165	6311	DUMBSF= PORTOUT^10+6001	/SKIP ON, CLEAR COMMUNICATIONS PORT DUMMY FLAG	17
18	166	6312	DUMBCF= PORTOUT^10+6002	/NOP (CLEAR THE AC?)	18
19	167	6313	ONOP1= PORTOUT^10+6003	/(NOP?)	19
20	168	6314	OPC= PORTOUT^10+6004	/LOAD COMMUNICATIONS PORT TRANSMIT BUFFER	20
21	169	6315	DUMBIE= PORTOUT^10+6005	/PORT DUMMY INTERRUPT ENABLE PER AC[11]	21
22	170	6316	OLS= PORTOUT^10+6006	/LOAD COMMUNICATIONS PORT TRANSMIT BUFFER	22
23	171	6317	ONOP2= PORTOUT^10+6007	/(NOP?)	23
24	172				24
25	173	0036	PORTCON=36	/COMMUNICATIONS PORT CONTROL DEVICE CODE	25
26	174				26
27	175	6360	MFL= PORTCON^10+6000	/SET MODEM CHANGE FLAG	27
28	176	6361	MSF= PORTCON^10+6001	/SKIP ON, CLEAR MODEM CHANGE FLAG	28
29	177	6362	MLC= PORTCON^10+6002	/LOAD MODEM CONTROL REGISTER	29
30	178	6363	MSB= PORTCON^10+6003	/LOAD BAUD RATE REGISTER	30
31	179	6364	MRS= PORTCON^10+6004	/READ MODEM STATUS REGISTER	31
32	180	6365	MIE= PORTCON^10+6005	/MODEM CHANGE INTERRUPT ENABLE PER AC[11]	32
33	181	6366	MPSCC= PORTCON^10+6006	/ACCESS MULTIPROTOCOL SERIAL CONTROLLER	33
34	182	6367	MPRESET=PORTCON^10+6007	/RESET MULTIPROTOCOL SERIAL CONTROLLER	34
35					35
36					36
37					37
38					38
39					39
40					40
41					41
42					42
43					43
44					44
45					45
46					46
47					47
48					48
49					49
50					50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	184	/	TTY: TRAP DEFINITIONS.			1
2	185					2
3	186	0007	TTYCON= 07	/DEVICE CODE FOR TTY: TRAP		3
4	187					4
5	188	6070	TTYFL= TTYCON^10+6000	/SET TTY: TRAP FLAG		5
6	189	6071	TTYSF= TTYCON^10+6001	/SKIP ON, CLEAR TTY: TRAP FLAG		6
7	190	6072	TTYCLR= TTYCON^10+6002	/CLEAR THE AC		7
8	191	6073	TTNOP1= TTYCON^10+6003	/NOP		8
9	192	6074	TTNOP2= TTYCON^10+6004	/NOP		9
10	193	6075	TTYIE= TTYCON^10+6005	/WRITE INTERRUPT ENABLE PER AC[11]		10
11	194	6076	TTNOP3= TTYCON^10+6006	/NOP		11
12	195	6077	TTNOP4= TTYCON^10+6007	/NOP		12
13	196					13
14	197	/	CLOCK DEFINITIONS.			14
15	198					15
16	199	0013	CLKCON= 13	/REAL-TIME CLOCK DEVICE CODE		16
17	200					17
18	201	6130	CLFL= CLKCON^10+6000	/SET CLOCK FLAG		18
19	202	6131	CLSK= CLKCON^10+6001	/SKIP ON, CLEAR CLOCK FLAG		19
20	203	6132	CLNOP1= CLKCON^10+6002	/NOP		20
21	204	6133	CLNOP2= CLKCON^10+6003	/NOP		21
22	205	6134	CLNOP3= CLKCON^10+6004	/NOP		22
23	206	6135	CLIE= CLKCON^10+6005	/WRITE INTERRUPT ENABLE PER AC[11]		23
24	207	6136	CLNOP4= CLKCON^10+6006	/NOP		24
25	208	6137	CLNOP5= CLKCON^10+6007	/NOP		25
26	209					26
27	210	/	KEYBOARD DEFINITIONS.			27
28	211					28
29	212	0011	KBICON= 11	/KEYBOARD INPUT DEVICE CODE		29
30	213					30
31	214	6110	KBIFL= KBICON^10+6000	/SET KEYBOARD INPUT FLAG		31
32	215	6111	KBISF= KBICON^10+6001	/SKIP ON, CLEAR KEYBOARD INPUT FLAG		32
33	216	6112	KBICLR= KBICON^10+6002	/CLEAR THE AC		33
34	217	6113	KINOP1= KBICON^10+6003	/NOP		34
35	218	6114	KBIRS= KBICON^10+6004	/OR AC WITH KEYBOARD INPUT DATA		35
36	219	6115	KBIIE= KBICON^10+6005	/WRITE INTERRUPT ENABLE PER AC[11]		36
37	220	6116	KBIRB= KBICON^10+6006	/READ KEYBOARD INPUT DATA INTO AC		37
38	221	6117	KINOP2= KBICON^10+6007	/NOP		38
39	222					39
40	223	0005	KBOCON= 05	/KEYBOARD OUTPUT DEVICE CODE		40
41	224					41
42	225	6050	KBOFL= KBOCON^10+6000	/SET KEYBOARD OUTPUT FLAG		42
43	226	6051	KBOSF= KBOCON^10+6001	/SKIP ON, CLEAR KEYBOARD OUTPUT FLAG		43
44	227	6052	KONOP1= KBOCON^10+6002	/NOP		44
45	228	6053	KONOP2= KBOCON^10+6003	/NOP		45
46	229	6054	KBOPC= KBOCON^10+6004	/OUTPUT AC TO KEYBOARD		46
47	230	6055	KBOIE= KBOCON^10+6005	/WRITE INTERRUPT ENABLE PER AC[11]		47
48	231	6056	KBOLS= KBOCON^10+6006	/OUTPUT AC TO KEYBOARD, CLEAR AC		48
49	232	6057	KONOP3= KBOCON^10+6007	/NOP		49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	234		/	RX50/RX01/RX02 DEFINITIONS.		1
2	235					2
3	236	0075		RXCON= 75	/RX50 DEVICE CODE	3
4	237					4
5	238	6750		SEL= RXCON^10+6000	/SELECT DISKETTE PAIR PER AC[0] AND AC[11]	5
6	239	6751		LCD= RXCON^10+6001	/LOAD COMMAND REGISTER, CLEAR AC	6
7	240	6752		XDR= RXCON^10+6002	/TRANSFER DATA	7
8	241	6753		STR= RXCON^10+6003	/SKIP ON TRANSFER FLAG, CLEAR TRANSFER FLAG	8
9	242	6754		SER= RXCON^10+6004	/SKIP ON ERROR FLAG, CLEAR ERROR FLAG	9
10	243	6755		SDN= RXCON^10+6005	/SKIP ON DONE FLAG, CLEAR DONE FLAG	10
11	244	6756		INTR= RXCON^10+6006	/WRITE INTERRUPT ENABLE PER AC[11]	11
12	245	6757		RXINIT= RXCON^10+6007	/INITIALIZE CONTROLLER AND DRIVES	12
13	246					13
14	247		/	MRI DEFINITIONS.		14
15	248					15
16	249	5600		JMPIC= JMP I .	/CURRENT PAGE JMP I	16
17	250					17
18	251		/	OTHER DEFINITIONS.		18
19	252					19
20	253	0000		A= 0	/EVEN HALF OF COMMUNICATIONS CHIP	20
21	254	0014		APUCON= 14	/APU/XPU DEVICE CODE	21
22	255	0001		B= 1	/ODD HALF OF COMMUNICATIONS CHIP	22
23	256	0400		DTIME= 400	/DISK WAIT TIME-OUT FACTOR	23
24	257	3660		ROWADR= 3660	/ROWTABLE ADDRESS IN FIELD 1	24
25	258	5350		T0S3BUF=5350	/TRACK 0, SECTOR 3 BUFFER IN FIELD 1	25
26						26
27						27
28						28
29						29
30						30
31						31
32						32
33						33
34						34
35						35
36						36
37						37
38						38
39						39
40						40
41						41
42						42
43						43
44						44
45						45
46						46
47						47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	260	0007	FIELD	7		/MOST OF THE CODE EXECUTES IN FIELD 7	1
2	261						2
3	262	*7600	PAGE	37		/STARTUP PAGE	3
4	263						4
5	264	/			THE ROM STARTS UP HERE (JUMPED TO FROM THE END OF THE PAGE). SINCE WE ARE		5
6	265	/			STARTING FROM POWERUP, BOTH CONTROL BITS ARE CLEARED. THE ROM IS REFERENCED		6
7	266	/			WHEN MEMORY IS READ VIA BOTH THE INSTRUCTION AND DATA FIELDS INITIALLY.		7
8	267	/			FURTHER, FORCE-ZERO MODE IS IN EFFECT WHICH MUST BE CLEARED TO USE EXTENDED		8
9	268	/			MEMORY.		9
10	269						10
11	270	7600	ROMGO=	.		/ROM STARTS HERE	11
12	271						12
13	272	077600	6202	ROMGO,	CIF 00	/SET OUR OWN FIELD; START CLEARING FZ MODE	13
14	273	077601	5202	JMP	+.1	/THIS JUST LOADS OUR FIELD, BUT CLEARS FZ MODE	14
15	274	077602	7330	NL4000		/SETUP PATTERN FOR RAM DATA FIELD	15
16	275	077603	6246	WSR		/ROM IS ONLY IF NOW, DF IS RAM	16
17	276						17
18	277	/			APPARENTLY THE RAM MAY NOT YET BE WORKING, SO VARIOUS TESTS ARE DONE UNTIL IT		18
19	278	/			RESPONDS CORRECTLY.		19
20	279						20
21	280	077604	7320	MEMTST,	CLA STL	/CLEAN UP AND SET A STOP BIT	21
22	281	077605	2745	WATLUP,	ISZ I ATABLE+0/(L4001)	/BUMP UP TEST LOCATION	22
23	282	077606	7000	NOP		/IN CASE IT SKIPS	23
24	283	077607	7004	RAL		/MOVE PATTERN OVER	24
25	284	077610	7500	SMA		/DONE ENOUGH?	25
26	285	077611	5205	JMP	WATLUP	/NO, KEEP GOING	26
27	286	077612	7200	CLA		/CLEAN UP	27
28	287	077613	6276	SPD		/FORCE INDIRECT REFERENCES TO CP MEMORY	28
29	288						29
30	289	/			TRY TO STORE ALL ZEROES INTO THE RAM TEST LOCATION.		30
31	290						31
32	291	077614	3745	DCA I	ATABLE+0/(L4001)	/ATTEMPT TO STORE CLEAR PATTERN	32
33	292	077615	1745	TAD I	ATABLE+0/(L4001)	/GET IT BACK	33
34	293	077616	7640	SZA CLA		/SKIP IF ACTUALLY CLEAR	34
35	294	077617	5204	JMP	MEMTST	/GO BACK IF IT FLUNKS	35
36	295						36
37	296	/			TRY TO STORE ALTERNATING ZEROES AND ONES INTO THE RAM TEST LOCATION.		37
38	297						38
39	298	077620	1317	TAD	L2525/(2525)	/GET A DATA PATTERN	39
40	299	077621	3745	DCA I	ATABLE+0/(L4001)	/ATTEMPT TO STORE IT	40
41	300	077622	1745	TAD I	ATABLE+0/(L4001)	/GET IT BACK	41
42	301	077623	1320	TAD	L5252/(5252)	/ADD ON NEGATED PATTERN	42
43	302	077624	7001	IAC		/ADD ONE TO FINISH THE COMPARISON	43
44	303	077625	7640	SZA CLA		/SKIP IF IT MATCHES	44
45	304	077626	5204	JMP	MEMTST	/GO DO IT AGAIN IF IT FLUNKS	45
46	305						46
47	306	/			TRY TO STORE ALTERNATING ONES AND ZEROES INTO THE RAM TEST LOCATION.		47
48	307						48
49	308	077627	1320	TAD	L5252/(5252)	/GET A DATA PATTERN	49
50	309	077630	3745	DCA I	ATABLE+0/(L4001)	/ATTEMPT TO STORE IT	50
51	310	077631	1745	TAD I	ATABLE+0/(L4001)	/GET IT BACK	51
52	311	077632	1317	TAD	L2525/(2525)	/ADD ON NEGATED PATTERN	52
53	312	077633	7001	IAC		/ADD ONE TO FINISH THE COMPARISON	53
54	313	077634	7640	SZA CLA		/SKIP IF IT MATCHES	54
55	314	077635	5204	JMP	MEMTST	/GO DO IT AGAIN IF IT FLUNKS	55
56							56
57							57
58							58

1	316		/	TRY TO STORE ALL ONES INTO THE RAM TEST LOCATION.		1
2	317					2
3	318	077636	7240	NL7777	/SET DATA PATTERN	3
4	319	077637	3745	DCA I ATABLE+0/(L4001)	/ATTEMPT TO STORE IT	4
5	320	077640	7301	NL0001 CLL	/SET NEGATED PATTERN FOR COMPARISON	5
6	321	077641	1745	TAD I ATABLE+0/(L4001)	/ADD ON TEST VALUE	6
7	322	077642	7640	SZA CLA	/SKIP IF IT MATCHES	7
8	323	077643	5204	JMP MEMTST	/GO DO IT AGAIN IF IT FLUNKS	8
9	324					9
10	325		/	MOVE THE IMAGE OF THE LOADING PROGRAM TO PAGE ZERO OF RAM.		10
11	326					11
12	327	077644	1321	TAD LPROGM+0/(4001)	/GET A WORD	12
13	328	077645	3745	DCA I ATABLE+0/(L4001)	/PUT A WORD	13
14	329	077646	1322	TAD LPROGM+1/(CODEXC-1)	/GET A WORD	14
15	330	077647	3746	DCA I ATABLE+1/(XR6)	/PUT A WORD	15
16	331	077650	1323	TAD LPROGM+2/(-ROMGO+1)	/GET A WORD	16
17	332	077651	3747	DCA I ATABLE+2/(MOVCNT)	/PUT A WORD	17
18	333	077652	1324	TAD LPROGM+3/(NL7777 CLL)	/GET A WORD	18
19	334	077653	3750	DCA I ATABLE+3/(LOADIT+0)	/PUT A WORD	19
20	335	077654	1325	TAD LPROGM+4/(DCA XR1)	/GET A WORD	20
21	336	077655	3751	DCA I ATABLE+4/(LOADIT+1)	/PUT A WORD	21
22	337	077656	1326	TAD LPROGM+5/(CDF 00)	/GET A WORD	22
23	338	077657	3752	DCA I ATABLE+5/(LOADLP+0)	/PUT A WORD	23
24	339	077660	1327	TAD LPROGM+6/(NL0001)	/GET A WORD	24
25	340	077661	3753	DCA I ATABLE+6/(LOADLP+1)	/PUT A WORD	25
26	341	077662	1330	TAD LPROGM+7/(WSR)	/GET A WORD	26
27	342	077663	3754	DCA I ATABLE+7/(LOADLP+2)	/PUT A WORD	27
28	343	077664	1331	TAD LPROGM+10/(TAD I XR1)	/GET A WORD	28
29	344	077665	3755	DCA I ATABLE+10/(LOADLP+3)	/PUT A WORD	29
30	345	077666	1332	TAD LPROGM+11/(MQL)	/GET A WORD	30
31	346	077667	3756	DCA I ATABLE+11/(LOADLP+4)	/PUT A WORD	31
32	347	077670	1333	TAD LPROGM+12/(TAD L4001)	/GET A WORD	32
33	348	077671	3757	DCA I ATABLE+12/(LOADLP+5)	/PUT A WORD	33
34	349	077672	1334	TAD LPROGM+13/(WSR)	/GET A WORD	34
35	350	077673	3760	DCA I ATABLE+13/(LOADLP+6)	/PUT A WORD	35
36	351	077674	1335	TAD LPROGM+14/(CLA MQA)	/GET A WORD	36
37	352	077675	3761	DCA I ATABLE+14/(LOADLP+7)	/PUT A WORD	37
38	353	077676	1336	TAD LPROGM+15/(CDF 70)	/GET A WORD	38
39	354	077677	3762	DCA I ATABLE+15/(LOADLP+10)	/PUT A WORD	39
40	355	077700	1337	TAD LPROGM+16/(DCA I XR6)	/GET A WORD	40
41	356	077701	3763	DCA I ATABLE+16/(LOADLP+11)	/PUT A WORD	41
42	357	077702	1340	TAD LPROGM+17/(ISZ MOVCNT)	/GET A WORD	42
43	358	077703	3764	DCA I ATABLE+17/(LOADLP+12)	/PUT A WORD	43
44	359	077704	1341	TAD LPROGM+20/(JMP LOADLP)	/GET A WORD	44
45	360	077705	3765	DCA I ATABLE+20/(LOADLP+13)	/PUT A WORD	45
46	361	077706	1342	TAD LPROGM+21/(CIF 70)	/GET A WORD	46
47	362	077707	3766	DCA I ATABLE+21/(LOADLP+14)	/PUT A WORD	47
48	363	077710	1343	TAD LPROGM+22/(JMP I .+1)	/GET A WORD	48
49	364	077711	3767	DCA I ATABLE+22/(LOADLP+15)	/PUT A WORD	49
50	365	077712	1344	TAD LPROGM+23/(CODESTART)	/GET A WORD	50
51	366	077713	3770	DCA I ATABLE+23/(LOADLP+16)	/PUT A WORD	51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	414	077745	0015	ATABLE, LOADER+00	/TABLE OF LOADING ADDRESSES FOR MOVED LOADER	1
2	415	077746	0016	LOADER+01	/	2
3	416	077747	0017	LOADER+02	/	3
4	417	077750	0020	LOADER+03	/	4
5	418	077751	0021	LOADER+04	/	5
6	419	077752	0022	LOADER+05	/	6
7	420	077753	0023	LOADER+06	/	7
8	421	077754	0024	LOADER+07	/	8
9	422	077755	0025	LOADER+10	/	9
10	423	077756	0026	LOADER+11	/	10
11	424	077757	0027	LOADER+12	/	11
12	425	077760	0030	LOADER+13	/	12
13	426	077761	0031	LOADER+14	/	13
14	427	077762	0032	LOADER+15	/	14
15	428	077763	0033	LOADER+16	/	15
16	429	077764	0034	LOADER+17	/	16
17	430	077765	0035	LOADER+20	/	17
18	431	077766	0036	LOADER+21	/	18
19	432	077767	0037	LOADER+22	/	19
20	433	077770	0040	LOADER+23	/LAST ADDRESS OF MOVED LOADER CODE	20
21	434					21
22	435	077771	0000	ZBLOCK 7776-.	/EMPTY SPACE	22
23	436					23
24	437			/	WHEN THE CODE IS IN RAM, THESE LOCATIONS IN CP FIELD 0 ARE USED AS THE START	24
25	438			/	OF THE CP INTERRUPT HANDLER. THE MAIN MEMORY INTERRUPTED PC IS STORED IN CP	25
26	439			/	00000 AND CONTROL IS REGAINED AT CP 07777. INVARIABLY, THE INSTRUCTION USED	26
27	440			/	THERE IS JMP I .-1. THE POINTER ADDRESS IS THEREFORE STORED HERE AT CP 07776.	27
28	441					28
29	442	077776	0000	ROMSP, .-.	/USED LATER AS CP INTERRUPT POINTER	29
30	443					30
31	444			/	THE ROM PROGRAM GAINS CONTROL HERE. ALSO USED LATER FOR CP INTERRUPTS.	31
32	445					32
33	446	077777	5200	ROMST, JMP ROMGO	/CONTINUE THERE	33
34						34
35						35
36						36
37						37
38						38
39						39
40						40
41						41
42						42
43						43
44						44
45						45
46						46
47						47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	448	/	START OF PAGE ZERO WHILE RUNNING IN FIELD 7. THESE LOCATIONS ARE NOT LOADED	1
2	449	/	BY THE ROM'S LOADING PROGRAM BUT ARE USED BY THE MOVED CODE AS TEMPORARIES.	2
3	450			3
4	451	*0000	*0 /START AT THE BEGINNING	4
5	452			5
6	453		NOPUNCH /DON'T GENERATE BINARY	6
7	454			7
8	455	RIMADR,	/APT RIM-LOADER TEMPORARY	8
9	456	INTADR,	/CP-INTERRUPTS PC STORED HERE	9
10	457			10
11	458	/	THE FOLLOWING ARE DEFINITIONS FOR MAIN MEMORY INTERRUPTS IN FIELD 0.	11
12	459			12
13	460	070000 0000	INTADR, .-. /MAIN-MEMORY INTERRUPTS PC STORED HERE	13
14	461	070001* 5402	INT1, JMP I INT2 /WHAT'S PUT HERE IN MAIN MEMORY FIELD 0	14
15	462	070002* 0250	INT2, INTHND /POINTER TO MAIN MEMORY INTERRUPT HANDLER	15
16	463	070003* 0000	TIMEOUT, .-. /TIME-OUT COUNTER FOR LOOPBACK TEST	16
17	464	070004* 0000	CLKTICK, .-. /CLOCK TICK COUNTER FOR LOOPBACK TEST	17
18	465	070005* 0000	CSTATUS, .-. /COUNT PATTERN PROGRESS STATUS	18
19	466	070006* 0000	PATTERN, .-. /COUNT PATTERN TEMPORARY	19
20	467	070007* 0000	TSTATUS, .-. /LOOPBACK CP TEST STATUS	20
21	468			21
22	469	*0010	*10 /GET TO AUTO-INDEX AREA	22
23	470			23
24	471	070010* 0000	XR0, .-. /AUTO-INDEX REGISTER 0	24
25	472	070011* 0000	XR1, .-. /AUTO-INDEX REGISTER 1	25
26	473	070012* 0000	XR2, .-. /AUTO-INDEX REGISTER 2	26
27	474	070013* 0000	XR3, .-. /AUTO-INDEX REGISTER 3	27
28	475	070014* 0000	XR4, .-. /AUTO-INDEX REGISTER 4	28
29	476	070015* 0000	XR5, .-. /AUTO-INDEX REGISTER 5	29
30	477	070016* 0000	XR6, .-. /AUTO-INDEX REGISTER 6	30
31	478	070017* 0000	XR7, .-. /AUTO-INDEX REGISTER 7	31
32	479			32
33	480	*0020	*20 /GET PAST AUTO-INDEX AREA	33
34	481			34
35	482	070020* 0000	CTMP1, .-. /OUTPUT COUNTING TEMPORARY FOR LOOPBACK TEST	35
36	483	070021* 0000	CTMP2, .-. /INPUT COUNTING TEMPORARY FOR LOOPBACK TEST	36
37	484	070022* 0000	CNTLO, .-. /LOW-ORDER COUNTER FOR LOOPBACK TESTING	37
38	485	070023* 0000	CNTHI, .-. /HIGH-ORDER COUNTER FOR LOOPBACK TESTING	38
39	486	070024* 0000	TEMP1, .-. /TEMPORARY	39
40	487	070025* 0000	TEMP2, .-. /TEMPORARY	40
41	488	070026* 0000	TEMP3, .-. /TEMPORARY	41
42	489			42
43	490	070027* 0000	ZBLOCK 1 /EMPTY SPACE	43
44	491			44
45	492	/	THESE SIX ARE PRINTED AS A GROUP.	45
46	493			46
47	494	070030* 0000	TMO, .-. /TEMPORARY	47
48	495	070031* 0000	COMTM1, .-. /COMMUNICATIONS PORT TESTING TEMPORARY	48
49	496	070032* 0000	MTM1, .-. /TEMPORARY	49
50	497	070033* 0000	MTM2, .-. /TEMPORARY	50
51	498	070034* 0000	SP1SAVE, .-. /FIRST STACK POINTER SAVED HERE	51
52	499	070035* 0000	SP2SAVE, .-. /SECOND STACK POINTER SAVED HERE	52
53				53
54				54
55				55
56				56
57				57
58				58

1	501	070036*	0000	MTM5, .-. /TEMPORARY	1
2	502	070037*	0000	MTM6, .-. /TEMPORARY	2
3	503	070040*	0000	MTM7, .-. /TEMPORARY	3
4	504	070041*	0000	TINC, .-. /TEMPORARY	4
5	505	070042*	0000	T0, .-. /TEMPORARY	5
6	506	070043*	0000	T1, .-. /TEMPORARY	6
7	507	070044*	0000	T2, .-. /TESTING TEMPORARY	7
8	508	070045*	0000	T3, .-. /TEMPORARY	8
9	509	070046*	0000	T4, .-. /TEMPORARY	9
10	510	070047*	0000	T5, .-. /TEMPORARY	10
11	511	070050*	0000	T6, .-. /TEMPORARY	11
12	512	070051*	0000	T7, .-. /TEMPORARY	12
13	513	070052*	0000	T8, .-. /TEMPORARY	13
14	514	070053*	0000	RXCMD, .-. /LATEST RX50 COMMAND	14
15	515	070054*	0000	ESTATUS, .-. /CUMULATIVE ERROR STATUS	15
16	516	070055*	0000	LCHAR, .-. /LATEST COMMAND CHARACTER IN SETUP ROUTINE	16
17	517	070056*	0000	ROW, .-. /ROW POSITION FOR CHARACTER ROUTINES	17
18	518	070057*	0000	COLUMN, .-. /COLUMN POSITION FOR CHARACTER ROUTINES	18
19	519	070060*	0000	CCNT, .-. /DELIMITER CHARACTER SEARCH COUNT	19
20	520	070061*	0000	PSTATUS, .-. /STATUS AT POWER-ON	20
21	521				21
22	522		ENPUNCH	/RESTORE BINARY	22
23					23
24					24
25					25
26					26
27					27
28					28
29					29
30					30
31					31
32					32
33					33
34					34
35					35
36					36
37					37
38					38
39					39
40					40
41					41
42					42
43					43
44					44
45					45
46					46
47					47
48					48
49					49
50					50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	524	/	START OF ROM-BASED CODE.			1
2	525					2
3	526	*0000	*0	/START AT THE BEGINNING		3
4	527					4
5	528	0000	ROMADR= .	/WHERE IT IS LOADED WHILE IN ROM		5
6	529					6
7	530	*0100	RELOC 100	/FOOL THE ASSEMBLER		7
8	531					8
9	532	0100	CODEXC= .	/WHERE IT RUNS IN FIELD 7		9
10	533					10
11	534	070100*	3732 3732	/THIS LOCATION IS PROBABLY A VERSION NUMBER		11
12	535					12
13	536	070101*	4204 PCHRPT,CHRPT	/POINTER TO SCREEN CHARACTER OUTPUT ROUTINE		13
14	537	070102*	1400 PPPTABL,PPTABLE	/POINTER TO PROGRAMMING TABLE POINTER		14
15	538	070103*	1451 PROWLOA,ROWLOAD	/POINTER TO ROWTABLE AND REGISTER LOAD ROUTINE		15
16	539	070104*	0007 Z7, 7	/CONSTANT 0007		16
17	540	070105*	0010 Z10, 10	/CONSTANT 0010		17
18	541	070106*	0012 Z12, 12	/CONSTANT 0012		18
19	542	070107*	0017 Z17, 17	/CONSTANT 0017		19
20	543	070110*	0020 Z20, 20	/CONSTANT 0020		20
21	544	070111*	0102 Z102, 102	/CONSTANT 0102		21
22	545	070112*	0200 Z200, 200	/CONSTANT 0200		22
23	546	070113*	0237 Z237, 237	/CONSTANT 0237		23
24	547	070114*	0301 Z0301, 0301	/CONSTANT 0301		24
25	548	070115*	0377 Z377, 377	/CONSTANT 0377		25
26	549	070116*	0400 Z400, 400	/CONSTANT 0400		26
27	550	070117*	2525 Z2525, 2525	/CONSTANT 2525		27
28	551	070120*	3777 Z3777, 3777	/CONSTANT 3777		28
29	552	070121*	5776 JMPIM1, ROMSP&177+JMPIC	/JMP I .-1 CONSTANT FOR RESTART ADDRESS		29
30	553	070122*	6201 ZCDF, CDF 00	/CONSTANT 6201		30
31	554	070123*	6400 PWRIBUF,WRIBUFFER	/POINTER TO WRITE BUFFER		31
32	555	070124*	6500 PAPTR0U,APTROUTINE	/POINTER TO APT ROUTINE		32
33	556					33
34	557	070125*	6777 6777	/CONSTANT 6777 (UNUSED?)		34
35	558					35
36	559	070126*	7766 Z7766, 7766	/CONSTANT 7766		36
37	560	070127*	7760 Z7760, 7760	/CONSTANT 7760		37
38	561	070130*	7741 Z7741, 7741	/CONSTANT 7741		38
39	562	070131*	7400 Z7400, 7400	/CONSTANT 7400		39
40	563	070132*	7400 DTIMOUT,-DTIME	/DISK WAIT TIMEOUT FACTOR		40
41	564	070133*	7775 Z7775, 7775	/CONSTANT 7775		41
42	565		CPIADDR,/ROMSP	/POINTER ADDRESS FOR CP-INTERRUPTS		42
43	566	070134*	7776 Z7776, 7776	/CONSTANT 7776		43
44	567	070135*	7777 PROMST, ROMST	/ROM (RE)START ADDRESS FOR INTERRUPTS, ETC.		44
45	568	070136*	2577 LBLADR, LBLOAD-1	/LOOPBACK TEST CODE ADDRESS (-1)		45
46	569	070137*	7631 LBMCNT, LBTEST-LBTEND	/LENGTH OF LOOPBACK TEST CODE TO BE MOVED		46
47	570	070140*	2777 CPLADR, CPLOAD-1	/CP-INTERRUPT CODE ADDRESS (-1)		47
48	571	070141*	5177 CPMCNT, CPLOAD-CPEND2	/LENGTH OF CP-INTERRUPT CODE TO BE MOVED		48
49	572	070142*	5137 PCOMLOA,COMLOAD	/=> COMMUNICATIONS CHIP REGISTER LOAD ROUTINE		49
50	573	070143*	5115 PCOMREA,COMREAD	/=> COMMUNICATIONS CHIP REGISTER READ ROUTINE		50
51	574	070144*	1513 PCLR23, CLR23	/POINTER TO FIELD 2, 3 CLEAR ROUTINE		51
52	575	070145*	5200 PRXCOMD,RXCOMD	/POINTER TO RXCOMD ROUTINE		52
53	576	070146*	2000 PSSTRIN,SSTRING	/POINTER TO SSTRING ROUTINE		53
54						54
55						55
56						56
57						57
58						58

1	578	/	THE FOLLOWING LOCATION IS UNREFERENCED; PERHAPS THERE IS A CONVENTION ABOUT	1
2	579	/	WHERE TO RESTART THE ROM PROGRAM?	2
3	580			3
4	581	070147* 0214	PRGDONE /POINTER TO PRGDONE ROUTINE (UNUSED?)	4
5	582			5
6	583	070150* 1421	PMEMCOM, MEMCOMP /POINTER TO MEMORY COMPARE ROUTINE	6
7	584	070151* 3672	PKBDIN, KBDIN /POINTER TO KEYBOARD INPUT ROUTINE	7
8	585	070152* 5600	PCALLSU, CALLSUB /POINTER TO TEST (SUBROUTINE) LOCATION	8
9	586	070153* 0205	Z205, 205 /CONSTANT 0205	9
10	587			10
11	588	070154* 0000	ZBLOCK 21 /EMPTY SPACE	11
12	589			12
13	590	/	*****	13
14	591			14
15	592	070175* 3412	SUBR /POINTER TO SUBR	15
16	593	070176* 0000	SUB176, .-. /ENTRY HERE	16
17	594	070177* 5575	JMP I .-2/[SUBR] /GO THERE	17
18				18
19				19
20				20
21				21
22				22
23				23
24				24
25				25
26				26
27				27
28				28
29				29
30				30
31				31
32				32
33				33
34				34
35				35
36				36
37				37
38				38
39				39
40				40
41				41
42				42
43				43
44				44
45				45
46				46
47				47
48				48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1	648	/	FOR SOME PURPOSE, WE NEED TO STORE 0100 INTO CP20301.				1
2	649						2
3	650	070240* 7303	NL0100	CLL	/GET VALUE	3	
4	651	070241* 6221	CDF	20	/GOTO BUFFER FIELD	4	
5	652	070242* 3514	DCA I	Z0301/[0301]	/STORE IN BUFFER	5	
6	653	070243* 6271	CDF	70	/BACK TO OUR FIELD	6	
7	654						7
8	655	/	IF THE PRINTER PORT HAS THE APTEN LINE GROUNDED, WE GO IMMEDIATELY INTO THE				8
9	656	/	AUTOMATIC PRODUCT TEST DOWNLOAD MODE. THIS COULD BE A SECONDARY CALL TO THE				9
10	657	/	DOWNLOAD ROUTINE IF A PREVIOUSLY LOADED ROUTINE EXITED VIA A PANEL REQUEST				10
11	658	/	INSTEAD OF HLT.				11
12	659						12
13	660	070244* 6121	APTSKP		/AUTOMATIC PRODUCT TEST REQUESTED?	13	
14	661	070245* 5251	JMP	MEMTEST	/NO, JUST KEEP GOING	14	
15	662	070246* 4647	JMS I	+.1; DOWNLOAD	/YES, GO DOWNLOAD SOME STUFF.	15	
16	663	070247* 0640				16	
17	664						17
18	665	070250* 0072	L72,	72	/CONSTANT 0072	18	
19	666						19
20	667	/	MEMORY TESTING. ALTERNATING ONES AND ZEROES ARE STORED INTO ALL FIELDS IN				20
21	668	/	MAIN MEMORY AND FIELDS 2-6 IN CP MEMORY. CHECK IF THE PATTERN WORKED AND THEN				21
22	669	/	REVERSE ALL PARAMETERS EACH PASS FOR SEVERAL ITERATIONS.				22
23	670						23
24	671	070251* 6266	MEMTEST,	CPD	/INDIRECT TO MAIN MEMORY FROM NOW ON	24	
25	672	070252* 6271	CDF	70	/RESET TO OUR FIELD FIRST (WHY?)	25	
26	673	070253* 3044	DCA	T2	/CLEAR FIELD COUNTER	26	
27	674						27
28	675	/	EXECUTE CDF TO LATEST TEST FIELD.				28
29	676						29
30	677	070254* 1044	MLODLP,	TAD T2	/GET LATEST FIELD	30	
31	678	070255* 0104	AND	Z7/[7]	/JUST FIELD BITS	31	
32	679	070256* 7014	R3L		/MOVE UP	32	
33	680	070257* 1122	TAD	ZCDF/[CDF]	/MAKE IT A CDF INSTRUCTION	33	
34	681	070260* 3261	DCA	+.1	/STORE IN-LINE	34	
35	682	070261* 7402	HLT+.-.		/WILL BE CDF TEST FIELD INSTRUCTION	35	
36	683	070262* 3043	DCA	T1	/CLEAR ADDRESS POINTER	36	
37	684						37
38	685	/	STORE TEST PATTERN IN LATEST TEST FIELD.				38
39	686						39
40	687	070263* 1117	FILLIT,	TAD Z2525	/GET ALTERNATING TEST PATTERN	40	
41	688	070264* 3443	DCA I	T1	/STASH INTO TEST MEMORY	41	
42	689	070265* 2043	ISZ	T1	/BUMP TO NEXT	42	
43	690	070266* 5263	JMP	FILLIT	/GO BACK FOR MORE	43	
44	691						44
45	692	/	BUMP TO NEXT FIELD AND CHECK WHICH MEMORY BANK WE ARE TESTING.				45
46	693						46
47	694	070267* 2044	ISZ	T2	/BUMP TO NEXT FIELD	47	
48	695	070270* 1044	TAD	T2	/GET LATEST FIELD	48	
49	696	070271* 0105	AND	Z10/[10]	/JUST GROUP BIT	49	
50	697	070272* 7650	SNA	CLA	/SKIP IF FIRST GROUP DONE	50	
51	698	070273* 5254	JMP	MLODLP	/GO BACK AGAIN	51	
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	751	/	BACKUP TO FIELD 6 TO START NEXT TEST PASS.			1
2	752					2
3	753	070340* 7240	NL7777		/BACKUP	3
4	754	070341* 1044	TAD	T2	/ADD ON LATEST FIELD	4
5	755	070342* 3044	BACKEND, DCA	T2	/STORE BACK	5
6	756	070343* 5377	JMP	PASSEND	/CONTINUE THERE	6
7	757					7
8	758	/	COUNTING FIELDS DOWN; CHECK IF WE JUST FINISHED A PASS.			8
9	759					9
10	760	070344* 1044	CDWNTST, TAD	T2	/GET CURRENT FIELD	10
11	761	070345* 7710	SPA CLA		/SKIP IF STILL VALID	11
12	762	070346* 5342	JMP	BACKEND	/JUMP IF NOT	12
13	763					13
14	764	/	CHECK IF WE ARE ABOUT TO START TESTING CP MEMORY.			14
15	765					15
16	766	070347* 1044	CPMTST, TAD	T2	/GET CURRENT FIELD	16
17	767	070350* 0105	AND	Z10/[10]	/CHECK IF INTO CP GROUP	17
18	768	070351* 7650	SNA CLA		/SKIP IF SO	18
19	769	070352* 5323	JMP	MTSTLP	/KEEP GOING	19
20	770	070353* 1041	TAD	TINC	/GET INCREMENTAL FACTOR	20
21	771	070354* 7710	SPA CLA		/SKIP IF STILL POSITIVE	21
22	772	070355* 5366	JMP	NEGTST	/JUMP IF NOT	22
23	773					23
24	774	/	CHECK IF FIRST TIME IN CP MEMORY; SWITCH OVER TO CP INDIRECTS AND BYPASS			24
25	775	/	TESTING OF FIELD 0 AND FIELD 1.			25
26	776					26
27	777	070356* 1044	TAD	T2	/GET LATEST FIELD	27
28	778	070357* 0104	AND	Z7/[7]	/JUST FIELD BITS	28
29	779	070360* 7650	SNA CLA		/SKIP IF NOT AT ZERO	29
30	780	070361* 7305	NL002A		/ELSE START HERE	30
31	781	070362* 1044	TAD	T2	/ADD ON CURRENT FIELD	31
32	782	070363* 3044	DCA	T2	/STORE BACK	32
33	783	070364* 6276	SPD		/PANEL INDIRECTS FROM NOW ON	33
34	784	070365* 5323	JMP	MTSTLP	/CONTINUE THERE	34
35	785					35
36	786	/	IF COUNTING FIELDS DOWN, WE MUST AVOID FIELD 0 AND FIELD 1; IF CP FIELD 2 WAS			36
37	787	/	JUST FINISHED, THEN SWITCH OVER TO MAIN FIELD 7.			37
38	788					38
39	789	070366* 7240	NEGTST, NL7777		/-1	39
40	790	070367* 1044	TAD	T2	/COMPARE TO LATEST FIELD	40
41	791	070370* 0104	AND	Z7/[7]	/JUST FIELD BITS	41
42	792	070371* 7640	SZA CLA		/SKIP IF WE ARE AT FIELD 1	42
43	793	070372* 5323	JMP	MTSTLP	/JUST KEEP GOING	43
44	794	070373* 6266	CPD		/MAIN MEMORY INDIRECTS FROM NOW ON	44
45	795	070374* 1104	TAD	Z7/[7]	/SETUP FOR FIELD 7	45
46	796	070375* 3044	DCA	T2	/STASH IN FIELD COUNTER	46
47	797	070376* 5323	JMP	MTSTLP	/CONTINUE THERE	47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	799	/	COMES HERE AT THE END OF A TEST PASS OVER ALL TESTED FIELDS.				1
2	800					2	
3	801	070377* 1041	PASSEND, TAD	TINC	/GET INCREMENT FACTOR	3	
4	802	070400* 7041	CIA		/INVERT IT	4	
5	803	070401* 3041	DCA	TINC	/STORE BACK	5	
6	804	070402* 1042	TAD	T0	/GET TEST PATTERN COMPARE VALUE	6	
7	805	070403* 7040	CMA		/INVERT THE BITS	7	
8	806	070404* 3042	DCA	T0	/STORE BACK	8	
9	807	070405* 1041	TAD	TINC	/GET INCREMENT AGAIN	9	
10	808	070406* 7710	SPA CLA		/SKIP IF POSITIVE	10	
11	809	070407* 7240	NL7777		/ELSE SET -1	11	
12	810	070410* 3043	DCA	T1	/STORE EITHER WAY	12	
13	811	070411* 2011	ISZ	XR1	/DONE ENOUGH PASSES?	13	
14	812	070412* 5664	JMP I	PMTSTLP/(MTSTLP)	/NO, KEEP GOING	14	
15	813	070413* 4544	JMS I	PCLR23/[CLR23]	/CLEAR FIELDS 2, 3 (AND BITS IN 13705, 13707)	15	
16	814	070414* 6271	CDF	70	/BACK TO OUR FIELD	16	
17	815	070415* 6121	APTSKP		/AUTOMATIC PRODUCT TESTING?	17	
18	816	070416* 7410	SKP		/SKIP IF NOT	18	
19	817	070417* 4524	JMS I	PAPTRoutine/[APTRoutine]	/CALL ROUTINE IF SO	19	
20	818					20	
21	819	/	MOVE CP 72600-72746 TO MAIN 00200-00346. THIS IS THE MAIN MEMORY ROUTINE AND				21
22	820	/	INTERRUPT HANDLER FOR THE PRINTER LOOPBACK TEST.				22
23	821					23	
24	822	070420* 1136	TAD	LBLADR/[LBLOAD-1]	/SETUP THE	24	
25	823	070421* 3010	DCA	XR0	/SOURCE POINTER	25	
26	824	070422* 1137	TAD	LBMCNT/[LBTEST-LBTEND]	/SETUP THE	26	
27	825	070423* 3011	DCA	XR1	/MOVE COUNTER	27	
28	826	070424* 1263	TAD	LBADR/(LBTEST-1)	/SETUP THE	28	
29	827	070425* 3012	DCA	XR2	/DESTINATION POINTER	29	
30	828	070426* 1410	LBMVLP, TAD I	XR0	/GET A WORD	30	
31	829	070427* 6201	CDF	00	/GOING TO FIELD 0	31	
32	830	070430* 6266	CPD		/GOING TO MAIN MEMORY	32	
33	831	070431* 3412	DCA I	XR2	/PUT A WORD	33	
34	832	070432* 6276	SPD		/INDIRECTS BACK TO CP MEMORY AGAIN	34	
35	833	070433* 6271	CDF	70	/BACK TO OUR FIELD	35	
36	834	070434* 2011	ISZ	XR1	/DONE YET?	36	
37	835	070435* 5226	JMP	LBMVLP	/NO, GO BACK	37	
38						38	
39						39	
40						40	
41						41	
42						42	
43						43	
44						44	
45						45	
46						46	
47						47	
48						48	
49						49	
50						50	
51						51	
52						52	
53						53	
54						54	
55						55	
56						56	
57						57	
58						58	

1	837	/	MOVE CP 73000-75600 TO CP 04000-06600. THIS IS THE CP-INTERRUPT HANDLER FOR	1
2	838	/	THE PRINTER LOOPBACK TEST.	2
3	839			3
4	840	/	PROGRAMMING NOTE: THE LENGTH OF EXECUTABLE CODE REQUIRED TO BE RELOCATED IS	4
5	841	/	APPARENTLY MUCH SMALLER THAN THE ACTUAL AMOUNT OF CODE MOVED. THE LENGTH	5
6	842	/	SHOULD BE CPEND-CPINT INSTEAD OF CPEND2-CLOAD. THIS LENGTH MAY INTERFERE	6
7	843	/	WITH AUTOMATIC PRODUCT TESTING OR CAUSE SOME OTHER BUG. IT IS NOT KNOWN AT	7
8	844	/	THIS TIME IF THE ORIGINAL PROGRAMMER'S INTENTION WAS TO OVERLAY ANY OTHER CODE	8
9	845	/	BEYOND THE INTERRUPT HANDLER, BUT THIS SOURCE CODE FOLLOWS THE ROM CONTENTS,	9
10	846	/	BUGS AND ALL :-).	10
11	847			11
12	848	070436* 1140	TAD CPLADR/[CLOAD-1] /SETUP THE	12
13	849	070437* 3010	DCA XR0 /SOURCE POINTER	13
14	850	070440* 1141	TAD CPMCNT/[CLOAD-CPEND2] /SETUP THE	14
15	851	070441* 3011	DCA XR1 /MOVE COUNTER	15
16	852	070442* 7350	NL3777/TAD (CPINT-1) /SETUP THE	16
17	853	070443* 3012	DCA XR2 /DESTINATION POINTER	17
18	854	070444* 1410 CPMVLP,	TAD I XR0 /GET A WORD	18
19	855	070445* 6201	CDF 00 /GOING TO FIELD 0	19
20	856	070446* 3412	DCA I XR2 /PUT A WORD	20
21	857	070447* 6271	CDF 70 /BACK TO OUR FIELD	21
22	858	070450* 2011	ISZ XR1 /DONE YET?	22
23	859	070451* 5244	JMP CPMVLP /NO, KEEP GOING	23
24	860			24
25	861	/	NOW WE HAVE TO MOVE JMP I .-1 TO CP 07777 AND 4000 TO CP 07776 TO COMPLETE THE	25
26	862	/	INTERRUPT HANDLER.	26
27	863			27
28	864	070452* 7330	NL4000/TAD (CPINT) /GET INTERRUPT HANDLING ADDRESS	28
29	865	070453* 6201	CDF 00 /GOING TO FIELD 0	29
30	866	070454* 3534	DCA I CPIADDRESS/[ROMSP] /STORE IN POINTER LOCATION	30
31	867	070455* 1121	TAD JMPIM1/[ROMSP&177+JMPIC] /GET JMP I .-1 INSTRUCTION	31
32	868	070456* 3535	DCA I PROMST/[ROMST] /STORE IN RESTART ADDRESS	32
33	869	070457* 6203	CIF CDF 00 /GOING TO FIELD 0	33
34	870	070460* 6266	CPD /INDIRECTS TO MAIN MEMORY	34
35	871	070461* 6004	PEX /LEAVING CP MEMORY	35
36	872	070462* 5512	JMP I Z200/[200] /GO START IT UP	36
37	873			37
38	874	070463* 0177 LBADR, LBTEST-1	/LOOPBACK TEST ADDRESS (-1)	38
39	875	070464* 0323 PMTSTLP,MTSTLP	/POINTER TO LOOP BRANCH	39
40				40
41				41
42				42
43				43
44				44
45				45
46				46
47				47
48				48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1	877	/	COMES HERE WHEN THE PRINTER LOOPBACK TEST IS DONE TO TEST THE KEYBOARD IN	1
2	878	/	LOOPBACK MODE.	2
3	879			3
4	880	070465* 3054	LBDONE, DCA ESTATUS /SAVE (POSSIBLE ERROR) RESULTS	4
5	881	070466* 1320	TAD L16/(16) /GET 9600 BAUD VALUE	5
6	882	070467* 6333	PRSB /SET BAUD RATE	6
7	883	070470* 7200	CLA /CLEAN UP	7
8	884	070471* 3020	DCA CTMP1 /CLEAR OUTPUT TEST VALUE	8
9	885	070472* 3021	DCA CTMP2 /CLEAR INPUT TEST VALUE	9
10	886	070473* 3022	DCA CNTLO /CLEAR LOW-ORDER COUNTER	10
11	887	070474* 1346	TAD M52/(-52) /SETUP THE	11
12	888	070475* 3023	DCA CNTHI /HIGH-ORDER COUNTER	12
13	889	070476* 6121	APTSKP /AUTOMATIC PRODUCT TESTING?	13
14	890	070477* 1110	TAD Z20/[20] /NO, GET LOOPBACK MODE VALUE	14
15	891	070500* 6126	VIDCON /LOAD NEW MODE EITHER WAY	15
16	892	070501* 6116	KBIRB /READ KEYBOARD DATA TO SET UART	16
17	893	070502* 7200	CLA /THROW IT AWAY	17
18	894	070503* 6054	KBOPC /OUTPUT <NUL> TO UART NOW	18
19	895	070504* 6111	LUPLUP, KBISF /KEYBOARD INPUT FLAG UP?	19
20	896	070505* 7410	SKP /SKIP IF NOT	20
21	897	070506* 5317	JMP LUPIN /JUMP IF SO	21
22	898	070507* 6051	KBOSF /KEYBOARD OUTPUT FLAG UP?	22
23	899	070510* 7410	SKP /SKIP IF NOT	23
24	900	070511* 5334	JMP LUPOUT /JUMP IF SO	24
25	901	070512* 2022	ISZ CNTLO /DONE WAITING?	25
26	902	070513* 5304	JMP LUPLUP /NO, KEEP GOING	26
27	903	070514* 2023	ISZ CNTHI /ENOUGH TIMES?	27
28	904	070515* 5304	JMP LUPLUP /NO, KEEP GOING	28
29	905	070516* 5353	JMP LUPTIMOUT /YES, CONTINUE THERE	29
30	906			30
31	907	070517* 6111	LUPIN, KBISF /SKIP ON, CLEARING THE INPUT FLAG	31
32	908	070520* 0016	L16, 16 /9600 BAUD RATE VALUE; HERE IN CASE IT SKIPS	32
33	909	070521* 6116	KBIRB /GET THE INPUT VALUE	33
34	910	070522* 0115	AND Z377/[377] /JUST 8-BIT CHARACTER	34
35	911	070523* 7041	CIA /INVERT FOR TEST	35
36	912	070524* 1021	TAD CTMP2 /COMPARE TO TEST VALUE	36
37	913	070525* 7640	SZA CLA /SKIP IF IT MATCHES	37
38	914	070526* 6126	VIDCON /CLEAR LOOPBACK IF NOT	38
39	915	070527* 1021	TAD CTMP2 /GET INPUT PATTERN VALUE	39
40	916	070530* 4747	JMS I PCOUNTUP/(COUNTUP) /BUMP TO NEXT VALUE	40
41	917	070531* 5354	JMP LUPDONE /ALL VALUES USED	41
42	918	070532* 3021	DCA CTMP2 /SAVE UPDATED VALUE	42
43	919	070533* 5304	JMP LUPLUP /KEEP GOING	43
44	920			44
45	921	070534* 6051	LUPOUT, KBOSF /SKIP ON, THUS CLEARING OUTPUT FLAG	45
46	922	070535* 7000	NOP /JUST IN CASE IT SKIPS	46
47	923	070536* 1020	TAD CTMP1 /GET COUNT PATTERN VALUE	47
48	924	070537* 4747	JMS I PCOUNTUP/(COUNTUP) /BUMP TO NEXT VALUE	48
49	925	070540* 5304	JMP LUPLUP /ALL VALUES USED; KEEP GOING FOR INPUT'S SAKE	49
50	926	070541* 3020	DCA CTMP1 /SAVE UPDATED VALUE	50
51	927	070542* 1020	TAD CTMP1 /GET IT BACK	51
52	928	070543* 6056	KBOLS /OUTPUT IT	52
53	929	070544* 7200	CLA /CLEAN UP	53
54	930	070545* 5304	JMP LUPLUP /KEEP GOING	54
55				55
56				56
57				57
58				58

1	932	070546*	7726	M52,	-52	/CONSTANT 7726	1
2	933	070547*	2734	PCOUNTU,	COUNTUP-LBTEST+LBLOAD	/POINTER TO COUNTUP ROUTINE	2
3	934	070550*	0375	L375,	375	/CONSTANT 0375	3
4	935	070551*	5600	PTSTBUF,	TSTBUFFER-1	/POINTER TO TEST BUFFER	4
5	936	070552*	0630	PKBTEND,	KBTEND	/POINTER TO END OF KEYBOARD TEST	5
6	937						6
7	938			/	COMES HERE WHEN THE LOOPBACK TEST TIMES OUT.		7
8	939						8
9	940	070553*	7315	LUPTIMO,	NL0010 CLL	/INDICATE TIME-OUT ENDED THE TEST	9
10	941						10
11	942			/	COMES HERE WHEN THE COUNT PATTERN IS FINISHED TO (POSSIBLY) TEST THE KEYBOARD.		11
12	943						12
13	944	070554*	1054	LUPDONE,	TAD ESTATUS	/GET TEST ERROR STATUS	13
14	945	070555*	3054	DCA	ESTATUS	/STORE (POSSIBLY) UPDATED VALUE	14
15	946	070556*	6126	VIDCON		/CLEAR VIDEO CONTROL	15
16	947	070557*	6121	APTSKP		/AUTOMATIC PRODUCT TESTING?	16
17	948	070560*	7410	SKP		/SKIP IF NOT	17
18	949	070561*	5752	JMP I	PKBTEND/(KBTEND)	/JUMP IF SO TO BYPASS TEST	18
19	950	070562*	3012	DCA	XR2	/CLEAR RECEIVED CHARACTER COUNT	19
20	951	070563*	6116	KBIRB		/READ BUFFER NOW TO RESET UART	20
21	952	070564*	7200	CLA		/CLEAN UP	21
22	953	070565*	1350	TAD	L375/[375]	/GET "JUMP TO POWER UP" VALUE	22
23	954	070566*	6056	KBOLS		/OUTPUT TO KEYBOARD	23
24	955	070567*	6051	KBOSF		/WAIT FOR IT	24
25	956	070570*	5367	JMP	.-1	/'TIL DONE	25
26	957	070571*	6051	KBOSF		/SKIP ON, THUS CLEARING THE FLAG	26
27	958	070572*	0040	40/NOP		/HERE IN CASE IT SKIPS (UNREFERENCED?)	27
28	959	070573*	7200	CLA		/CLEAN UP	28
29	960	070574*	1126	TAD	Z7766/[-12]	/SETUP THE	29
30	961	070575*	3023	DCA	CNTHI	/HIGH-ORDER COUNT	30
31	962	070576*	3022	DCA	CNTLO	/CLEAR LOW-ORDER COUNT	31
32	963	070577*	1351	TAD	PTSTBUFF/(TSTBUFFER-1)	/SETUP THE	32
33	964	070600*	3011	DCA	XR1	/BUFFER POINTER	33
34	965	070601*	6111	KEYLUP,	KBISF	/FLAG UP?	34
35	966	070602*	7410	SKP		/SKIP IF NOT	35
36	967	070603*	5211	JMP	GOTKEY	/JUMP IF SO	36
37	968	070604*	2022	ISZ	CNTLO	/WAITING TOO LONG?	37
38	969	070605*	5201	JMP	KEYLUP	/NO, KEEP TRYING	38
39	970	070606*	2023	ISZ	CNTHI	/TOO MANY TIMES?	39
40	971	070607*	5201	JMP	KEYLUP	/NO, KEEP TRYING	40
41	972						41
42	973			/	NO (FURTHER) KEYBOARD INPUT.		42
43	974						43
44	975	070610*	5220	JMP	KBTIMOUT	/CONTINUE THERE	44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	977	/	COMES HERE WHEN A KEYBOARD CHARACTER IS RECEIVED.		1
2	978				2
3	979	070611* 6111	GOTKEY, KBISF	/SKIP ON, THUS CLEARING THE FLAG	3
4	980	070612* 7000	NOP	/IN CASE IT SKIPS	4
5	981	070613* 6116	KBIRB	/GET THE CHARACTER	5
6	982	070614* 0115	AND Z377/[377]	/JUST 8 BITS	6
7	983	070615* 3411	DCA I XR1	/STORE INTO THE BUFFER	7
8	984	070616* 2012	ISZ XR2	/BUMP CHARACTER COUNT	8
9	985	070617* 5201	JMP KEYLUP	/KEEP GOING	9
10	986				10
11	987	/	COMES HERE WHEN KEYBOARD INPUT TIMES OUT.		11
12	988				12
13	989	070620* 1233	KBTIMOU, TAD L7774/(-4)	/GET EXPECTED COUNT	13
14	990	070621* 1012	TAD XR2	/COMPARE TO ACTUAL	14
15	991	070622* 7450	SNA	/SKIP IF OTHER THAN EXPECTED	15
16	992	070623* 1632	TAD I PTSTBEND/(TSTBEND)	/ELSE LOOK AT LAST CHARACTER RECEIVED	16
17	993	070624* 7640	SZA CLA	/SKIP IF LAST CHARACTER WAS A <NUL>	17
18	994	070625* 1110	TAD Z20/[20]	/ELSE ADD ON KEYBOARD ERROR VALUE	18
19	995	070626* 1054	TAD ESTATUS	/GET CUMULATIVE STATUS	19
20	996	070627* 3054	DCA ESTATUS	/STORE BACK	20
21	997	070630* 5631	KBTEND, JMP I .+1/(COMTEST)	/CONTINUE THERE	21
22	998	070631* 4652	COMTEST	/THROUGH HERE	22
23	999				23
24	1000	070632* 5604	PTSTBEN, TSTBEND-1	/POINTER TO END OF TEST KEYBOARD BUFFER	24
25	1001	070633* 7774	L7774, 7774	/CONSTANT 7774	25
26	1002	070634* 0342	L342, 342	/CONSTANT 0342	26
27	1003	070635* 0320	L320, 320	/CONSTANT 0320	27
28	1004	070636* 7740	A7740, 7740	/CONSTANT 7740	28
29	1005	070637* 6000	PAPTSTR, APTSTR	/STARTING ADDRESS FOR DOWNLOAD	29
30					30
31					31
32					32
33					33
34					34
35					35
36					36
37					37
38					38
39					39
40					40
41					41
42					42
43					43
44					44
45					45
46					46
47					47
48					48
49					49
50					50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	1007	/	DIRECT DOWN-LOAD ROUTINE. CALLED VIA JMS I .+1;DOWNLOAD IF APTEN IS SET. THE			1
2	1008	/	FORMAT IS STANDARD PAPER-TAPE RIM FORMAT WITHOUT A LEADER. LOADING IS TO MAIN			2
3	1009	/	MEMORY FIELD 7. THE ROUTINE IS ALSO CALLABLE AFTER A PREVIOUSLY LOADED			3
4	1010	/	PROGRAM EXITS VIA A PANEL REQUEST INSTEAD OF A HLT INSTRUCTION. THIS CAUSES A			4
5	1011	/	TOTAL RESTART OF ALL CODE INITIALLY MOVED FROM THE ROM TO CP FIELD 7. IN THIS			5
6	1012	/	SITUATION, CODE LOADED INTO MAIN MEMORY 76000-77777 WILL BE MOVED TO CP			6
7	1013	/	76000-7777 TO BECOME THE NEXT CP-INTERRUPT HANDLER, ETC. EXIT VIA HLT			7
8	1014	/	INSTRUCTION RETURNS TO A DESIGNATED ENTRY POINT TO LOAD MORE CODE.			8
9	1015					9
10	1016	070640* 0000	DOWNLOA,.-.		/DOWN-LOAD ROUTINE	10
11	1017	070641* 2240	ISZ	DOWNLOAD	/BUMP PAST CALL POINTER	11
12	1018	070642* 6271	CDF	70	/ENSURE OUR FIELD	12
13	1019	070643* 1116	TAD	Z400/[400]	/SET POWER-ON MASK	13
14	1020	070644* 0061	AND	PSTATUS	/GET POWER-ON BIT	14
15	1021	070645* 7650	SNA CLA		/SKIP IF SET	15
16	1022	070646* 5337	JMP	DWNAGN	/JUMP IF NOT	16
17	1023	070647* 3355	DCA	CMSW	/INDICATE MOVABLE CODE NOT LOADED YET	17
18	1024					18
19	1025	/	THE CODE IS REJOINED HERE WHEN DOWN-LOADED CODE EXITS WITH A HLT INSTRUCTION.			19
20	1026					20
21	1027	070650* 6271	DWNENTR,CDF	70	/GOTO MAIN MEMORY LOADING FIELD	21
22	1028	070651* 1255	TAD	PBAUD	/GET BAUD RATE	22
23	1029	070652* 6333	PRSB		/SETUP 9600 BAUD RATE	23
24	1030	070653* 7200	CLA		/CLEAN UP	24
25	1031	070654* 6321	PRISKP		/SKIP ON, THUS CLEARING INPUT FLAG	25
26	1032	070655* 0016	PBAUD, 16		/RATE VALUE FOR 9600 BAUD; ALSO SKIP PROTECT	26
27	1033	070656* 1234	TAD	L342/(342)	/GET CHARACTER CODE	27
28	1034	070657* 6336	PROLS		/OUTPUT IT	28
29	1035	070660* 6326	PRIRB		/READ KEYBOARD BUFFER NOW FOR CHARACTERS LATER	29
30	1036	070661* 6266	CPD		/MAIN MEMORY INDIRECTS FROM NOW ON	30
31	1037	070662* 6321	RIMLUP, PRISKP		/FLAG UP	31
32	1038	070663* 5262	JMP	RIMLUP	/NO, WAIT FOR IT	32
33	1039	070664* 6321	PRISKP		/YES, MAKE SURE IT CLEARS	33
34	1040	070665* 7000	NOP		/IN CASE IT SKIPS AGAIN	34
35	1041	070666* 6326	PRIRB		/READ IN THE CHARACTER	35
36	1042	070667* 7106	CLL RTL;RTL		/MOVE UP	36
37	1043	070670* 7006				37
38	1044	070671* 7510	SPA		/SKIP IF NOT AT TRAILER	38
39	1045	070672* 5305	JMP	RIMEND	/JUMP IF DONE	39
40	1046	070673* 7006	RTL		/MOVE UP TO HIGH-ORDER	40
41	1047	070674* 6321	PRISKP		/WAIT FOR	41
42	1048	070675* 5274	JMP	.-1	/NEXT CHARACTER	42
43	1049	070676* 6321	PRISKP		/MAKE SURE IT CLEARS	43
44	1050	070677* 7000	NOP		/IN CASE IT SKIPS AGAIN	44
45	1051	070700* 6324	PRIRS		/OR IN LATEST CHARACTER	45
46	1052	070701* 7420	SNL		/SKIP IF ORIGIN SETTING	46
47	1053	070702* 3400	DCA I	RIMADR	/STORE THE DATA	47
48	1054	070703* 3000	DCA	RIMADR	/STORE AS NEW ORIGIN	48
49	1055	070704* 5262	JMP	RIMLUP	/KEEP GOING	49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	1057	/	COMES HERE AT THE END OF THE RIM LOADING.					1
2	1058							2
3	1059	070705* 7200	RIMEND, CLA			/CLEAN UP		3
4	1060	070706* 1235	TAD	L320/(320)		/GET THIS VALUE		4
5	1061	070707* 3636	DCA I	A7740/(7740)		/SOMEONE NEEDS THIS THERE		5
6	1062	070710* 6276	SPD			/CP MEMORY INDIRECTS		6
7	1063	070711* 3010	DCA	XR0		/CLEAR INDEX (WHY?)		7
8	1064	070712* 6201	CDF	00		/GOING TO FIELD 0		8
9	1065	070713* 1357	TAD	PCPIHND/(CPIHND)		/GET INTERRUPT HANDLER ADDRESS		9
10	1066	070714* 3534	DCA I	CPIADDR/[ROMSP]		/STORE IN INTERRUPT POINTER		10
11	1067	070715* 1356	TAD	JMPIPM/(ROMSP&177+JMPIC)		/GET JMP I .-1		11
12	1068	070716* 3535	DCA I	PROMST/[ROMST]		/STORE IN RESTART ADDRESS		12
13	1069	070717* 7240	NL7777			/BACKUP		13
14	1070	070720* 1357	TAD	PCPIHND/(CPIHND)		/NOW HAVE AUTO-INCREMENT LOADING ADDRESS		14
15	1071	070721* 3010	DCA	XR0		/STASH THE POINTER		15
16	1072	070722* 1010	TAD	XR0		/GET IT AGAIN		16
17	1073	070723* 3011	DCA	XR1		/STASH FOR MOVING ALSO		17
18	1074	070724* 6271	CDF	70		/GOTO OUR FIELD		18
19	1075	070725* 1410	RCPLUP, TAD I	XR0		/GET A WORD		19
20	1076	070726* 7450	SNA			/END OF LIST?		20
21	1077	070727* 5334	JMP	RIMST		/JUMP IF SO		21
22	1078	070730* 6201	CDF	00		/GOTO FIELD 0		22
23	1079	070731* 3411	DCA I	XR1		/PUT A WORD		23
24	1080	070732* 6271	CDF	70		/BACK TO OUR FIELD		24
25	1081	070733* 5325	JMP	RCPLUP		/KEEP GOING		25
26	1082							26
27	1083	070734* 6273	RIMST, CIF CDF	70		/GOING TO FIELD 7		27
28	1084	070735* 6004	PEX			/LEAVING PANEL MEMORY		28
29	1085	070736* 5637	JMP I	PAPTSTRT/(APTSTRT)		/GO THERE		29
30	1086							30
31	1087	/	COMES HERE IF A PREVIOUSLY LOADED ROUTINE EXITED VIA A PANEL REQUEST WHICH					31
32	1088	/	CAUSES A TOTAL RESTART OF THE FORMERLY ROM-BASED CODE MOVED TO FIELD 7. CODE					32
33	1089	/	LOADED INTO MAIN MEMORY 76000-77777 SHOULD BE MOVED TO CP 76000-77777.					33
34	1090							34
35	1091	070737* 1355	DWNAGN, TAD	CMSW		/GET LOAD SWITCH		35
36	1092	070740* 7640	SZA CLA			/SKIP IF NOT LOADED YET		36
37	1093	070741* 5640	JMP I	DOWNLOAD		/RETURN TO CALLER		37
38	1094	070742* 7240	NL7777			/INDICATE CODE MOVED		38
39	1095	070743* 3355	DCA	CMSW		/FOR NEXT TIME		39
40	1096	070744* 1237	TAD	PAPTSTRT/(APTSTRT)		/GET STARTING ADDRESS		40
41	1097	070745* 3000	DCA	RIMADR		/STASH THE POINTER		41
42	1098	070746* 6266	MCLUP, CPD			/MAIN MEMORY INDIRECTS		42
43	1099	070747* 1400	TAD I	RIMADR		/GET A WORD		43
44	1100	070750* 6276	SPD			/CP MEMORY INDIRECTS		44
45	1101	070751* 3400	DCA I	RIMADR		/PUT A WORD		45
46	1102	070752* 2000	ISZ	RIMADR		/BUMP TO NEXT		46
47	1103	070753* 5346	JMP	MCLUP		/KEEP GOING		47
48	1104	070754* 5640	JMP I	DOWNLOAD		/RETURN TO CALLER		48
49								49
50								50
51								51
52								52
53								53
54								54
55								55
56								56
57								57
58								58

1	1106	070755*	0000	CMSW, .-. /CODE MOVED SWITCH	1
2	1107	070756*	5776	JMPIPM, ROMSP&177+JMPIC /JMP I .-1 CONSTANT FOR RESTART ADDRESS	2
3	1108	070757*	2536	PCPIHND, CPIHND /CP-INTERRUPT HANDLING ADDRESS	3
4	1109				4
5	1110			/ VALIDATION PATTERN FOR TRACK ZERO, SECTOR 4 DATA.	5
6	1111				6
7	1112	070760*	7777	TOS4DAT, -1	7
8	1113	070761*	7670	- "H!200	8
9	1114	070762*	7664	- "L!200	9
10	1115	070763*	7654	- "T!200	10
11	1116	070764*	7672	- "F!200	11
12	1117	070765*	7664	- "L!200	12
13	1118	070766*	7671	- "G!200	13
14	1119	070767*	7776	-2	14
15	1120	070770*	0000	0 /THIS ENDS THE LIST	15
16	1121				16
17	1122			/ DATA PATTERN USED TO TEST RX50 SILO BUFFER.	17
18	1123				18
19	1124	070771*	0000	TPATTRN, 000	19
20	1125	070772*	0252	252	20
21	1126	070773*	0125	125	21
22	1127	070774*	0063	063	22
23	1128	070775*	0017	017	23
24	1129	070776*	0377	377	24
25	1130				25
26	1131	070777*	0000	ZBLOCK 1 /EMPTY SPACE	26
27	1132				27
28	1133	071000*	0000	DINDEX, .-. /TEST DATA INDEX	28
29	1134	071001*	0000	DCNT, .-. /TEST DATA COUNTER	29
30	1135	071002*	0000	DPTR, .-. /TEST DATA POINTER	30
31	1136	071003*	7773	M5, -5 /CONSTANT 7773	31
32	1137	071004*	7000	M1000, -1000 /CONSTANT 7000	32
33	1138	071005*	0771	PTPATTR, TPATTRN /POINTER TO SILO TEST DATA	33
34	1139	071006*	5356	PRDTWAI, RDTWAIT /POINTER TO RDTWAIT ROUTINE	34
35					35
36					36
37					37
38					38
39					39
40					40
41					41
42					42
43					43
44					44
45					45
46					46
47					47
48					48
49					49
50					50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	1196	/	DONE FLAG IS UP; CHECK IF PROPER BYTE COUNT.			1
2	1197					2
3	1198	071067* 1201	DONEUP, TAD	DCNT	/GET DATA COUNTER	3
4	1199	071070* 1204	TAD	M1000/(-1000)	/COMPARE TO EXPECTED VALUE	4
5	1200	071071* 7640	SZA CLA		/SKIP IF IT MATCHES	5
6	1201	071072* 5376	JMP	RXERROR	/ELSE COMPLAIN	6
7	1202	071073* 7300	CLA CLL		/CLEAN UP	7
8	1203	071074* 3200	DCA	DINDEX	/CLEAR DATA INDEX	8
9	1204	071075* 3201	DCA	DCNT	/CLEAR DATA COUNTER	9
10	1205	071076* 1111	TAD	Z102/[102]	/GET EMPTY BUFFER COMMAND (8-BIT) VALUE	10
11	1206	071077* 6751	LCD		/LOAD THE COMMAND	11
12	1207	071100* 4340	JMS	RXWAIT	/CALL WAIT ROUTINE	12
13	1208	071101* 5304	JMP	CMDOK	/TRANSFER FLAG CAME UP	13
14	1209	071102* 5376	JMP	RXERROR	/DONE FLAG CAME UP, COMPLAIN	14
15	1210	071103* 5376	JMP	RXERROR	/ERROR OR TIME-OUT	15
16	1211					16
17	1212	071104* 7200	CMDOK, CLA		/CLEAN UP	17
18	1213	071105* 6752	XDR		/FINISH THE COMMAND	18
19	1214	071106* 4340	EMPTLUP, JMS	RXWAIT	/CALL WAIT ROUTINE	19
20	1215	071107* 7410	SKP		/TRANSFER FLAG CAME UP	20
21	1216	071110* 5332	JMP	DONUP	/DONE FLAG CAME UP	21
22	1217					22
23	1218	/	OOPS! FORGOT THE "JMP RXERROR" IF IT GETS AN ERROR OR TIMES OUT.			23
24	1219					24
25	1220	071111* 1200	TAD	DINDEX	/GET DATA INDEX	25
26	1221	071112* 1205	TAD	PTPATTRN/(TPATTRN)	/POINT TO PROPER ELEMENT IN LIST	26
27	1222	071113* 3202	DCA	DPTR	/STASH THE POINTER	27
28	1223	071114* 1602	TAD I	DPTR	/GET THE DATA	28
29	1224	071115* 3202	DCA	DPTR	/SAVE IT	29
30	1225	071116* 6752	XDR		/GET THE BYTE	30
31	1226	071117* 7041	CIA		/INVERT FOR TESTING	31
32	1227	071120* 1202	TAD	DPTR	/COMPARE TO TEST VALUE	32
33	1228	071121* 7640	SZA CLA		/SKIP IF IT MATCHES	33
34	1229	071122* 5376	JMP	RXERROR	/ELSE COMPLAIN	34
35	1230	071123* 2201	ISZ	DCNT	/BUMP DATA COUNTER	35
36	1231	071124* 2200	ISZ	DINDEX	/BUMP DATA INDEX	36
37	1232	071125* 1200	TAD	DINDEX	/GET DATA INDEX	37
38	1233	071126* 1203	TAD	M5/(-5)	/COMPARE TO LARGEST VALUE	38
39	1234	071127* 7740	SMA SZA CLA		/SKIP IF NOT TOO FAR	39
40	1235	071130* 3200	DCA	DINDEX	/ELSE RESET DATA INDEX	40
41	1236	071131* 5306	JMP	EMPTLUP	/KEEP GOING	41
42	1237					42
43	1238	/	DONE FLAG IS UP; CHECK IF PROPER BYTE COUNT.			43
44	1239					44
45	1240	071132* 1201	DONUP, TAD	DCNT	/GET DATA COUNTER	45
46	1241	071133* 1204	TAD	M1000/(-1000)	/COMPARE TO EXPECTED VALUE	46
47	1242	071134* 7640	SZA CLA		/SKIP IF IT MATCHES	47
48	1243	071135* 5376	JMP	RXERROR	/ELSE COMPLAIN	48
49	1244	071136* 5737	JMP I	+.1/(RXOK)	/CONTINUE THERE	49
50	1245	071137* 1201	RXOK		/THROUGH HERE	50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	1247	/				RX50 WAIT ROUTINE. TAKES IMMEDIATE RETURN IF TRANSFER FLAG RAISES. TAKES	1
2	1248	/				SKIP RETURN IF DONE FLAG RAISES. TAKES DOUBLE-SKIP RETURN IF AN ERROR OCCURS	2
3	1249	/				OR THE HARDWARE TIMES OUT.	3
4	1250						4
5	1251	071140*	0000	RXWAIT, .-. .			5
6	1252	071141*	1132	TAD	DTIMOUT/[-DTIME]	/SETUP THE	6
7	1253	071142*	3050	DCA	T6	/TIME-OUT COUNTER	7
8	1254	071143*	3051	DCA	T7	/CLEAR INNER TIME-OUT COUNTER	8
9	1255	071144*	6753	RXWATLP, STR		/TRANSFER FLAG UP?	9
10	1256	071145*	7410	SKP		/SKIP IF NOT	10
11	1257	071146*	5740	JMP I	RXWAIT	/YES, TAKE IMMEDIATE RETURN	11
12	1258	071147*	6755	SDN		/DONE FLAG UP?	12
13	1259	071150*	5354	JMP	RXWATING	/NO, KEEP TRYING	13
14	1260	071151*	6754	SER		/YES, WERE THERE ERRORS?	14
15	1261	071152*	5361	JMP	RXSKIP	/NO, TAKE SKIP RETURN	15
16	1262	071153*	5360	JMP	RXSKP2	/YES, TAKE DOUBLE SKIP RETURN	16
17	1263						17
18	1264	071154*	2051	RXWATIN, ISZ	T7	/WAITING TOO LONG?	18
19	1265	071155*	5344	JMP	RXWATLP	/NO, KEEP TRYING	19
20	1266	071156*	2050	ISZ	T6	/WAITED TOO LONG?	20
21	1267	071157*	5344	JMP	RXWATLP	/NO, KEEP TRYING	21
22	1268	071160*	2340	RXSKP2, ISZ	RXWAIT	/YES, TAKE DOUBLE-SKIP RETURN	22
23	1269	071161*	2340	RXSKIP, ISZ	RXWAIT	/SET SKIP RETURN FOR DONE FLAG UP CONDITION	23
24	1270	071162*	5740	JMP I	RXWAIT	/RETURN AS NECESSARY	24
25	1271						25
26	1272	071163*	7700	M100, -100		/CONSTANT 7700	26
27	1273						27
28	1274	071164*	0000	TIMOUT, .-. .		/GENERAL TIME-OUT ROUTINE	28
29	1275	071165*	3050	DCA	T6	/CLEAR INNER COUNTER	29
30	1276	071166*	1363	TAD	M100/(-100)	/SETUP THE	30
31	1277	071167*	3051	DCA	T7	/OUTER COUNTER	31
32	1278	071170*	2050	TIMLUP, ISZ	T6	/WAITING LONG ENOUGH?	32
33	1279	071171*	5370	JMP	TIMLUP	/NO, KEEP GOING	33
34	1280	071172*	2051	ISZ	T7	/WAITED TOO LONG?	34
35	1281	071173*	5370	JMP	TIMLUP	/NO, KEEP GOING	35
36	1282	071174*	5764	JMP I	TIMOUT	/YES, RETURN TO CALLER	36
37	1283						37
38	1284	071175*	0040	L40, 40		/CONSTANT 0040	38
39	1285						39
40	1286	071176*	1375	RXERROR, TAD	L40/(40)	/GET RX50 ERROR VALUE	40
41	1287	071177*	1054	TAD	ESTATUS	/UPDATE CUMULATIVE ERROR STATUS	41
42	1288	071200*	3054	DCA	ESTATUS	/STORE BACK	42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	1290	071201*	7315	RXOK,	NL0010	CLL	/SET CRT VALUE	1
2	1291	071202*	6126		VIDCON		/LOAD ECHRMOD	2
3	1292	071203*	7200		CLA		/CLEAN UP	3
4	1293	071204*	4643		JMS I	PENDUP/(ENDUP)	/FINISH UP SCREEN DISPLAY (POSSIBLY ERRORS)	4
5	1294	071205*	4635		JMS I	PRDSETUP/(RDSETUP)	/GET RD STARTUP CODE READ IN (IF PRESENT)	5
6	1295	071206*	4634		JMS I	PRXRDY/(RXRDY)	/WAIT UNTIL DRIVE ZERO IS READY	6
7	1296	071207*	5206		JMP	.-1	/WAIT FOR IT TO BE READY INDEFINITELY	7
8	1297	071210*	4544		JMS I	PCLR23/[CLR23]	/CLEAR FIELDS 2, 3 (AND BITS IN 13705, 13707)	8
9	1298	071211*	7240	RTRK1,	NL7777		/SETUP THE	9
10	1299	071212*	3010		DCA	XR0	/STORAGE POINTER	10
11	1300	071213*	6201		CDF	00	/STORING IN FIELD 0	11
12	1301	071214*	1111		TAD	Z102/[102]	/GET EMPTY SILO (8-BIT) COMMAND	12
13	1302	071215*	3053		DCA	RXCMD	/STASH IT	13
14	1303			/	CLA		/CALL WITH AC CLEAR (1-1)	14
15	1304	071216*	4311		JMS	FREAD	/CALL FLOPPY READ ROUTINE	15
16	1305	071217*	7776		-1-1		/ONE SECTOR	16
17	1306	071220*	0001		1		/ON TRACK 1	17
18	1307	071221*	5211		JMP	RTRK1	/ERROR, TRY AGAIN	18
19	1308	071222*	4632		JMS I	PVALID8/(VALID8)	/VALIDATE THE PREFERENCE SETTINGS	19
20	1309	071223*	5236		JMP	FLPBAD	/FLOPPY IS BAD, GO COMPLAIN	20
21	1310	071224*	4244	SLSHAGN,	JMS	RXLOAD	/LOAD IN VARIOUS SLUSHWARE ITEMS	21
22	1311	071225*	5240		JMP	SLSHBAD	/COULDN'T DO IT	22
23	1312	071226*	4544		JMS I	PCLR23/[CLR23]	/CLEAR FIELDS 2, 3 (AND BITS IN 13705, 13707)	23
24	1313	071227*	7200		CLA		/CLEAN UP	24
25	1314	071230*	6203		CIF	CDF 00	/GOING TO FIELD 0	25
26	1315	071231*	5512		JMP I	Z200/[200]	/GO START IT UP	26
27	1316							27
28	1317	071232*	1600	PVALID8,	VALID8		/POINTER TO VALIDATION ROUTINE	28
29	1318	071233*	1554	PFBLINK,	FBLINK		/POINTER TO FBLINK ROUTINE	29
30	1319	071234*	3701	PRXRDY,	RXRDY		/POINTER TO RX50 READY CHECK ROUTINE	30
31	1320	071235*	5000	PRDSETU,	RDSETUP		/POINTER TO RD51 SETUP ROUTINE	31
32	1321							32
33	1322			/		COMES HERE TO DISPLAY A BLINKING BIG FLOPPY PATTERN IF THE CURRENT DISKETTE		33
34	1323			/		HAS AN INVALID BOOT BLOCK IN TRACK 1, SECTOR 1.		34
35	1324							35
36	1325	071236*	4633	FLPBAD,	JMS I	PFBLINK/(FBLINK)	/GO DISPLAY THE PATTERN ONCE	36
37	1326	071237*	5211		JMP	RTRK1	/TRY AGAIN	37
38	1327							38
39	1328			/		COMES HERE TO DISPLAY THE BIG FLOPPY PATTERN IF THE SLUSHWARE STUFF CAN'T BE		39
40	1329			/		COMPLETELY READ IN.		40
41	1330							41
42	1331	071240*	4633	SLSHBAD,	JMS I	PFBLINK/(FBLINK)	/GO DISPLAY THE PATTERN ONCE	42
43	1332	071241*	5224		JMP	SLSHAGN	/TRY AGAIN	43
44	1333							44
45	1334	071242*	1140	PRXWAIT,	RXWAIT		/POINTER TO RX50 WAIT ROUTINE	45
46	1335	071243*	2444	PENDUP,	ENDUP		/POINTER TO ENDUP ROUTINE	46
47								47
48								48
49								49
50								50
51								51
52								52
53								53
54								54
55								55
56								56
57								57
58								58

1	1337	071244*	0000	RXLOAD, .-. /		/MULTIPLE RX50 LOADING ROUTINE	1
2	1338	071245*	7240	NL7777		/SETUP THE	2
3	1339	071246*	3010	DCA XR0		/STORAGE POINTER	3
4	1340	071247*	1111	TAD Z102/[102]		/SETUP THE	4
5	1341	071250*	3053	DCA RXCMD		/EMPTY BUFFER COMMAND (8-BIT)	5
6	1342	071251*	4707	JMS I PT0S3RD/(T0S3RD)		/READ AND VALIDATE TRACK 0, SECTOR 3	6
7	1343	071252*	5644	JMP I RXLOAD		/COULDN'T DO IT, TAKE FAILURE RETURN	7
8	1344	071253*	4710	JMS I PDOCG/(DOCG)		/READ AND LOAD CHARACTER GENERATOR RAM DATA	8
9	1345	071254*	7305	NL002A		/SETUP 12-BIT EMPTY SILO COMMAND VALUE	9
10	1346	071255*	3053	DCA RXCMD		/STASH THE COMMAND	10
11	1347	071256*	7240	NL7777		/SETUP THE	11
12	1348	071257*	3010	DCA XR0		/DATA POINTER	12
13	1349	071260*	6201	CDF 00		/SETUP THE DATA FIELD	13
14	1350			CLA /		/CALL WITH AC CLEAR (1-1)	14
15	1351	071261*	4311	JMS FREAD		/CALL FLOPPY READ ROUTINE	15
16	1352	071262*	7765	-12-1		/READ 10 SECTORS (1-10)	16
17	1353	071263*	0116	116		/TRACK 78	17
18	1354	071264*	5644	JMP I RXLOAD		/COULDN'T DO IT, TAKE FAILURE RETURN	18
19	1355	071265*	6201	CDF 00		/SETUP THE DATA FIELD	19
20	1356			CLA /		/CALL WITH AC CLEAR (1-1)	20
21	1357	071266*	4311	JMS FREAD		/CALL FLOPPY READ ROUTINE	21
22	1358	071267*	7771	-6-1		/READ 6 SECTORS (1-6)	22
23	1359	071270*	0117	117		/TRACK 79	23
24	1360	071271*	5644	JMP I RXLOAD		/COULDN'T DO IT, TAKE FAILURE RETURN	24
25	1361	071272*	6211	CDF 10		/SETUP THE DATA FIELD	25
26	1362	071273*	7327	NL0006		/SETUP FOR SECTOR 7 FIRST	26
27	1363	071274*	4311	JMS FREAD		/CALL FLOPPY READ ROUTINE	27
28	1364	071275*	7765	-12-1		/READ 4 SECTORS (7-10)	28
29	1365	071276*	0117	117		/TRACK 79	29
30	1366	071277*	5644	JMP I RXLOAD		/COULDN'T DO IT, TAKE FAILURE RETURN	30
31	1367	071300*	6201	CDF 00		/BACK TO PRIMARY CODE FIELD	31
32	1368	071301*	3010	DCA XR0		/CLEAR THE POINTER	32
33	1369	071302*	4706	JMS I PSLCHECK/(SLCHECK)		/GO VALIDATE THE DATA	33
34	1370	071303*	5644	JMP I RXLOAD		/BAD DATA, TAKE FAILURE RETURN	34
35	1371	071304*	2244	ISZ RXLOAD		/GOOD DATA, BUMP RETURN ADDRESS	35
36	1372	071305*	5644	JMP I RXLOAD		/TAKE SKIP RETURN TO CALLER	36
37	1373						37
38	1374	071306*	4153	PSLCHEC,SLCHECK		/POINTER TO SLUSHWARE VALIDATION ROUTINE	38
39	1375	071307*	3111	PT0S3RD,T0S3RD		/POINTER TO TRACK 0, SECTOR 3 READ ROUTINE	39
40	1376	071310*	3152	PDOCG, DOCG		/POINTER TO DOCG ROUTINE	40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	1431	071360*	6752	COMFINI,XDR		/FINISH THE COMMAND	1
2	1432	071361*	7402	TRNCDF, HLT+.-.		/WILL BE CDF XX INSTRUCTION TO BUFFER FIELD	2
3	1433	071362*	7200	CLA		/CLEAN UP	3
4	1434	071363*	6753	FRDLUP, STR		/TRANSFER FLAG UP?	4
5	1435	071364*	5367	JMP	TRYDONE	/NO, TRY DONE FLAG	5
6	1436	071365*	6752	XDR		/YES, GET THE LATEST	6
7	1437	071366*	3410	DCA I	XR0	/STORE IT	7
8	1438	071367*	6755	TRYDONE,SDN		/DONE FLAG UP?	8
9	1439	071370*	5363	JMP	FRDLUP	/NO, TRY TRANSFER	9
10	1440	071371*	6754	SER		/YES, ANY ERRORS?	10
11	1441	071372*	5325	JMP	FNXTSCT	/NO, GO DO NEXT SECTOR	11
12	1442	071373*	5343	JMP	FERR	/YES, GO COMPLAIN	12
13	1443						13
14	1444	071374*	0000	ZBLOCK	4	/EMPTY SPACE	14
15							15
16							16
17							17
18							18
19							19
20							20
21							21
22							22
23							23
24							24
25							25
26							26
27							27
28							28
29							29
30							30
31							31
32							32
33							33
34							34
35							35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	1446	*1400	PAGE			1
2	1447					2
3	1448	/	AUTO-INDEXED POINTER TO PIE CHIP PROGRAMMING TABLE (TABLE FOLLOWS).			3
4	1449					4
5	1450	071400* 1400	PPTABLE,PTABLE-1	/POINTER TO PROGRAMMING TABLE.		5
6	1451					6
7	1452	/	PROGRAMMING NOTE: BY PLACING THIS POINTER HERE, AN EXTRANEIOUS PAGE ZERO			7
8	1453	/	REFERENCE TO THE POINTER IS REQUIRED, INSTEAD OF MERELY A POINTER TO THE TABLE			8
9	1454	/	ON PAGE ZERO ITSELF.			9
10	1455					10
11	1456	/	PIE CHIP PROGRAMMING TABLE.			11
12	1457					12
13	1458	1401	PTABLE= .	/PROGRAMMING TABLE HERE		13
14	1459					14
15	1460	/	FIRST CHIP.			15
16	1461					16
17	1462	071401* 1034	KEY^10+1004	/DEVICE 03		17
18	1463	071402* 1044	TTY^10+1004	/DEVICE 04		18
19	1464	071403* 3130	CLKCON^10+3000	/DEVICE 13		19
20	1465	071404* 3322	PRICON^10+3002	/DEVICE 32		20
21	1466	071405* 3337	PROCON^10+3007	/DEVICE 33		21
22	1467					22
23	1468	/	SECOND CHIP.			23
24	1469					24
25	1470	071406* 1302	PORTIN^10+1002	/DEVICE 30		25
26	1471	071407* 1313	PORTOUT^10+1003	/DEVICE 31		26
27	1472	071410* 3364	PORTCON^10+3004	/DEVICE 36		27
28	1473	071411* 0754	RXCON^10+0004	/DEVICE 75		28
29	1474	071412* 1144	APUCON^10+1004	/DEVICE 14		29
30	1475					30
31	1476	/	THIRD CHIP (CP INTERRUPTS ONLY).			31
32	1477					32
33	1478	071413* 1070	TTYCON^10+1000	/DEVICE 07		33
34	1479	071414* 3060	VIDINT^10+3000	/DEVICE 06		34
35	1480	071415* 7053	KBOCON^10+7003	/DEVICE 05		35
36	1481	071416* 7112	KBICON^10+7002	/DEVICE 11		36
37	1482	071417* 4124	VIDDEV^10+4004	/DEVICE 12		37
38	1483					38
39	1484	071420* 0000	0	/THIS ENDS THE LIST		39
40						40
41						41
42						42
43						43
44						44
45						45
46						46
47						47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	1486	/	MEMORY COMPARE ROUTINE.			1
2	1487					2
3	1488	/	THIS ROUTINE CHECKS IF THE CURRENT TEST FIELD MATCHES THE TEST PATTERN IN			3
4	1489	/	EVERY WORD. ALL MATCHING WORDS ARE INVERTED FOR THE NEXT TEST. THE IMMEDIATE			4
5	1490	/	RETURN IS TAKEN ON A MATCH FAILURE; SKIP RETURN IS TAKEN IF ALL WORDS MATCH.			5
6	1491					6
7	1492	071421* 0000	MEMCOMP, .-. .		/MEMORY COMPARE ROUTINE	7
8	1493	071422* 1044	TAD T2		/GET LATEST FIELD	8
9	1494	071423* 0104	AND Z7/[7]		/JUST FIELD BITS	9
10	1495	071424* 7014	R3L		/MOVE UP	10
11	1496	071425* 1122	TAD ZCDF/[CDF]		/FORM CDF INSTRUCTION	11
12	1497	071426* 3227	DCA .+1		/STORE IN-LINE	12
13	1498	071427* 7402	HLT+ .-. .		/WILL BE CDF XX INSTRUCTION	13
14	1499	071430* 1443	MCHKLP, TAD I T1		/GET A WORD	14
15	1500	071431* 1042	TAD T0		/COMPARE TO TEST VALUE	15
16	1501	071432* 7001	IAC		/SHOULD BE ZERO	16
17	1502	071433* 7640	SZA CLA		/SKIP IF IT MATCHES	17
18	1503	071434* 5621	JMP I MEMCOMPARE		/TAKE IMMEDIATE RETURN IF IT FAILS	18
19	1504	071435* 1042	TAD T0		/GET INVERTED TEST VALUE	19
20	1505	071436* 3443	DCA I T1		/STORE BACK	20
21	1506	071437* 1043	TAD T1		/GET ADDRESS	21
22	1507	071440* 1041	TAD TINC		/BUMP UP	22
23	1508	071441* 3043	DCA T1		/STORE BACK	23
24	1509	071442* 2010	ISZ XR0		/DONE ALL IN THIS FIELD	24
25	1510	071443* 5230	JMP MCHKLP		/NO, GO BACK FOR MORE	25
26	1511	071444* 2221	ISZ MEMCOMPARE		/YES, BUMP RETURN ADDRESS	26
27	1512	071445* 5621	JMP I MEMCOMPARE		/TAKE SKIP RETURN TO CALLER	27
28	1513					28
29	1514	071446* 1746	PROWTAB, ROWTABLE-1		/POINTER TO HIGH-ORDER ROW TABLE	29
30	1515	071447* 1663	PREGLIS, REGLIST-1		/POINTER TO REGISTER AND CONTENTS LIST	30
31	1516	071450* 3657	PROWADR, ROWADR-1		/POINTER TO ROWTABLE ADDRESS	31
32	1517					32
33	1518	071451* 0000	ROWLOAD, .-. .		/ROUTINE TO LOAD ROWTABLE AND VIDEO REGISTERS	33
34	1519	071452* 7300	CLA CLL		/CLEAN UP	34
35	1520	071453* 6126	VIDCON		/CLEAR SCREEN CONTROL REGISTER	35
36	1521	071454* 6271	CDF 70		/ENSURE OUR FIELD	36
37	1522	071455* 1250	TAD PROWADR/(ROWADR-1)		/GET ROWTABLE TABLE POINTER	37
38	1523	071456* 3010	DCA XR0		/STASH THE POINTER	38
39	1524	071457* 1246	TAD PROWTABLE/(ROWTABLE-1)		/GET HIGH-ORDER ROW ADDRESS TABLE VALUE	39
40	1525	071460* 3011	DCA XR1		/STASH THE POINTER	40
41	1526	071461* 1411	ROWLUP, TAD I XR1		/GET A LIST ELEMENT	41
42	1527	071462* 7510	SPA		/SKIP IF NOT AT END OF LIST	42
43	1528	071463* 5274	JMP FHERE		/JUMP IF AT END OF LIST	43
44	1529	071464* 7421	MQL		/STASH IT FOR NOW	44
45	1530	071465* 6211	CDF 10		/GOTO TABLE FIELD	45
46	1531	071466* 1153	TAD Z205/[205]		/GET LOW-ORDER BITS VALUE	46
47	1532	071467* 3410	DCA I XR0		/STASH IN TABLE	47
48	1533	071470* 7701	CLA MQA		/GET HIGH-ORDER AND ATTRIBUTES	48
49	1534	071471* 3410	DCA I XR0		/STASH IN TABLE	49
50	1535	071472* 6271	CDF 70		/BACK TO OUR FIELD	50
51	1536	071473* 5261	JMP ROWLUP		/KEEP GOING	51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	1593	071552*	2765	PCLR2, CLR2	/POINTER TO CLR2 ROUTINE	1
2	1594	071553*	1164	PTIMOUT, TIMOUT	/POINTER TO TIME-OUT ROUTINE	2
3	1595					3
4	1596	071554*	0000	FBLINK, .-. .	/BIG FLOPPY DISPLAY ROUTINE	4
5	1597	071555*	7315	NL0010 CLL	/SET EXTENDED CHARACTER MODE	5
6	1598	071556*	6126	VIDCON	/LOAD NEW CONTROL VALUE	6
7	1599	071557*	7200	CLA	/CLEAN UP	7
8	1600	071560*	1370	TAD BL40/(" &177)	/GET SPACE CODE	8
9	1601	071561*	4546	JMS I PSSTRING/[SSTRING]	/CLEAR ALL ITEMS OF	9
10	1602	071562*	2047	BIGFLPY	/THE BIG FLOPPY PATTERN	10
11	1603	071563*	4753	JMS I PTIMOUT/(TIMOUT)	/GO WAIT FOR AWHILE	11
12	1604	071564*	4546	JMS I PSSTRING/[SSTRING]	/GO DISPLAY ALL ITEMS OF	12
13	1605	071565*	2047	BIGFLPY	/THE BIG FLOPPY PATTERN	13
14	1606	071566*	4753	JMS I PTIMOUT/(TIMOUT)	/GO WAIT FOR AWHILE	14
15	1607	071567*	5754	JMP I FBLINK	/RETURN	15
16	1608					16
17	1609	071570*	0040	BL40, " &177	/CONSTANT 0040	17
18	1610					18
19	1611	071571*	0000	ZBLOCK 7	/EMPTY SPACE	19
20						20
21						21
22						22
23						23
24						24
25						25
26						26
27						27
28						28
29						29
30						30
31						31
32						32
33						33
34						34
35						35
36						36
37						37
38						38
39						39
40						40
41						41
42						42
43						43
44						44
45						45
46						46
47						47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	1651	/	THE REST OF THE TESTED BYTES ARE POINTED TO THE THIRD BYTE. THE FOUR BYTES				1
2	1652	/	SHOULD SUM TO 377. THE POINTER WAS INCREMENTED BY TWO INITIALLY, SO WE MAY				2
3	1653	/	HAVE TO SUBTRACT TWO TO LOCATE THE PROPER FOUR BYTE SEQUENCE.				3
4	1654						4
5	1655	071624* 3012	VALAGN, DCA	XR2	/SET TO 0000 (OR 7776)	5	
6	1656	071625* 7240		NL7777	/BACKUP	6	
7	1657	071626* 1011	TAD	XR1	/ADD ON POINTER	7	
8	1658	071627* 1012	TAD	XR2	/ADD 0000 OR 7776	8	
9	1659	071630* 3010	DCA	XR0	/STASH THE POINTER	9	
10	1660	071631* 6201	CDF	00	/GOTO THE DATA FIELD	10	
11	1661	071632* 1410	TAD I	XR0	/\	11	
12	1662	071633* 1410	TAD I	XR0	/ \GET THE NEXT	12	
13	1663	071634* 1410	TAD I	XR0	/ /FOUR BYTES	13	
14	1664	071635* 1410	TAD I	XR0	//	14	
15	1665	071636* 6271	CDF	70	/BACK TO OUR FIELD	15	
16	1666	071637* 7001	IAC		/ADD ONE	16	
17	1667	071640* 0115	AND	Z377/[377]	/JUST LOW-ORDER BITS	17	
18	1668	071641* 7640	SZA CLA		/SKIP IF ALL TOGETHER IT'S ZERO	18	
19	1669	071642* 5257	JMP	VALALT	/JUMP IF NOT	19	
20	1670						20
21	1671	/	HAVING LOCATED THE CORRECT FOUR BYTE SEQUENCE, CONFIRM THAT THE FIRST TWO HAVE				21
22	1672	/	BIT[8] SET.				22
23	1673						23
24	1674	071643* 7240		NL7777	/BACKUP	24	
25	1675	071644* 1011	TAD	XR1	/ADD ON POINTER	25	
26	1676	071645* 1012	TAD	XR2	/ADD ON 0000 OR 7776	26	
27	1677	071646* 3010	DCA	XR0	/STASH THE POINTER	27	
28	1678	071647* 6201	CDF	00	/GOTO THE DATA FIELD	28	
29	1679	071650* 7315	NL0010	CLL	/SET THE BIT MASK	29	
30	1680	071651* 0410	AND I	XR0	/GET THE FIRST ONE'S BIT	30	
31	1681	071652* 0410	AND I	XR0	/AND THE SECOND ONE'S BIT	31	
32	1682	071653* 6271	CDF	70	/BACK TO OUR FIELD	32	
33	1683	071654* 7640	SZA CLA		/SKIP IF OFF	33	
34	1684	071655* 2200	ISZ	VALID8	/BUMP TO GOOD RETURN	34	
35	1685	071656* 5600	JMP I	VALID8	/RETURN EITHER WAY	35	
36	1686						36
37	1687	071657* 1012	VALALT, TAD	XR2	/GET THE OFFSET	37	
38	1688	071660* 7640	SZA CLA		/SKIP IF ZERO	38	
39	1689	071661* 5600	JMP I	VALID8	/RETURN IF 7776 WAS ALSO UNSUCCESSFUL	39	
40	1690	071662* 7344		NL7776	/SET -2 VALUE	40	
41	1691	071663* 5224	JMP	VALAGN	/TRY IT AGAIN	41	
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	1693	/	THIS IS THE LIST OF VALUES USED WITH LSCREG AND VLOAD IN PAIRS.		1
2	1694				2
3	1695	071664* 0026	REGLIST,26	/REGISTER 16	3
4	1696	071665* 0000	000	/DUMMY VALUE (DO A RESET)	4
5	1697	071666* 0000	00	/REGISTER 0	5
6	1698	071667* 0143	144-1	/100 CHARACTERS PER HORIZONTAL PERIOD	6
7	1699	071670* 0001	01	/REGISTER 1	7
8	1700	071671* 0117	120-1	/80 CHARACTERS PER DATA ROW	8
9	1701	071672* 0002	02	/REGISTER 2	9
10	1702	071673* 0016	016	/14 IS HORIZONTAL DELAY	10
11	1703	071674* 0003	03	/REGISTER 3	11
12	1704	071675* 0007	007	/7 IS HORIZONTAL SYNC WIDTH	12
13	1705	071676* 0004	04	/REGISTER 4	13
14	1706	071677* 0003	003	/3 IS VERTICAL SYNC WIDTH	14
15	1707	071700* 0005	05	/REGISTER 5	15
16	1708	071701* 0025	025	/21 IS VERTICAL DELAY	16
17	1709	071702* 0006	06	/REGISTER 6	17
18	1710	071703* 0022	022	/PINS 00, CURSOR SKEW 2, BLANK SKEW 2	18
19	1711	071704* 0007	07	/REGISTER 7	19
20	1712	071705* 0027	30-1	/24 VISIBLE DATA ROWS PER FRAME	20
21	1713	071706* 0010	10	/REGISTER 8	21
22	1714	071707* 0051	051	/SCAN LINES/FRAME 1, SCAN LINES/DATA ROW 9	22
23	1715	071710* 0011	11	/REGISTER 9	23
24	1716	071711* 0011	011	/9 SCAN LINES PER FRAME	24
25	1717	071712* 0012	12	/REGISTER A	25
26	1718	071713* 0160	160	/DMA BURST DELAY 7, DMA BURST COUNT 0	26
27	1719	071714* 0013	13	/REGISTER B	27
28	1720	071715* 0010	010	/OPERATION MODE 4 ****	28
29	1721	071716* 0014	0014	/REGISTER C	29
30	1722	071717* 0260	ROWADR&377	/LOW-ORDER 8-BITS OF 13660	30
31	1723	071720* 0015	0015	/REGISTER D	31
32	1724	071721* 0227	ROWADR%400+0220	/CONTIGUOUS ROW TABLE AT 13660	32
33	1725	071722* 0016	16	/REGISTER E	33
34	1726	071723* 0000	000	/LOW-ORDER AUXILIARY ADDRESS REGISTER	34
35	1727	071724* 0017	17	/REGISTER F	35
36	1728	071725* 0000	000	/HIGH-ORDER AUXILIARY ADDRESS REGISTER	36
37	1729	071726* 0020	20	/REGISTER 10	37
38	1730	071727* 0077	077	/SEQUENTIAL BREAK REGISTER DISABLE VALUE	38
39	1731	071730* 0021	21	/REGISTER 11	39
40	1732	071731* 0000	000	/PAGE BLANK, SMOOTH SCROLL CAN START AT 000	40
41	1733	071732* 0022	22	/REGISTER 12	41
42	1734	071733* 0030	030	/DATA ROW END PAST LINE 23	42
43	1735	071734* 0027	27	/REGISTER 17	43
44	1736	071735* 0000	000	/NO SMOOTH SCROLL OFFSET	44
45	1737	071736* 0030	30	/REGISTER 18	45
46	1738	071737* 0000	000	/CURSOR APPEARS VERTICAL AT 000	46
47	1739	071740* 0031	31	/REGISTER 19	47
48	1740	071741* 0000	000	/CURSOR APPEARS HORIZONTALLY AT 000	48
49	1741	071742* 0032	32	/REGISTER 1A	49
50	1742	071743* 0100	100	/ENABLE VERTICAL RETRACE INTERRUPTS	50
51	1743	071744* 0025	25	/REGISTER 15	51
52	1744	071745* 0000	000	/ISSUE START COMMAND	52
53	1745	071746* 4000	4000	/THIS ENDS THE LIST	53
54					54
55					55
56					56
57					57
58					58

1	1747	/	THIS TABLE IS USED TO LOAD THE HIGH-ORDER BYTES OF THE CONTIGUOUS ROW ADDRESS				1
2	1748	/	TABLE. THE FORMAT IS TWO ATTRIBUTE BITS AND SIX ADDRESS BITS. THE LOW-ORDER				2
3	1749	/	ADDRESS BYTE IS ALWAYS 205. THE ATTRIBUTE BITS ARE: 00=NORMAL DATA ROW,				3
4	1750	/	01=SINGLE HEIGHT, DOUBLE WIDTH; 10=TOP OF DOUBLE HEIGHT, DOUBLE WIDTH;				4
5	1751	/	11=BOTTOM OF DOUBLE HEIGHT, DOUBLE WIDTH.				5
6	1752						6
7	1753	071747* 0141	ROWTABL,0141	/LINE 00	DATA STARTS AT 20605 (DOUBLE WIDTH)	7	
8	1754	071750* 0042	0042	/LINE 01	DATA STARTS AT 21205	8	
9	1755	071751* 0043	0043	/LINE 02	DATA STARTS AT 21605	9	
10	1756	071752* 0044	0044	/LINE 03	DATA STARTS AT 22205	10	
11	1757	071753* 0045	0045	/LINE 04	DATA STARTS AT 22605	11	
12	1758	071754* 0046	0046	/LINE 05	DATA STARTS AT 23205	12	
13	1759	071755* 0047	0047	/LINE 06	DATA STARTS AT 23605	13	
14	1760	071756* 0050	0050	/LINE 07	DATA STARTS AT 24205	14	
15	1761	071757* 0051	0051	/LINE 08	DATA STARTS AT 24605	15	
16	1762	071760* 0052	0052	/LINE 09	DATA STARTS AT 25205	16	
17	1763	071761* 0053	0053	/LINE 10	DATA STARTS AT 25605	17	
18	1764	071762* 0054	0054	/LINE 11	DATA STARTS AT 26205	18	
19	1765	071763* 0055	0055	/LINE 12	DATA STARTS AT 26605	19	
20	1766	071764* 0056	0056	/LINE 13	DATA STARTS AT 27205	20	
21	1767	071765* 0057	0057	/LINE 14	DATA STARTS AT 27605	21	
22	1768	071766* 0072	0072	/LINE 15	DATA STARTS AT 35205	22	
23	1769	071767* 0061	0061	/LINE 16	DATA STARTS AT 30605	23	
24	1770	071770* 0062	0062	/LINE 17	DATA STARTS AT 31205	24	
25	1771	071771* 0063	0063	/LINE 18	DATA STARTS AT 31605	25	
26	1772	071772* 0064	0064	/LINE 19	DATA STARTS AT 32205	26	
27	1773	071773* 0065	0065	/LINE 20	DATA STARTS AT 32605	27	
28	1774	071774* 0066	0066	/LINE 21	DATA STARTS AT 33205	28	
29	1775	071775* 0067	0067	/LINE 22	DATA STARTS AT 33605	29	
30	1776	071776* 0070	0070	/LINE 23	DATA STARTS AT 34205	30	
31	1777	071777* 7777	7777	/THIS	ENDS THE LIST	31	
32						32	
33						33	
34						34	
35						35	
36						36	
37						37	
38						38	
39						39	
40						40	
41						41	
42						42	
43						43	
44						44	
45						45	
46						46	
47						47	
48						48	
49						49	
50						50	
51						51	
52						52	
53						53	
54						54	
55						55	
56						56	
57						57	
58						58	

1	1834		/	DATA PATTERN THAT LOOKS LIKE A BIG FLOPPY (USED FOR ERROR DISPLAY).		1
2	1835					2
3	1836	072047*	0000	BIGFLPY,0000	/FIELD 2	3
4	1837	072050*	1445	1445	/ADDRESS	4
5	1838	072051*	0015	0015		5
6	1839	072052*	0022	0022		6
7	1840	072053*	0022	0022		7
8	1841	072054*	0022	0022		8
9	1842	072055*	0022	0022		9
10	1843	072056*	0022	0022		10
11	1844	072057*	0022	0022		11
12	1845	072060*	0022	0022		12
13	1846	072061*	0022	0022		13
14	1847	072062*	0022	0022		14
15	1848	072063*	0022	0022		15
16	1849	072064*	0022	0022		16
17	1850	072065*	0022	0022		17
18	1851	072066*	0022	0022		18
19	1852	072067*	0022	0022		19
20	1853	072070*	0022	0022		20
21	1854	072071*	0022	0022		21
22	1855	072072*	0022	0022		22
23	1856	072073*	0022	0022		23
24	1857	072074*	0022	0022		24
25	1858	072075*	0022	0022		25
26	1859	072076*	0022	0022		26
27	1860	072077*	0022	0022		27
28	1861	072100*	0022	0022		28
29	1862	072101*	0022	0022		29
30	1863	072102*	0022	0022		30
31	1864	072103*	0014	0014		31
32	1865	072104*	0000	0000	/FIELD 2	32
33	1866	072105*	2045	2045	/ADDRESS	33
34	1867	072106*	0031	0031		34
35	1868	072107*	0000	0000	/FIELD 2	35
36	1869	072110*	2070	2070	/ADDRESS	36
37	1870	072111*	0015	0015		37
38	1871	072112*	0022	0022		38
39	1872	072113*	0022	0022		39
40	1873	072114*	0022	0022		40
41	1874	072115*	0022	0022		41
42	1875	072116*	0022	0022		42
43	1876	072117*	0014	0014		43
44	1877	072120*	0031	0031		44
45	1878	072121*	0000	0000	/FIELD 2	45
46	1879	072122*	2445	2445	/ADDRESS	46
47	1880	072123*	0031	0031		47
48	1881	072124*	0000	0000	/FIELD 2	48
49	1882	072125*	2470	2470	/ADDRESS	49
50	1883	072126*	0031	0031		50
51	1884	072127*	0000	0000	/FIELD 2	51
52	1885	072130*	2476	2476	/ADDRESS	52
53	1886	072131*	0031	0031		53
54	1887	072132*	0031	0031		54
55						55
56						56
57						57
58						58

1	1889	072133*	0000	0000	/FIELD 2	1
2	1890	072134*	3045	3045	/ADDRESS	2
3	1891	072135*	0031	0031		3
4	1892	072136*	0000	0000	/FIELD 2	4
5	1893	072137*	3070	3070	/ADDRESS	5
6	1894	072140*	0031	0031		6
7	1895	072141*	0000	0000	/FIELD 2	7
8	1896	072142*	3076	3076	/ADDRESS	8
9	1897	072143*	0031	0031		9
10	1898	072144*	0031	0031		10
11	1899	072145*	0000	0000	/FIELD 2	11
12	1900	072146*	3445	3445	/ADDRESS	12
13	1901	072147*	0031	0031		13
14	1902	072150*	0000	0000	/FIELD 2	14
15	1903	072151*	3462	3462	/ADDRESS	15
16	1904	072152*	4025	4025		16
17	1905	072153*	0000	0000	/FIELD 2	17
18	1906	072154*	3470	3470	/ADDRESS	18
19	1907	072155*	0016	0016		19
20	1908	072156*	0022	0022		20
21	1909	072157*	0022	0022		21
22	1910	072160*	0022	0022		22
23	1911	072161*	0022	0022		23
24	1912	072162*	0022	0022		24
25	1913	072163*	0013	0013		25
26	1914	072164*	0031	0031		26
27	1915	072165*	0000	0000	/FIELD 2	27
28	1916	072166*	4045	4045	/ADDRESS	28
29	1917	072167*	0031	0031		29
30	1918	072170*	0000	0000	/FIELD 2	30
31	1919	072171*	4060	4060	/ADDRESS	31
32	1920	072172*	4026	4026		32
33	1921	072173*	4031	4031		33
34	1922	072174*	0000	0000	/FIELD 2	34
35	1923	072175*	4063	4063	/ADDRESS	35
36	1924	072176*	4031	4031		36
37	1925	072177*	4026	4026		37
38	1926	072200*	0000	0000	/FIELD 2	38
39	1927	072201*	4072	4072	/ADDRESS	39
40	1928	072202*	0015	0015		40
41	1929	072203*	0022	0022		41
42	1930	072204*	0022	0022		42
43	1931	072205*	0022	0022		43
44	1932	072206*	0014	0014		44
45	1933	072207*	0031	0031		45
46	1934	072210*	0000	0000	/FIELD 2	46
47	1935	072211*	4445	4445	/ADDRESS	47
48	1936	072212*	0031	0031		48
49	1937	072213*	0050	0050		49
50	1938	072214*	0440	" +200		50
51	1939	072215*	0440	" +200		51
52	1940	072216*	0440	" +200		52
53	1941	072217*	0440	" +200		53
54	1942	072220*	0440	" +200		54
55	1943	072221*	0051	0051		55
56						56
57						57
58						58

1	1945	072222*	0000	0000	/FIELD 2	1
2	1946	072223*	4460	4460	/ADDRESS	2
3	1947	072224*	4027	4027		3
4	1948	072225*	0000	0000	/FIELD 2	4
5	1949	072226*	4464	4464	/ADDRESS	5
6	1950	072227*	4027	4027		6
7	1951	072230*	0000	0000	/FIELD 2	7
8	1952	072231*	4472	4472	/ADDRESS	8
9	1953	072232*	0031	0031		9
10	1954	072233*	0000	0000	/FIELD 2	10
11	1955	072234*	4476	4476	/ADDRESS	11
12	1956	072235*	0031	0031		12
13	1957	072236*	0031	0031		13
14	1958	072237*	0000	0000	/FIELD 2	14
15	1959	072240*	5045	5045	/ADDRESS	15
16	1960	072241*	0031	0031		16
17	1961	072242*	0000	0000	/FIELD 2	17
18	1962	072243*	5061	5061	/ADDRESS	18
19	1963	072244*	4030	4030		19
20	1964	072245*	4027	4027		20
21	1965	072246*	4030	4030		21
22	1966	072247*	0000	0000	/FIELD 2	22
23	1967	072250*	5072	5072	/ADDRESS	23
24	1968	072251*	0031	0031		24
25	1969	072252*	0000	0000	/FIELD 2	25
26	1970	072253*	5076	5076	/ADDRESS	26
27	1971	072254*	0031	0031		27
28	1972	072255*	0031	0031		28
29	1973	072256*	0000	0000	/FIELD 2	29
30	1974	072257*	5445	5445	/ADDRESS	30
31	1975	072260*	0031	0031		31
32	1976	072261*	0000	0000	/FIELD 2	32
33	1977	072262*	5460	5460	/ADDRESS	33
34	1978	072263*	0007	0007		34
35	1979	072264*	0000	0000	/FIELD 2	35
36	1980	072265*	5472	5472	/ADDRESS	36
37	1981	072266*	0031	0031		37
38	1982	072267*	0000	0000	/FIELD 2	38
39	1983	072270*	5476	5476	/ADDRESS	39
40	1984	072271*	0031	0031		40
41	1985	072272*	0031	0031		41
42	1986	072273*	0000	0000	/FIELD 2	42
43	1987	072274*	6045	6045	/ADDRESS	43
44	1988	072275*	0031	0031		44
45	1989	072276*	0000	0000	/FIELD 2	45
46	1990	072277*	6072	6072	/ADDRESS	46
47	1991	072300*	0031	0031		47
48	1992	072301*	0000	0000	/FIELD 2	48
49	1993	072302*	6076	6076	/ADDRESS	49
50	1994	072303*	0031	0031		50
51	1995	072304*	0031	0031		51
52	1996	072305*	0000	0000	/FIELD 2	52
53	1997	072306*	6445	6445	/ADDRESS	53
54	1998	072307*	0031	0031		54
55						55
56						56
57						57
58						58

1	2000	072310*	0000	0000	/FIELD 2	1
2	2001	072311*	6472	6472	/ADDRESS	2
3	2002	072312*	0031	0031		3
4	2003	072313*	0000	0000	/FIELD 2	4
5	2004	072314*	6476	6476	/ADDRESS	5
6	2005	072315*	0031	0031		6
7	2006	072316*	0031	0031		7
8	2007	072317*	0000	0000	/FIELD 2	8
9	2008	072320*	7045	7045	/ADDRESS	9
10	2009	072321*	0031	0031		10
11	2010	072322*	0000	0000	/FIELD 2	11
12	2011	072323*	7070	7070	/ADDRESS	12
13	2012	072324*	0015	0015		13
14	2013	072325*	0014	0014		14
15	2014	072326*	0016	0016		15
16	2015	072327*	0022	0022		16
17	2016	072330*	0022	0022		17
18	2017	072331*	0022	0022		18
19	2018	072332*	0013	0013		19
20	2019	072333*	0031	0031		20
21	2020	072334*	0000	0000	/FIELD 2	21
22	2021	072335*	7445	7445	/ADDRESS	22
23	2022	072336*	0016	0016		23
24	2023	072337*	0022	0022		24
25	2024	072340*	0022	0022		25
26	2025	072341*	0022	0022		26
27	2026	072342*	0022	0022		27
28	2027	072343*	0022	0022		28
29	2028	072344*	0022	0022		29
30	2029	072345*	0022	0022		30
31	2030	072346*	0022	0022		31
32	2031	072347*	0022	0022		32
33	2032	072350*	0022	0022		33
34	2033	072351*	0022	0022		34
35	2034	072352*	0022	0022		35
36	2035	072353*	0022	0022		36
37	2036	072354*	0022	0022		37
38	2037	072355*	0022	0022		38
39	2038	072356*	0022	0022		39
40	2039	072357*	0022	0022		40
41	2040	072360*	0022	0022		41
42	2041	072361*	0013	0013		42
43	2042	072362*	0016	0016		43
44	2043	072363*	0022	0022		44
45	2044	072364*	0022	0022		45
46	2045	072365*	0022	0022		46
47	2046	072366*	0022	0022		47
48	2047	072367*	0022	0022		48
49	2048	072370*	0013	0013		49
50	2049	072371*	7777	7777	/END OF LIST	50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	2051		/	ERROR NUMBER DISPLAY LIST.		1
2	2052					2
3	2053	072372*	0000	EDISPLI,0000	/FIELD 2	3
4	2054	072373*	5422	5422	/ADDRESS	4
5	2055	072374*	0000	.-.	/HIGH-ORDER ERROR DIGIT	5
6	2056	072375*	0000	.-.	/LOW-ORDER ERROR DIGIT	6
7	2057	072376*	0000	0000	/FIELD 2	7
8	2058	072377*	6022	6022	/ADDRESS	8
9	2059	072400*	0040	J00, 0040+.-.	/HIGH-ORDER ERROR DIGIT	9
10	2060	072401*	0040	J01, 0040+.-.	/LOW-ORDER ERROR DIGIT	10
11	2061	072402*	7777	7777	/END OF LIST	11
12	2062					12
13	2063		/	NON-ERROR (DECMATE II LOGO) DISPLAY LIST.		13
14	2064					14
15	2065	072403*	0000	DMLOGO, 0000	/FIELD 2	15
16	2066	072404*	5415	5415	/ADDRESS	16
17	2067	072405*	0440	" +200		17
18	2068	072406*	0504	"D+200		18
19	2069	072407*	0505	"E+200		19
20	2070	072410*	0503	"C+200		20
21	2071	072411*	0555	"M+240		21
22	2072	072412*	0541	"A+240		22
23	2073	072413*	0564	"T+240		23
24	2074	072414*	0545	"E+240		24
25	2075	072415*	0440	" +200		25
26	2076	072416*	0511	"I+200		26
27	2077	072417*	0511	"I+200		27
28	2078	072420*	0440	" +200		28
29	2079	072421*	0000	0000	/FIELD 2	29
30	2080	072422*	6015	6015	/ADDRESS	30
31	2081	072423*	0440	" +200		31
32	2082	072424*	0504	"D+200		32
33	2083	072425*	0505	"E+200		33
34	2084	072426*	0503	"C+200		34
35	2085	072427*	0555	"M+240		35
36	2086	072430*	0541	"A+240		36
37	2087	072431*	0564	"T+240		37
38	2088	072432*	0545	"E+240		38
39	2089	072433*	0440	" +200		39
40	2090	072434*	0511	"I+200		40
41	2091	072435*	0511	"I+200		41
42	2092	072436*	0440	" +200		42
43	2093	072437*	7777	7777	/END OF LIST	43
44	2094					44
45	2095	072440*	0023	L23, 23	/CONSTANT 0023	45
46	2096	072441*	0247	L247, 247	/CONSTANT 0247	46
47	2097	072442*	0202	L202, 202	/CONSTANT 0202	47
48	2098	072443*	0460	L460, "0+200	/CONSTANT 0460	48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	2100	072444*	0000	ENDUP,	.-.	/FINISH UP ROUTINE	1
2	2101	072445*	7300		CLA CLL	/CLEAN UP	2
3	2102	072446*	4730		JMS I PSET2/(SET2)	/SET ATTRIBUTE BITS IN 13705, 13707	3
4	2103	072447*	1054		TAD ESTATUS	/GET ERROR STATUS	4
5	2104	072450*	7650		SNA CLA	/SKIP IF ANY PROBLEMS	5
6	2105	072451*	5315		JMP NOERRORS	/JUMP IF NONE	6
7	2106	072452*	3010		DCA XR0	/CLEAR QUOTIENT	7
8	2107	072453*	1054		TAD ESTATUS	/GET ERROR STATUS	8
9	2108	072454*	6121		APTSKP	/AUTOMATIC PRODUCT TEST?	9
10	2109	072455*	7410		SKP	/SKIP IF NOT	10
11	2110	072456*	5735		JMP I PAPT2/(APTRO2)	/ELSE GO THERE	11
12	2111	072457*	1126	SUBLUP,	TAD Z7766/[7766]	/SUBTRACT 10	12
13	2112	072460*	7510		SPA	/GONE TOO FAR?	13
14	2113	072461*	5264		JMP ADDFIX	/YES, GO THERE TO FIX UP	14
15	2114	072462*	2010		ISZ XR0	/NO, BUMP QUOTIENT	15
16	2115	072463*	5257		JMP SUBLUP	/KEEP GOING	16
17	2116						17
18	2117	072464*	1106	ADDFIX,	TAD Z12/[12]	/RESTORE VALUE	18
19	2118	072465*	1243		TAD L460/("0+200)	/MAKE IT ASCII WITH ATTRIBUTES SET	19
20	2119	072466*	3731		DCA I LOW1/(EDISPLIST+3)	/STORE IN BUFFER	20
21	2120	072467*	1731		TAD I LOW1/(EDISPLIST+3)	/GET IT BACK	21
22	2121	072470*	3732		DCA I LOW2/(EDISPLIST+7)	/STORE THERE ALSO	22
23	2122	072471*	1010		TAD XR0	/GET QUOTIENT	23
24	2123	072472*	1243		TAD L460/("0+200)	/MAKE IT ASCII WITH ATTRIBUTES SET	24
25	2124	072473*	3733		DCA I HIGH1/(EDISPLIST+2)	/STORE IN BUFFER	25
26	2125	072474*	1733		TAD I HIGH1/(EDISPLIST+2)	/GET IT BACK	26
27	2126	072475*	3734		DCA I HIGH2/(EDISPLIST+6)	/STORE THERE ALSO	27
28	2127	072476*	1240		TAD L23/(23)	/GET KEYBOARD CODE TO LIGHT LEDS	28
29	2128	072477*	6056		KBOLS	/SEND IT TO KEYBOARD	29
30	2129	072500*	3012		DCA XR2	/CLEAR WAIT COUNTER	30
31	2130	072501*	2012		ISZ XR2	/WAITING LONG ENOUGH?	31
32	2131	072502*	5301		JMP .-1	/NO, KEEP GOING	32
33	2132	072503*	1242		TAD L202/(202)	/GET KEYBOARD CODE FOR LED PATTERN	33
34	2133	072504*	6056		KBOLS	/SEND IT TO KEYBOARD	34
35	2134	072505*	7200		CLA	/CLEAN UP	35
36	2135	072506*	4546		JMS I PSSTRING/[SSTRING]	/CALL STRING STORE FOR DISPLAY ROUTINE	36
37	2136	072507*	2372		EDISPLIST	/ERROR DIGITS DISPLAY LIST POINTER	37
38	2137	072510*	6111		KBISF	/KEYBOARD FLAG UP?	38
39	2138	072511*	5310		JMP .-1	/NO, WAIT FOR IT	39
40	2139	072512*	6116		KBIRB	/YES, READ IN THE CHARACTER	40
41	2140	072513*	7200		CLA	/THROW IT AWAY	41
42	2141	072514*	5644		JMP I ENDUP	/RETURN	42
43	2142						43
44	2143			/	COMES HERE IF NO ERRORS FOUND EARLIER.		44
45	2144						45
46	2145	072515*	6121	NOERROR,	APTSKP	/AUTOMATIC PRODUCT TESTING?	46
47	2146	072516*	5322		JMP NOTAPT	/JUMP IF NOT	47
48	2147	072517*	4524		JMS I PAPROUTINE/[APROUTINE]	/CALL ROUTINE IF SO	48
49	2148	072520*	5721		JMP I .+1/(PRGDONE)	/GO THERE	49
50	2149	072521*	0214		PRGDONE	/THROUGH HERE	50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2151	072522*	1241	NOTAPT, TAD	L247/(247)	/GET KEYBOARD CODE FOR THE BEEPER	1
2	2152	072523*	6056	KBOLS		/SEND IT	2
3	2153	072524*	7200	CLA		/CLEAN UP	3
4	2154	072525*	4546	JMS I	PSSTRING/[SSTRING]	/DISPLAY THE	4
5	2155	072526*	2403	DMLOGO		/DECIMATE II LOGO	5
6	2156	072527*	5644	JMP I	ENDUP	/RETURN	6
7	2157						7
8	2158	072530*	2747	PSET2, SET2		/POINTER TO 2-WORD ATTRIBUTE BITS SET ROUTINE	8
9	2159	072531*	2375	LOW1, EDISPLIST+3		/POINTER TO LOW-ORDER DIGIT STORAGE (1)	9
10	2160	072532*	2401	LOW2, EDISPLIST+7		/POINTER TO LOW-ORDER DIGIT STORAGE (2)	10
11	2161	072533*	2374	HIGH1, EDISPLIST+2		/POINTER TO HIGH-ORDER DIGIT STORAGE (1)	11
12	2162	072534*	2400	HIGH2, EDISPLIST+6		/POINTER TO HIGH-ORDER DIGIT STORAGE (2)	12
13	2163	072535*	6520	PAPT2, APTRO2		/POINTER TO APTROUTINE ERROR HANDLING ENTRY	13
14	2164						14
15	2165			/	CP-INTERRUPT HANDLER FOR DOWN-LOADED CODE. THE DOWNLOADED CODE EXITS VIA A		15
16	2166			/	HLT INSTRUCTION OR PANEL REQUEST. HLT ENDS THE EXECUTION AND RETURNS CONTROL		16
17	2167			/	TO OUR CODE; PANEL REQUEST CAUSES A TOTAL RESTART OF ALL MOVED ROM CODE.		17
18	2168						18
19	2169	072536*	3017	CPIHND, DCA	XR7	/SAVE AC	19
20	2170	072537*	6256	GCF		/GET CURRENT FIELDS, ETC. STATUS	20
21	2171	072540*	3016	DCA	XR6	/STASH IT	21
22	2172	072541*	6000	PRS		/GET PANEL STATUS	22
23	2173	072542*	3010	DCA	XR0	/STASH IT	23
24	2174	072543*	6276	SPD		/CP INDIRECTS FROM NOW ON	24
25	2175	072544*	1010	TAD	XR0	/GET STATUS BACK	25
26	2176	072545*	7500	SMA		/SKIP IF BOOTSTRAP (CPREQ) SET	26
27	2177	072546*	5365	JMP	HALTEST	/JUMP IF NOT	27
28	2178	072547*	7200	CLA		/CLEAN UP	28
29	2179	072550*	6061	VSF		/SKIP ON, CLEAR VIDEO INTERRUPT FLAG	29
30	2180	072551*	7000	NOP		/IN CASE IT SKIPS	30
31	2181	072552*	6071	TTYSF		/SKIP ON, CLEAR TTY: TRAP FLAG	31
32	2182	072553*	7000	NOP		/IN CASE IT SKIPS	32
33	2183	072554*	6111	KBISF		/SKIP ON, CLEAR KEYBOARD INPUT FLAG	33
34	2184	072555*	7000	NOP		/IN CASE IT SKIPS	34
35	2185	072556*	6051	KBOSF		/SKIP ON, CLEAR KEYBOARD OUTPUT FLAG	35
36	2186	072557*	7000	NOP		/IN CASE IT SKIPS	36
37	2187	072560*	1016	TAD	XR6	/GET FIELDS BACK	37
38	2188	072561*	6005	RTF		/RESTORE THEM	38
39	2189	072562*	1017	TAD	XR7	/RESTORE AC	39
40	2190	072563*	6004	PEX		/LEAVING PANEL MEMORY	40
41	2191	072564*	5400	JMP I	INTADR	/RETURN TO INTERRUPTED MAIN MEMORY PROGRAM	41
42	2192						42
43	2193	072565*	6273	HALTEST, CIF	CDF 70	/STAYING IN OUR FIELD	43
44	2194	072566*	0374	AND	L200/(200)	/JUST HLTFLG BIT	44
45	2195	072567*	7650	SNA CLA		/SKIP IF SET	45
46	2196	072570*	5774	JMP I	L200/(CODESTART)	/JUMP IF NOT (PANEL REQUEST)	46
47	2197	072571*	6003	PGO		/RESET RUN FLIP-FLOP	47
48	2198	072572*	5773	JMP I	+.1/(DWNENTRY)	/GO THERE	48
49	2199	072573*	0650	DWNENTRY		/THROUGH HERE	49
50	2200						50
51	2201	072574*	0200	L200, 200	/CODESTART	/CONSTANT 0200	51
52	2202						52
53	2203	072575*	0000	0000		/THIS ENDS THE LIST OF INSTRUCTIONS TO MOVE	53
54	2204						54
55	2205	072576*	0000	ZBLOCK 2		/EMPTY SPACE	55
56							56
57							57
58							58

1	2307	/	COMES HERE WHEN NULL JOB TIMES OUT.		1
2	2308				2
3	2309	070321* 7344	TIMCHK, NL7776	/SETUP VALUE	3
4	2310	070322* 1004	TAD CLKTICK	/COMPARE TO CLOCK TICK COUNT	4
5	2311	070323* 7710	SPA CLA	/SKIP IF >=2	5
6	2312	070324* 7305	NL002A	/SET STATUS BIT IF NOT	6
7	2313	070325* 1005	TAD CSTATUS	/GET COUNT STATUS EITHER WAY	7
8	2314	070326* 3005	DCA CSTATUS	/STORE BACK	8
9	2315	070327* 6007	CAF	/CLEAR ALL FLAGS NOW	9
10	2316	070330* 1005	TAD CSTATUS	/GET COUNT STATUS	10
11	2317	070331* 1007	TAD TSTATUS	/GET CP TEST STATUS	11
12	2318	070332* 6236	PRQ3	/DO A PANEL REQUEST TO EXIT	12
13	2319	070333* 5333	JMP .	/WILL HANG HERE IF IT FAILS!	13
14	2320				14
15	2321	070334* 0000	COUNTUP, .-. .	/PATTERN COUNT ROUTINE	15
16	2322	070335* 3006	DCA PATTERN	/SAVE PASSED VALUE	16
17	2323	070336* 1006	TAD PATTERN	/GET IT BACK	17
18	2324	070337* 7001	IAC	/ADD ONE	18
19	2325	070340* 0246	AND L377/(377)	/JUST GOOD BITS	19
20	2326	070341* 7450	SNA	/SKIP IF NOT AT END	20
21	2327	070342* 5734	JMP I COUNTUP	/TAKE IMMEDIATE RETURN IF AT END	21
22	2328	070343* 3006	DCA PATTERN	/SAVE UPDATED VALUE	22
23	2329	070344* 2334	ISZ COUNTUP	/BUMP RETURN ADDRESS	23
24	2330	070345* 1006	TAD PATTERN	/GET NEW PATTERN VALUE	24
25	2331	070346* 5734	JMP I COUNTUP	/TAKE SKIP RETURN TO CALLER	25
26	2332				26
27	2333	0347	LBTEND= .	/END OF CODE TO BE MOVED	27
28	2334				28
29	2335	*2747	RELOC LBTEND-LBTEST+LBLOAD	/RE-ESTABLISH ORIGIN	29
30	2336				30
31	2337	/	SUBROUTINE TO SET HIGH-ORDER BITS IN CP 13705 AND 13707. BIT[4] IS SET IN		31
32	2338	/	13705; BITS[4-5] ARE SET IN 13707. THIS MAKES THE LINES INTO DOUBLE-HEIGHT,		32
33	2339	/	DOUBLE WIDTH UPPER AND LOWER HALVES RESPECTIVELY. THIS ROUTINE DEPENDS ON THE		33
34	2340	/	PREVIOUS CONTENTS OF THE BITS BEING ZERO.		34
35	2341				35
36	2342	072747* 0000	SET2, .-. .	/SET HIGH BITS IN 2 WORDS ROUTINE	36
37	2343	072750* 6211	CDF 10	/GOTO BUFFER FIELD	37
38	2344	072751* 1761	TAD I L3705/(3705)	/GET CURRENT CONTENTS OF 13705	38
39	2345	072752* 1112	TAD Z200/[200]	/ADD ON A HIGH-ORDER BIT	39
40	2346	072753* 3761	DCA I L3705/(3705)	/STORE BACK	40
41	2347	072754* 1762	TAD I L3707/(3707)	/GET CURRENT CONTENTS OF 13707	41
42	2348	072755* 1363	TAD L300/(300)	/ADD ON HIGH-ORDER BITS	42
43	2349	072756* 3762	DCA I L3707/(3707)	/STORE BACK	43
44	2350	072757* 6271	CDF 70	/BACK TO OUR FIELD	44
45	2351	072760* 5747	JMP I SET2	/RETURN	45
46	2352				46
47	2353	072761* 3705	L3705, 3705	/CONSTANT 3705	47
48	2354	072762* 3707	L3707, 3707	/CONSTANT 3707	48
49	2355	072763* 0300	L300, 300	/CONSTANT 0300	49
50	2356	072764* 0077	L77, 77	/CONSTANT 0077	50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	2358	/	SUBROUTINE TO CLEAR THE ATTRIBUTE BITS OF CP 13705 AND 13707. THIS MAKES THE				1
2	2359	/	CORRESPONDING LINES NORMAL WIDTH AND HEIGHT.				2
3	2360						3
4	2361	072765* 0000	CLR2,	.-.		/CLEAR BITS IN TWO WORDS ROUTINE.	4
5	2362	072766* 6211		CDF	10	/GOTO BUFFER FIELD	5
6	2363	072767* 1761		TAD I	L3705/(3705)	/GET CURRENT CONTENTS OF 13705	6
7	2364	072770* 0364		AND	L77/(77)	/JUST KEEP LOW-ORDER HALF	7
8	2365	072771* 3761		DCA I	L3705/(3705)	/STORE BACK	8
9	2366	072772* 1762		TAD I	L3707/(3707)	/GET CURRENT CONTENTS OF 13707	9
10	2367	072773* 0364		AND	L77/(77)	/JUST KEEP LOW-ORDER HALF	10
11	2368	072774* 3762		DCA I	L3707/(3707)	/STORE BACK	11
12	2369	072775* 6271		CDF	70	/BACK TO OUR FIELD	12
13	2370	072776* 5765		JMP I	CLR2	/RETURN	13
14	2371						14
15	2372	072777* 0000		ZBLOCK	1	/EMPTY SPACE	15
16							16
17							17
18							18
19							19
20							20
21							21
22							22
23							23
24							24
25							25
26							26
27							27
28							28
29							29
30							30
31							31
32							32
33							33
34							34
35							35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2421	074034*	6071	TSTTRAP,TTYSF		/DEVICE 03, 04 TRAP FLAG SET?	1
2	2422	074035*	5245	JMP	TSTKYIN	/JUMP IF NOT	2
3	2423	074036*	6075	TTYIE		/PREVENT FURTHER TTY: TRAP INTERRUPTS	3
4	2424	074037*	1305	TAD	PTTYPC/(TTYPC)	/GET EXPECTED INTERRUPTED ADDRESS	4
5	2425	074040*	7041	CIA		/INVERT FOR COMPARISON	5
6	2426	074041*	1000	TAD	INTADR	/COMPARE TO ACTUAL INTERRUPT PC	6
7	2427	074042*	7650	SNA CLA		/SKIP IF OTHER THAN EXPECTED	7
8	2428	074043*	5265	JMP	BUMPIT	/JUMP IF OK	8
9	2429						9
10	2430			/	INTERRUPT PC WRONG.		10
11	2431						11
12	2432	074044*	5266	JMP	JUSTEXIT	/CONTINUE THERE	12
13	2433						13
14	2434	074045*	6111	TSTKYIN,KBISF		/KEYBOARD INPUT FLAG SET?	14
15	2435	074046*	5256	JMP	TSTKYOUT	/NO, TRY OUTPUT	15
16	2436	074047*	6115	KBIIE		/PREVENT FURTHER KEYBOARD INPUT INTERRUPTS	16
17	2437	074050*	1306	TAD	PKBIPC/(KBIPC)	/GET EXPECTED INTERRUPTED ADDRESS	17
18	2438	074051*	7041	CIA		/INVERT FOR COMPARISON	18
19	2439	074052*	1000	TAD	INTADR	/COMPARE TO ACTUAL INTERRUPT PC	19
20	2440	074053*	7650	SNA CLA		/SKIP IF OTHER THAN EXPECTED	20
21	2441	074054*	5265	JMP	BUMPIT	/JUMP IF OK	21
22	2442						22
23	2443			/	INTERRUPT PC WRONG.		23
24	2444						24
25	2445	074055*	5266	JMP	JUSTEXIT	/CONTINUE THERE	25
26	2446						26
27	2447	074056*	6051	TSTKYOU,KBOSF		/KEYBOARD OUTPUT FLAG SET?	27
28	2448	074057*	5266	JMP	JUSTEXIT	/NO, FORGET IT	28
29	2449	074060*	6055	KBOIE		/PREVENT FURTHER KEYBOARD OUTPUT INTERRUPTS	29
30	2450	074061*	1307	TAD	PKBOIPC/(KBOIPC)	/GET EXPECTED INTERRUPTED ADDRESS	30
31	2451	074062*	7041	CIA		/INVERT FOR COMPARISON	31
32	2452	074063*	1000	TAD	INTADR	/COMPARE TO ACTUAL INTERRUPT PC	32
33	2453	074064*	7650	SNA CLA		/SKIP IF OTHER THAN EXPECTED	33
34	2454						34
35	2455			/	COMES HERE TO BUMP THE PC PAST THE NL0004 INSTRUCTIONS IF THE PROPER INTERRUPT		35
36	2456			/	ADDRESS WAS PRESENT AT THE POINT OF INTERRUPT.		36
37	2457						37
38	2458	074065*	2000	BUMPIT, ISZ	INTADR	/BUMP INTERRUPTED ADDRESS	38
39	2459						39
40	2460			/	SKIPS PREVIOUS TO EXIT NORMALLY; THIS COULD POSSIBLY BE CONSIDERED AN ERROR.		40
41	2461						41
42	2462	074066*	6004	JUSTEXI,PEX		/EXITING FROM PANEL MEMORY	42
43	2463	074067*	5400	JMP I	INTADR	/BACK TO INTERRUPTED MAIN MEMORY PROGRAM	43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2465	/	COMES HERE IF THE HALT FLAG WAS SET	1
2	2466			2
3	2467	074070* 7200	TESTHLT,CLA /CLEAN UP	3
4	2468	074071* 6003	PG0 /RESET THE HALT FLAG	4
5	2469	074072* 1310	TAD PHLTPC/(HLTPC) /GET EXPECTED INTERRUPTED ADDRESS	5
6	2470	074073* 7041	CIA /INVERT FOR COMPARISON	6
7	2471	074074* 1000	TAD INTADR /COMPARE TO ACTUAL INTERRUPT PC	7
8	2472	074075* 7650	SNA CLA /SKIP IF OTHER THAN EXPECTED	8
9	2473	074076* 5265	JMP BUMPIT /JUMP IF OK	9
10	2474			10
11	2475	/	INTERRUPT PC WRONG.	11
12	2476			12
13	2477	074077* 5266	JMP JUSTEXIT /CONTINUE THERE	13
14	2478			14
15	2479	/	COMES HERE IF MAIN MEMORY PROGRAM EXITED VIA PANEL REQUEST.	15
16	2480			16
17	2481	074100* 7200	PRQEXIT,CLA /CLEAN UP	17
18	2482	074101* 1017	TAD XR7 /RESTORE EXITING AC	18
19	2483	074102* 6273	CIF CDF 70 /BACK TO OUR FIELD	19
20	2484	074103* 5620	JMP I PLBDONE/(LBDONE) /RETURN TO CALLER	20
21	2485			21
22	2486	074104* 0205	PVIDPC, VIDPC /EXPECTED PC FOR VIDEO INTERRUPT	22
23	2487	074105* 0207	PTTYPC, TTYPC /EXPECTED PC FOR TTY: TRAP INTERRUPT	23
24	2488	074106* 0213	PKBIPC, KBIPC /EXPECTED PC FOR KEYBOARD INPUT INTERRUPT	24
25	2489	074107* 0211	PKBOIPC,KBOIPC /EXPECTED PC FOR KEYBOARD OUTPUT INTERRUPT	25
26	2490	074110* 0203	PHLTPC, HLTPC /EXPECTED PC FOR HALT INTERRUPT	26
27	2491			27
28	2492	4111	CPEND= . /END OF CP-INTERRUPT HANDLER CODE THAT OUGHT	28
29	2493		/TO BE MOVED	29
30	2494			30
31	2495	*3111	RELOC CPEND-CPINT+CPLOAD /RE-ESTABLISH ORIGIN	31
32				32
33				33
34				34
35				35
36				36
37				37
38				38
39				39
40				40
41				41
42				42
43				43
44				44
45				45
46				46
47				47
48				48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1	2497	/	THIS ROUTINE READS IN TRACK 0, SECTOR 3 INTO CP 15350-16347. THE FIRST EIGHT			1
2	2498	/	BYTES ARE VALIDATED AGAINST A KNOWN PATTERN. ALL ELEMENTS READ IN AS 377 ARE			2
3	2499	/	REPLACED WITH 4000 IN THE RUNNING COPY.			3
4	2500					4
5	2501	073111* 0000	T0S3RD, .-. /	/TRACK 0, SECTOR 3 READ ROUTINE		5
6	2502	073112* 1344	TAD	PT0S3BUFF/(T0S3BUFFER-1)	/SETUP THE	6
7	2503	073113* 3010	DCA	XR0	/DATA STORAGE POINTER	7
8	2504	073114* 6211	CDF	10	/SET THE DATA FIELD FOR THE TRANSFER	8
9	2505	073115* 7305	NL002A		/SET SECTOR OFFSET (3-1)	9
10	2506	073116* 4742	JMS I	PFREAD/(FREAD)	/CALL FLOPPY READ ROUTINE	10
11	2507	073117* 7774	-3-1		/SECTOR THREE IS THE LAST ONE	11
12	2508	073120* 0000	0		/TRACK 0000	12
13	2509	073121* 5711	JMP I	T0S3RD	/ERROR, TAKE NON-SKIP RETURN	13
14	2510	073122* 1343	TAD	PT0S3DATA/(T0S3DATA-1)	/SETUP THE VALIDATION LIST VALUE	14
15	2511	073123* 3010	DCA	XR0	/STASH THE POINTER	15
16	2512	073124* 1344	TAD	PT0S3BUFF/(T0S3BUFFER-1)	/POINT TO THE	16
17	2513	073125* 3011	DCA	XR1	/READ-IN DATA	17
18	2514	073126* 1410	LISTLUP, TAD I	XR0	/GET A TEST VALUE	18
19	2515	073127* 7450	SNA		/END OF LIST?	19
20	2516	073130* 5337	JMP	LISTOK	/JUMP IF SO	20
21	2517	073131* 6211	CDF	10	/GOTO DATA FIELD	21
22	2518	073132* 1411	TAD I	XR1	/COMPARE TO READ-IN VALUE	22
23	2519	073133* 6271	CDF	70	/BACK TO OUR FIELD	23
24	2520	073134* 7640	SZA CLA		/SKIP IF IT MATCHES	24
25	2521	073135* 5711	JMP I	T0S3RD	/TAKE FAILURE RETURN IF NOT	25
26	2522	073136* 5326	JMP	LISTLUP	/KEEP GOING	26
27	2523					27
28	2524	073137* 4745	LISTOK, JMS I	PRUBCHK/(RUBCHK)	/REPLACE ALL 377 CODES WITH 4000	28
29	2525	073140* 2311	ISZ	T0S3RD	/BUMP TO SUCCESSFUL RETURN	29
30	2526	073141* 5711	JMP I	T0S3RD	/TAKE SKIP-RETURN TO CALLER	30
31	2527					31
32	2528	073142* 1311	PFREAD, FREAD		/POINTER TO FLOPPY READ ROUTINE	32
33	2529	073143* 3774	PT0S3DA, T0S3DATA-1		/POINTER TO TEST DATA FOR TRACK 0, SECTOR 3	33
34	2530	073144* 5347	PT0S3BU, T0S3BUFFER-1		/POINTER TO TRACK 0, SECTOR 3 LOADING ADDRESS	34
35	2531	073145* 5400	PRUBCHK, RUBCHK		/POINTER TO RUBCHK ROUTINE	35
36						36
37						37
38						38
39						39
40						40
41						41
42						42
43						43
44						44
45						45
46						46
47						47
48						48
49						49
50						50
51						51
52						52
53						53
54						54
55						55
56						56
57						57
58						58

1	2533	/	COMES BACK HERE AFTER LOADING THE CHARACTER GENERATOR RAM TO EXIT.		1
2	2534				2
3	2535	073146* 5752	CGDONE, JMP I	DOCG /RETURN TO ORIGINAL CALLER	3
4	2536				4
5	2537	073147* 3763	PT0S7DA,T0S7DATA-1	/POINTER TO TEST DATA FOR TRACK 0, SECTOR 7	5
6	2538	073150* 3205	PUSEDAT,USEDATA	/POINTER TO USE DATA ROUTINE	6
7	2539	073151* 3200	PNODATA,NODATA	/POINTER TO NO DATA ROUTINE	7
8	2540				8
9	2541	/	THE CHARACTER GENERATOR RAM LOAD ROUTINE IS ENTERRED HERE.		9
10	2542				10
11	2543	073152* 0000	DOCG, .-. /CHARACTER GENERATOR RAM LOAD ROUTINE		11
12	2544	073153* 7240	NL7777	/SETUP THE	12
13	2545	073154* 3010	DCA XR0	/STORE POINTER	13
14	2546	073155* 6201	CDF 00	/GOTO DATA FIELD	14
15	2547	073156* 7327	NL0006	/SET OFFSET FOR SECTOR 7 (7-10)	15
16	2548	073157* 4742	JMS I PFREAD/(FREAD)	/CALL FLOPPY READ ROUTINE	16
17	2549	073160* 7765	-12-1	/WANT SECTORS 7-10	17
18	2550	073161* 0000	0	/ON TRACK 0	18
19	2551	073162* 5751	JMP I PNODATA/(NODATA)	/ERROR, TRY TO KEEP GOING	19
20	2552				20
21	2553	/	THE DATA READ IN OK, SO VALIDATE IT.		21
22	2554				22
23	2555	073163* 1347	TAD PT0S7DATA/(T0S7DATA-1)	/SETUP THE	23
24	2556	073164* 3010	DCA XR0	/TEST DATA POINTER	24
25	2557	073165* 7240	NL7777	/SETUP THE	25
26	2558	073166* 3011	DCA XR1	/DATA POINTER	26
27	2559	073167* 1410	TSTD LUP, TAD I XR0	/GET A TEST WORD	27
28	2560	073170* 7450	SNA	/SKIP IF NOT AT END OF LIST	28
29	2561	073171* 5750	JMP I PUSEDATA/(USEDATA)	/END OF LIST, GO USE GOOD DATA	29
30	2562	073172* 6201	CDF 00	/GOTO DATA FIELD	30
31	2563	073173* 1411	TAD I XR1	/COMPARE DATA TO TEST WORD	31
32	2564	073174* 6271	CDF 70	/BACK TO OUR FIELD	32
33	2565	073175* 7650	SNA CLA	/SKIP IF NO MATCH	33
34	2566	073176* 5367	JMP TSTD LUP	/KEEP GOING IF IT MATCHES	34
35	2567				35
36	2568	/	THE DATA ISN'T VALID, SO IGNORE IT AS IF IT HAD BEEN AN I/O ERROR.		36
37	2569				37
38	2570	073177* 5751	JMP I PNODATA/(NODATA)	/CONTINUE THERE	38
39					39
40					40
41					41
42					42
43					43
44					44
45					45
46					46
47					47
48					48
49					49
50					50
51					51
52					52
53					53
54					54
55					55
56					56
57					57
58					58

1	2627	073250*	7315	NL0010	CLL	/SETUP RAM ENABLE AND SCREEN ENABLE	1
2	2628	073251*	6126	VIDCON		/LOAD NEW CONTROL VALUE	2
3	2629	073252*	7200	CLA		/CLEAN UP	3
4	2630	073253*	5654	JMP I	.+1/(CGDONE)	/GO THERE	4
5	2631	073254*	3146	CGDONE		/THROUGH HERE	5
6	2632						6
7	2633	073255*	0110	L110,	110	/CONSTANT 0110	7
8	2634	073256*	3256	PCGINDX,	CGINDX-1	/POINTER TO CG INDEX TABLE	8
9	2635						9
10	2636			/	TABLE OF CG CELL LINE INDICES; (THE FOUR BITS ARE READ BACKWARDS).		10
11	2637						11
12	2638	073257*	0000	CGINDX,	00	/LINE 00	12
13	2639	073260*	0010		10	/LINE 01	13
14	2640	073261*	0004		04	/LINE 02	14
15	2641	073262*	0014		14	/LINE 03	15
16	2642	073263*	0002		02	/LINE 04	16
17	2643	073264*	0012		12	/LINE 05	17
18	2644	073265*	0006		06	/LINE 06	18
19	2645	073266*	0016		16	/LINE 07	19
20	2646	073267*	0001		01	/LINE 08	20
21	2647	073270*	0011		11	/LINE 09	21
22	2648	073271*	0005		05	/LINE 10	22
23	2649	073272*	0015		15	/LINE 11	23
24	2650	073273*	0003		03	/LINE 12	24
25	2651	073274*	0013		13	/LINE 13	25
26	2652	073275*	0007		07	/LINE 14	26
27	2653	073276*	0017		17	/LINE 15	27
28	2654						28
29	2655	073277*	0000	CGPTR,	0000	/CHARACTER GENERATOR DATA POINTER	29
30	2656						30
31	2657			/	NOTE: ALL ROUTINES AND DATA BEYOND THIS POINT MARKED WITH ***** ARE NOT		31
32	2658			/	ACCESSED BY THE NORMAL USAGE OF THE ROM EXCEPT POSSIBLY INDIRECTLY THROUGH A		32
33	2659			/	DOWN-LOADED AUTOMATIC PRODUCT TEST ROUTINE. ALL OF THE CODE IS SUBORDINATE TO		33
34	2660			/	THE SUBROUTINE ENTERRED VIA JMS 0176, WHICH IS ITSELF NOT CALLED BY ANY		34
35	2661			/	EXISTING ROM ROUTINES.		35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2719	/	*****				1
2	2720						2
3	2721	/	COUNT FOR FIELDS PRIOR TO HIGHLIGHTED RATE FIELD.				3
4	2722						4
5	2723	073365*	7777 CLIST,	-1	/50 BAUD		5
6	2724	073366*	7777	-1	/75 BAUD		6
7	2725	073367*	7777	-1	/110 BAUD		7
8	2726	073370*	7777	-1	/134.5 BAUD		8
9	2727	073371*	7777	-1	/150 BAUD		9
10	2728	073372*	7777	-1	/300 BAUD		10
11	2729	073373*	7776	-2	/600 BAUD		11
12	2730	073374*	7775	-3	/1200 BAUD		12
13	2731	073375*	7774	-4	/1800 BAUD		13
14	2732	073376*	7774	-4	/2000 BAUD		14
15	2733	073377*	7774	-4	/2400 BAUD		15
16	2734	073400*	7773	-5	/3600 BAUD		16
17	2735	073401*	7773	-5	/4800 BAUD		17
18	2736	073402*	7772	-6	/7200 BAUD		18
19	2737	073403*	7772	-6	/9600 BAUD		19
20	2738	073404*	7772	-6	/19200 BAUD		20
21	2739						21
22	2740	/	*****				22
23	2741						23
24	2742	073405*	6203 CIDEF,	CIF CDF 00	/CONSTANT 6203		24
25	2743	073406*	1311 AFREAD,	FREAD	/POINTER TO FLOPPY READ ROUTINE		25
26	2744	073407*	4007 PPRMPRT,	PRMPRT	/POINTER TO SETUP PARAMETER PRINT ROUTINE		26
27	2745	073410*	4321 PLOC36,	LOC36	/POINTER TO 36 OR 37 LOCATOR ROUTINE		27
28	2746	073411*	3754 PTOS1DA,	TOS1DAT-1	/TRACK 0 SECTORS 1-2 VALIDATION DATA POINTER		28
29							29
30							30
31							31
32							32
33							33
34							34
35							35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2804	073470*	4551	NEXTCHR, JMS I	PKBDIN/[KBDIN]	/GET A CHARACTER	1
2	2805	073471*	3314	METRENT, DCA	SUBAC	/SAVE LATEST CHARACTER	2
3	2806	073472*	1316		TAD PCOMTABLE/(COMTABLE-1)	/SETUP THE	3
4	2807	073473*	3010		DCA XR0	/TABLE POINTER	4
5	2808	073474*	1410	COMSRCH, TAD I	XR0	/GET A WORD	5
6	2809	073475*	7450		SNA	/SKIP IF NOT AT END OF LIST	6
7	2810	073476*	5270		JMP NEXTCHR	/END OF LIST, IGNORE AND GET ANOTHER	7
8	2811	073477*	1314		TAD SUBAC	/COMPARE TO CHARACTER	8
9	2812	073500*	7650		SNA CLA	/SKIP IF NOT A MATCH	9
10	2813	073501*	5304		JMP FINDIT	/JUMP IF IT MATCHES	10
11	2814	073502*	2010		ISZ XR0	/BUMP PAST DISPATCH ADDRESS	11
12	2815	073503*	5274		JMP COMSRCH	/TRY AGAIN	12
13	2816						13
14	2817	073504*	1410	FINDIT, TAD I	XR0	/GET DISPATCH ADDRESS	14
15	2818	073505*	3315		DCA DISPADR	/STASH IT	15
16	2819	073506*	1314		TAD SUBAC	/GET THE CHARACTER ITSELF AS AN ARGUMENT	16
17	2820	073507*	4715		JMS I DISPADR	/DISPATCH TO ROUTINE	17
18	2821	073510*	5270		JMP NEXTCHR	/NO-CHANGE RETURN, GO THERE	18
19	2822	073511*	5265		JMP NEXTPRM	/POSSIBLE-CHANGE RETURN, GO THERE	19
20	2823	073512*	7402	SUBCDF, HLT+.-.		/DOUBLE-SKIP RETURN; WILL BE CIF CDF RETURN	20
21	2824	073513*	5576		JMP I SUB176	/RETURN TO CALLER	21
22	2825						22
23	2826	073514*	0000	SUBAC, .-.		/AC DATA SAVED HERE	23
24	2827	073515*	0000	DISPADR, .-.		/DISPATCH ADDRESS TEMPORARY	24
25	2828	073516*	3735	PCOMTAB, COMTABLE-1		/POINTER TO COMMAND DISPATCH TABLE	25
26	2829	073517*	3715	PVIDCLR, VIDCLR		/POINTER TO VIDEO CLEAR ROUTINE	26
27							27
28							28
29							29
30							30
31							31
32							32
33							33
34							34
35							35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2935	/	*****				1
2	2936						2
3	2937	073657* 0000	SELSHFT, .-. .			/SELECT ROUTINE COUNTER	3
4	2938	073660* 7762	M16, -16			/CONSTANT 7762	4
5	2939	073661* 7773	L7773, -5			/CONSTANT 7773	5
6	2940	073662* 0005	L5, 5			/CONSTANT 0005	6
7	2941	073663* 3664	PBTABLE, BTABLE			/POINTER TO BAUD RATE TABLE	7
8	2942						8
9	2943	/	BAUD RATE TABLE FOR SUPPORTED RATES.				9
10	2944						10
11	2945	073664* 0005	BTABLE, 05			/300 BAUD	11
12	2946	073665* 0006	06			/600 BAUD	12
13	2947	073666* 0007	07			/1200 BAUD	13
14	2948	073667* 0012	0012			/2400 BAUD	14
15	2949	073670* 0014	14			/4800 BAUD	15
16	2950	073671* 0016	16			/9600 BAUD	16
17	2951						17
18	2952	/	*****				18
19	2953						19
20	2954	073672* 0000	KBDIN, .-. .			/COMMAND INPUT ROUTINE	20
21	2955	073673* 6111	KBISF			/FLAG UP?	21
22	2956	073674* 5273	JMP .-1			/NO, WAIT FOR IT	22
23	2957	073675* 6116	KBIRB			/YES, READ IT IN	23
24	2958	073676* 0115	AND Z377/[377]			/JUST EIGHT BITS	24
25	2959	073677* 5672	JMP I KBDIN			/RETURN	25
26	2960						26
27	2961	073700* 1140	ARXWAIT, RXWAIT			/POINTER TO RX50 WAIT ROUTINE	27
28	2962						28
29	2963	073701* 0000	RXRDY, .-. .			/DRIVE READY CHECK ROUTINE	29
30	2964	073702* 1106	TAD Z12/[12]			/GET READ STATUS VALUE	30
31	2965	073703* 6751	LCD			/LOAD THE COMMAND	31
32	2966	073704* 4700	JMS I ARXWAIT/(RXWAIT)			/WAIT FOR IT	32
33	2967	073705* 5701	JMP I RXRDY			/TRANSFER FLAG CAME UP, FORGET IT	33
34	2968	073706* 7410	SKP			/DONE FLAG CAME UP, THERE'S HOPE	34
35	2969	073707* 5701	JMP I RXRDY			/ERROR OR TIME-OUT	35
36	2970	073710* 6752	XDR			/GET STATUS	36
37	2971	073711* 0112	AND Z200/[200]			/JUST DRIVE READY BIT	37
38	2972	073712* 7640	SZA CLA			/SKIP IF NOT READY	38
39	2973	073713* 2301	ISZ RXRDY			/ELSE TAKE SKIP RETURN	39
40	2974	073714* 5701	JMP I RXRDY			/RETURN EITHER WAY	40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	2976	/	*****				1
2	2977						2
3	2978	073715* 0000	VIDCLR, .-. /VIDEO CLEAR (?) ROUTINE				3
4	2979	073716* 7333	NL6000 /SET BOTH CONTROL BITS				4
5	2980	073717* 6126	VIDCON /LOAD CONTROL REGISTER				5
6	2981	073720* 7200	CLA /CLEAN UP				6
7	2982	073721* 3010	DCA XR0 /CLEAR INNER COUNTER				7
8	2983	073722* 7346	NL7775 /SET THE				8
9	2984	073723* 3011	DCA XR1 /OUTER COUNTER				9
10	2985	073724* 2010	VIDWAIT, ISZ XR0 /WAITING ENOUGH?				10
11	2986	073725* 5324	JMP VIDWAIT /NO, KEEP GOING				11
12	2987	073726* 2011	ISZ XR1 /WAITED TOO LONG?				12
13	2988	073727* 5324	JMP VIDWAIT /NO, KEEP GOING				13
14	2989	073730* 6126	VIDCON /RESET TO NORMAL VIDEO MODE				14
15	2990	073731* 5715	JMP I VIDCLR /RETURN				15
16	2991						16
17	2992	/	SLUSHWARE TEST VALIDATION DATA.				17
18	2993						18
19	2994	073732* 1577	SLTDAT, -6201				19
20	2995	073733* 7745	-0033				20
21	2996	073734* 7701	-0077				21
22	2997	073735* 0000	0 /THIS ENDS THE LIST				22
23	2998						23
24	2999	/	*****				24
25	3000						25
26	3001	/	COMMAND DISPATCH TABLE FOR BUILT-IN COMMANDS.				26
27	3002						27
28	3003	073736* 7564	COMTABL, -214; BOOTIT /REMOVE				28
29	3004	073737* 4401					29
30	3005	073740* 7563	-215; SELCT /SELECT				30
31	3006	073741* 3600					31
32	3007	073742* 7603	-175; DOKEY /DO				32
33	3008	073743* 4446					33
34	3009	073744* 7531	-247; MOVLEFT /LEFT ARROW				34
35	3010	073745* 4275					35
36	3011	073746* 7530	-250; MOVWRITE /RIGHT ARROW				36
37	3012	073747* 4311					37
38	3013	073750* 7503	-275; RETURN /RETURN				38
39	3014	073751* 4606					39
40	3015	073752* 7514	-264; METRONOME /METRONOME				40
41	3016	073753* 4600					41
42	3017	073754* 0000	0 /THIS ENDS THE LIST				42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	3107	074064*	1025	PRTRATE, TAD	TEMP2	/GET SECOND PARAMETER	1
2	3108	074065*	4303		JMS NPRT	/OUTPUT IT	2
3	3109	074066*	1026		TAD TEMP3	/GET THIRD PARAMETER	3
4	3110	074067*	4303		JMS NPRT	/OUTPUT IT	4
5	3111	074070*	1410	OUTLUP, TAD I	XR0	/GET A CHARACTER	5
6	3112	074071*	7421		MQL	/LOAD FOR PRINTING	6
7	3113	074072*	7701		CLA MQA	/GET IT BACK	7
8	3114	074073*	1133		TAD Z7775/[7775]	/COMPARE TO END CHARACTER	8
9	3115	074074*	7650		SNA CLA	/SKIP IF NOT AT END	9
10	3116	074075*	5607		JMP I PRMPRT	/RETURN IF AT END	10
11	3117	074076*	4501		JMS I PCHRPRT/[CHRPRT]	/PRINT THE CHARACTER	11
12	3118	074077*	5270		JMP OUTLUP	/GO DO ANOTHER	12
13	3119						13
14	3120			/	UNREFERENCED LOCATION. PERHAPS ERRONEOUS NOTION THAT CHRPRT ROUTINE MIGHT		14
15	3121			/	SKIP OR OTHER OBSOLETE CONVENTION?		15
16	3122						16
17	3123	074100*	5607		JMP I PRMPRT	/RETURN	17
18	3124						18
19	3125	074101*	0000	PRMTM1, .-. .		/TEMPORARY	19
20	3126	074102*	0000	PRMTM2, .-. .		/TEMPORARY	20
21	3127						21
22	3128			/	*****		22
23	3129						23
24	3130			/	PRINT NUMERICAL PARAMETER ROUTINE. THE SELECTED PARAMETER IS A BAUD RATE.		24
25	3131			/	THE (CLOSEST TO THE) SELECTED RATE IS HIGHLIGHTED.		25
26	3132						26
27	3133	074103*	0000	NPRT, .-. .			27
28	3134	074104*	3351		DCA NTMP	/SAVE THE PASSED VALUE	28
29	3135	074105*	1351		TAD NTMP	/GET IT BACK	29
30	3136	074106*	0107		AND Z17/[17]	/JUST RATE BITS	30
31	3137	074107*	1352		TAD PCLIST/(CLIST)	/ADD ON TABLE BASE	31
32	3138	074110*	3351		DCA NTMP	/STASH THE POINTER	32
33	3139	074111*	1751		TAD I NTMP	/GET THE PROPER COUNT	33
34	3140	074112*	3351		DCA NTMP	/STASH IT	34
35	3141	074113*	1410	NPLUP, TAD I	XR0	/GET A CHARACTER	35
36	3142	074114*	7421		MQL	/LOAD IT FOR PRINTING	36
37	3143	074115*	7701		CLA MQA	/GET IT BACK	37
38	3144	074116*	1130		TAD Z7741/[-37]	/COMPARE TO FIELD DELIMITER	38
39	3145	074117*	7450		SNA	/SKIP IF NOT A MATCH	39
40	3146	074120*	5336		JMP FNDDELIM	/JUMP IF IT IS	40
41	3147	074121*	7001		IAC	/COMPARE TO ALTERNATE DELIMITER VALUE	41
42	3148	074122*	7650		SNA CLA	/SKIP IF NOT A MATCH THERE EITHER	42
43	3149	074123*	5336		JMP FNDDELIM	/JUMP IF IT IS A MATCH	43
44	3150	074124*	4501	NEXTREG, JMS I	PCHRPRT/[CHRPRT]	/PRINT THE CHARACTER	44
45	3151	074125*	5313		JMP NPLUP	/GO DO NEXT	45
46	3152						46
47	3153	074126*	4501	MOREPRM, JMS I	PCHRPRT/[CHRPRT]	/PRINT THE CHARACTER	47
48	3154	074127*	7701		CLA MQA	/GET THE CHARACTER	48
49	3155	074130*	1126		TAD Z7766/[7766]	/COMPARE TO MASTER DELIMITER	49
50	3156	074131*	7650		SNA CLA	/SKIP IF NOT AT END	50
51	3157	074132*	5703		JMP I NPRT	/RETURN IF AT END	51
52	3158	074133*	1410		TAD I XR0	/GET NEXT CHARACTER	52
53	3159	074134*	7421		MQL	/LOAD IT FOR PRINTING	53
54	3160	074135*	5326		JMP MOREPRM	/KEEP GOING	54
55							55
56							56
57							57
58							58

1	3162		/	COMES HERE IF FIELD DELIMITER FOUND.				1
2	3163							2
3	3164	074136*	2351	FNDDELI, ISZ	NTMP		/FOUND ENOUGH REGULAR FIELDS?	3
4	3165	074137*	5324		JMP	NEXTREG	/NO, KEEP GOING	4
5	3166	074140*	4501	BOLDLP, JMS I	PCHRPRT/[CHRPRT]		/PRINT THE CHARACTER	5
6	3167	074141*	1410		TAD I	XR0	/GET ANOTHER CHARACTER	6
7	3168	074142*	7421		MQL		/LOAD IT FOR PRINTING	7
8	3169	074143*	7701		CLA	MQA	/GET IT BACK	8
9	3170	074144*	1130		TAD	Z7741/[-37]	/COMPARE TO FIELD DELIMITER	9
10	3171	074145*	7750		SPA	SNA CLA	/SKIP IF HIGHER	10
11	3172	074146*	5326		JMP	MOREPRM	/JUMP IF NOT	11
12	3173	074147*	1116		TAD	Z400/[400]	/GET REVERSE VIDEO BIT	12
13	3174	074150*	5340		JMP	BOLDLP	/GO DO NEXT	13
14	3175							14
15	3176	074151*	0000	NTMP, .-. .			/TEMPORARY	15
16	3177	074152*	3365	PCLIST, CLIST			/POINTER TO REGULAR FIELD COUNT LIST	16
17	3178							17
18	3179			/	SLUSHWARE CODE VALIDATION ROUTINE.			18
19	3180							19
20	3181	074153*	0000	SLCHECK, .-. .			/SLUSHWARE VALIDATION ROUTINE	20
21	3182	074154*	6214		RDF		/GET DATA FIELD	21
22	3183	074155*	1122		TAD	ZCDF/[CDF]	/FORM CDF INSTRUCTION	22
23	3184	074156*	3365		DCA	SLUCDF	/STORE IN-LINE	23
24	3185	074157*	1375		TAD	PSLTDAT/(SLTDAT-1)	/SETUP THE	24
25	3186	074160*	3011		DCA	XR1	/VALIDATION POINTER	25
26	3187	074161*	6271		CDF	70	/BACK TO OUR FIELD	26
27	3188	074162*	1411	SLNEXT, TAD I	XR1		/GET A TEST WORD	27
28	3189	074163*	7450		SNA		/SKIP IF NOT AT END OF LIST	28
29	3190	074164*	5373		JMP	SLGOOD	/JUMP IF AT END	29
30	3191	074165*	6201	SLUCDF, CDF	00+.-.		/WILL BE CDF XX INSTRUCTION	30
31	3192	074166*	1410		TAD I	XR0	/COMPARE TO READ-IN DATA	31
32	3193	074167*	6271		CDF	70	/BACK TO OUR FIELD	32
33	3194	074170*	7650		SNA	CLA	/SKIP IF NOT A MATCH	33
34	3195	074171*	5362		JMP	SLNEXT	/JUMP IF IT MATCHES	34
35	3196	074172*	5753		JMP I	SLCHECK	/TAKE FAILURE RETURN	35
36	3197							36
37	3198	074173*	2353	SLGOOD, ISZ	SLCHECK		/BUMP TO GOOD RETURN	37
38	3199	074174*	5753		JMP I	SLCHECK	/TAKE SKIP RETURN TO CALLER	38
39	3200							39
40	3201	074175*	3731	PSLTDAT, SLTDAT-1			/POINTER TO SLUSHWARE TEST DATA	40
41	3202							41
42	3203	074176*	0000		ZBLOCK	2	/EMPTY SPACE	42
43								43
44								44
45								45
46								46
47								47
48								48
49								49
50								50
51								51
52								52
53								53
54								54
55								55
56								56
57								57
58								58

1	3205	*4200	PAGE			1
2	3206					2
3	3207	/	*****			3
4	3208					4
5	3209	074200* 3661	HIGHROW,ROWADR+1		/POINTER TO ROWTABLE (HIGH-ORDER)	5
6	3210	074201* 2030	L2030, 2030		/CONSTANT 2030	6
7	3211	074202* 0031	L31, 31		/CONSTANT 0031	7
8	3212	074203* 7747	M31, -31		/CONSTANT 7747	8
9	3213					9
10	3214	074204* 0000	CHRPRT, .-. .		/CHARACTER PRINT ROUTINE	10
11	3215	074205* 3254	DCA ATTRVAL		/SAVE PASSED ATTRIBUTE VALUE	11
12	3216	074206* 7701	CLA MQA		/GET THE CHARACTER	12
13	3217	074207* 1126	TAD Z7766/[-"J!300]		/COMPARE TO <LF>	13
14	3218	074210* 7450	SNA		/SKIP IF OTHER	14
15	3219	074211* 5252	JMP JUSTPOS		/JUMP IF IT MATCHES	15
16	3220	074212* 1133	TAD Z7775/[-"M+"J]		/COMPARE TO <CR>	16
17	3221	074213* 7650	SNA CLA		/SKIP IF NO MATCH	17
18	3222	074214* 5252	JMP JUSTPOS		/JUMP IF IT MATCHES	18
19	3223	074215* 1056	TAD ROW		/GET DESIRED ROW NUMBER	19
20	3224	074216* 7104	CLL RAL		/*2	20
21	3225	074217* 1200	TAD HIGHROW/(ROWADR+1)		/POINT TO PROPER HIGH-ORDER ADDRESS	21
22	3226	074220* 3255	DCA CHARADR		/STASH THE POINTER	22
23	3227	074221* 6211	CDF 10		/GOTO ROWTABLE FIELD	23
24	3228	074222* 1655	TAD I CHARADR		/GET THE HIGH-ORDER ROW ADDRESS	24
25	3229	074223* 6271	CDF 70		/BACK TO OUR FIELD	25
26	3230	074224* 3255	DCA CHARADR		/STASH THE POINTER	26
27	3231	074225* 1255	TAD CHARADR		/GET HIGH-ORDER ROW ADDRESS	27
28	3232	074226* 7110	CLL RAR		/MOVE OVER	28
29	3233	074227* 0201	AND L2030/(30)		/JUST FIELD BITS (CLOSE ENOUGH)	29
30	3234	074230* 1122	TAD ZCDF/[CDF]		/FORM CDF INSTRUCTION	30
31	3235	074231* 3247	DCA CHARCDF		/STORE IN-LINE	31
32	3236	074232* 1255	TAD CHARADR		/GET HIGH-ORDER ROW ADDRESS AGAIN	32
33	3237	074233* 0107	AND Z17/[17]		/JUST QUARTER-FIELD BITS	33
34	3238	074234* 7106	CLL RTL;BSW		/*400	34
35	3239	074235* 7002				35
36	3240	074236* 1057	TAD COLUMN		/ADD ON DESIRED COLUMN	36
37	3241	074237* 1153	TAD Z205/[205]		/ADD ON LINE BASE	37
38	3242	074240* 3255	DCA CHARADR		/STASH THE POINTER	38
39	3243	074241* 7701	CLA MQA		/GET THE CHARACTER	39
40	3244	074242* 1256	TAD M40/(-40)		/COMPARE TO <SPACE>	40
41	3245	074243* 7740	SMA SZA CLA		/SKIP IF <SPACE> OR CONTROL CHARACTER	41
42	3246	074244* 7701	CLA MQA		/ELSE GET THE CHARACTER	42
43	3247	074245* 7440	SZA		/SKIP IF NOT DISPLAYABLE	43
44	3248	074246* 1254	TAD ATTRVAL		/ELSE ADD ON PASSED ATTRIBUTES	44
45	3249	074247* 7402	CHARCDF,HLT+.-.		/WILL BE CDF 20 (OR 30) INSTRUCTION	45
46	3250	074250* 3655	DCA I CHARADR		/STORE IN LINE BUFFER	46
47	3251	074251* 6271	CDF 70		/BACK TO OUR FIELD	47
48	3252	074252* 4257	JUSTPOS,JMS SCRPOS		/ADVANCE SCREEN POSITION	48
49	3253	074253* 5604	JMP I CHRPRT		/RETURN	49
50	3254					50
51	3255	074254* 0000	ATTRVAL,.-.		/PASSED ATTRIBUTE BITS SAVED HERE	51
52	3256	074255* 0000	CHARADR,.-.		/ADDRESS TEMPORARY	52
53	3257	074256* 7740	M40, -" !200		/CONSTANT 7740	53
54						54
55						55
56						56
57						57
58						58

1	3566	074647*	0000	MVCNT, .-		/MOVE COUNTER	1
2	3567	074650*	7401	M377, -377		/CONSTANT 7401	2
3	3568	074651*	0017	BAUD17, 17		/CONSTANT 0017	3
4	3569						4
5	3570			/	COMES HERE AFTER TESTING THE KEYBOARD (OR JUST AFTER THE TEST WAS BYPASSED IF		5
6	3571			/	APTEN IS IN EFFECT) TO SETUP THE COMMUNICATIONS CHIP.		6
7	3572						7
8	3573	074652*	1251	COMTEST, TAD	BAUD17/(17)	/GET BAUD RATE VALUE FOR 19200 BAUD	8
9	3574	074653*	6363	MSB		/SETUP COMMUNICATIONS CHIP RATE	9
10	3575	074654*	7200	CLA		/CLEAN UP	10
11	3576	074655*	6367	MPRESET		/RESET THE COMMUNICATIONS CHIP	11
12	3577	074656*	6121	APTSKP		/AUTOMATIC PRODUCT TESTING?	12
13	3578	074657*	7410	SKP		/SKIP IF NOT	13
14	3579	074660*	5270	JMP	TESTCOM	/JUMP IF SO	14
15	3580	074661*	1104	COMEXIT, TAD	Z7/[7]	/GET 1200 BAUD RATE VALUE	15
16	3581	074662*	6363	MSB		/SETUP COMM CHIP FOR 1200 BAUD	16
17	3582	074663*	7200	CLA		/CLEAN UP	17
18	3583	074664*	4667	JMS I	PREGINIT/(REGINIT)	/INITIALIZE THE REGISTERS	18
19	3584	074665*	5666	JMP I	+.1/(COMDONE)	/GO BACK TO MAIN-LINE CODE	19
20	3585	074666*	1007	COMDONE		/THROUGH HERE	20
21	3586						21
22	3587	074667*	5230	PREGINI, REGINIT		/POINTER TO REGISTER INITIALIZATION ROUTINE	22
23	3588						23
24	3589	074670*	4667	TESTCOM, JMS I	PREGINIT/(REGINIT)	/INITIALIZE THE REGISTERS	24
25	3590	074671*	3030	DCA	TM0	/CLEAR OUTPUT TEST VALUE	25
26	3591	074672*	3031	DCA	COMTM1	/CLEAR INPUT TEST VALUE	26
27	3592	074673*	3050	DCA	T6	/CLEAR LOW-ORDER TIME-OUT COUNTER	27
28	3593	074674*	1130	TAD	Z7741/[-37]	/SETUP THE	28
29	3594	074675*	3051	DCA	T7	/HIGH-ORDER TIME-OUT COUNTER	29
30	3595	074676*	6201	CDF	00	/SET EVEN FIELD TO ACCESS "A" SIDE	30
31	3596	074677*	6316	OLS		/OUTPUT A NULL CHARACTER	31
32	3597	074700*	6271	CDF	70	/BACK TO OUR FIELD	32
33	3598	074701*	2030	ISZ	TM0	/BUMP OUTPUT VALUE	33
34	3599	074702*	6301	COMWAIT, ISF		/COMMUNICATIONS PORT FLAG UP?	34
35	3600	074703*	7410	SKP		/SKIP IF NOT	35
36	3601	074704*	5315	JMP	COMCHK	/JUMP IF SO	36
37	3602	074705*	2050	ISZ	T6	/WAITING TOO LONG?	37
38	3603	074706*	5302	JMP	COMWAIT	/NO, KEEP TRYING	38
39	3604	074707*	2051	ISZ	T7	/WAITING TOO MANY TIMES?	39
40	3605	074710*	5302	JMP	COMWAIT	/NO, KEEP TRYING	40
41	3606	074711*	1372	COMBAD, TAD	L100/(100)	/GET COMMUNICATIONS PORT ERROR BIT	41
42	3607	074712*	1054	TAD	ESTATUS	/UPDATE CUMULATIVE ERROR STATUS	42
43	3608	074713*	3054	DCA	ESTATUS	/STORE BACK	43
44	3609	074714*	5261	JMP	COMEXIT	/STOP TESTING	44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	3611	074715*	4543	COMCHK, JMS I	PCOMREAD/[COMREAD]	/READ CONTENTS OF	1
2	3612	074716*	0012		B^10+2	/REGISTER 2B	2
3	3613	074717*	0107		AND Z17/[17]	/JUST LOW-ORDER BITS AFFECTING "A" HALF	3
4	3614	074720*	1373		TAD M6/(-6)	/RECEIVE CHARACTER AVAILABLE?	4
5	3615	074721*	7440		SZA	/SKIP IF AS EXPECTED	5
6	3616	074722*	5350		JMP CCHKMORE	/JUMP IF NOT	6
7	3617	074723*	6201		CDF 00	/NEED EVEN FIELD TO ACCESS "A" SIDE	7
8	3618	074724*	6306		IRB	/GET THE CHARACTER	8
9	3619	074725*	6271		CDF 70	/BACK TO OUR FIELD	9
10	3620	074726*	0115		AND Z377/[377]	/JUST DATA BITS	10
11	3621	074727*	7041		CIA	/INVERT FOR COMPARISON	11
12	3622	074730*	1031		TAD COMTM1	/COMPARE TO EXPECTED VALUE	12
13	3623	074731*	7650		SNA CLA	/SKIP IF DIFFERENT	13
14	3624	074732*	5337		JMP CREADOK	/JUMP IF IT MATCHES	14
15	3625	074733*	4542	JMS I	PCOMLOAD/[COMLOAD]	/CALL REGISTER LOAD ROUTINE	15
16	3626	074734*	0000		A^10+0	/REGISTER 0A	16
17	3627	074735*	0030		030	/CHANNEL RESET	17
18	3628	074736*	5311		JMP COMBAD	/FORGET IT	18
19	3629						19
20	3630	074737*	2031	CREADOK, ISZ	COMTM1	/BUMP INPUT TEST VALUE	20
21	3631	074740*	1031		TAD COMTM1	/GET THE VALUE	21
22	3632	074741*	0116		AND Z400/[400]	/JUST TOO-FAR BIT	22
23	3633	074742*	7640		SZA CLA	/SKIP IF STILL GOING	23
24	3634	074743*	5261		JMP COMEXIT	/JUMP IF FINISHED	24
25	3635	074744*	4542	CINTCLR, JMS I	PCOMLOAD/[COMLOAD]	/CALL REGISTER LOAD ROUTINE	25
26	3636	074745*	0000		A^10+0	/0A	26
27	3637	074746*	0070		070	/END OF INTERRUPT	27
28	3638	074747*	5302		JMP COMWAIT	/GO BACK FOR MORE	28
29	3639						29
30	3640	074750*	1374	CCHKMOR, TAD	L2/(-4+6)	/TRANSMITTER BUFFER EMPTY?	30
31	3641	074751*	7640		SZA CLA	/SKIP IF IT MATCHES	31
32	3642	074752*	5311		JMP COMBAD	/JUMP IF NOT	32
33	3643	074753*	1030		TAD TM0	/GET OUTPUT TEST VALUE	33
34	3644	074754*	0116		AND Z400/[400]	/JUST TOO-FAR BIT	34
35	3645	074755*	7650		SNA CLA	/SKIP IF NOW TOO FAR	35
36	3646	074756*	5363		JMP OUTMORE	/JUMP IF NOT	36
37	3647	074757*	4542	JMS I	PCOMLOAD/[COMLOAD]	/CALL REGISTER LOAD ROUTINE	37
38	3648	074760*	0000		A^10+0	/0A	38
39	3649	074761*	0050		050	/CLEAR TRANSMITTER INTERRUPT	39
40	3650	074762*	5344		JMP CINTCLR	/CONTINUE THERE	40
41	3651						41
42	3652	074763*	1030	OUTMORE, TAD	TM0	/GET OUTPUT VALUE	42
43	3653	074764*	6201		CDF 00	/SET EVEN FIELD TO ACCESS "A" HALF	43
44	3654	074765*	6316		OLS	/OUTPUT THE CHARACTER	44
45	3655	074766*	6271		CDF 70	/BACK TO OUR FIELD	45
46	3656	074767*	7200		CLA	/CLEAN UP	46
47	3657	074770*	2030		ISZ TM0	/BUMP TEST VALUE	47
48	3658	074771*	5344		JMP CINTCLR	/CONTINUE THERE	48
49	3659						49
50	3660	074772*	0100	L100, 100		/CONSTANT 0100	50
51	3661	074773*	7772	M6, -6		/CONSTANT 7772	51
52	3662	074774*	0002	L2, -4+6		/CONSTANT 0002	52
53	3663						53
54	3664	074775*	0000	ZBLOCK 3		/EMPTY SPACE	54
55							55
56							56
57							57
58							58

1	3720	075054*	7750	M30,	-30	/MIDDLE TIME-OUT CONSTANT	1
2	3721	075055*	0032	L32,	32	/CONSTANT 0032	2
3	3722	075056*	7766	M12,	-12	/OUTER TIME-OUT CONSTANT	3
4	3723						4
5	3724	075057*	0006		6	/AN UNUSED CONSTANT 0006	5
6	3725						6
7	3726	075060*	5060	PTSTDAT,	TSTDATA-1	/POINTER TO TEST DATA TABLE (-1)	7
8	3727						8
9	3728			/	TEST CONTENTS OF CALLABLE SUBROUTINE. ALL VALUES ARE INVERTED. THE ACTUAL		9
10	3729			/	READ-IN CODE IS EXECUTABLE AS AND INSTRUCTIONS.		10
11	3730						11
12	3731	075061*	7777	TSTDATA,	-1		12
13	3732	075062*	7656		-"R!200		13
14	3733	075063*	7674		-"D!200		14
15	3734	075064*	7713		-"5!200		15
16	3735	075065*	7717		-"1!200		16
17	3736	075066*	7776		-2		17
18	3737	075067*	0000		0	/THIS ENDS THE LIST	18
19	3738						19
20	3739			/	RD51 WAIT ROUTINE. TAKES IMMEDIATE RETURN ON ERROR OR TIME-OUT; TAKES SKIP		20
21	3740			/	RETURN ON TRANSFER REQUEST FLAG; TAKES DOUBLE-SKIP RETURN ON DONE FLAG.		21
22	3741						22
23	3742	075070*	0000	RDWAIT,	.-.	/RD51 WAIT ROUTINE	23
24	3743	075071*	1132	TAD	DTIMOUT/[-DTIME]	/SETUP THE	24
25	3744	075072*	3051	DCA	T7	/TIME-OUT COUNTER	25
26	3745	075073*	3050	DCA	T6	/CLEAR INNER TIME-OUT COUNTER	26
27	3746	075074*	6701	RDWATLP,	RDSR	/DATA REQUEST FLAG UP?	27
28	3747	075075*	7410	SKP		/SKIP IF NOT	28
29	3748	075076*	5313	JMP	RDTRAN	/JUMP IF SO	29
30	3749	075077*	6703	RDSR		/DONE FLAG UP?	30
31	3750	075100*	7410	SKP		/SKIP IF NOT	31
32	3751	075101*	5307	JMP	RDDONE	/JUMP IF SO	32
33	3752	075102*	2050	ISZ	T6	/WAITED ENOUGH?	33
34	3753	075103*	5274	JMP	RDWATLP	/NO, KEEP GOING	34
35	3754	075104*	2051	ISZ	T7	/WAITING TOO LONG?	35
36	3755	075105*	5274	JMP	RDWATLP	/NO, KEEP GOING	36
37	3756	075106*	5670	JMP I	RDWAIT	/YES, TAKE IMMEDIATE RETURN	37
38	3757						38
39	3758	075107*	6706	RDDONE,	RDSE	/ANY ERRORS?	39
40	3759	075110*	7410	SKP		/SKIP IF NOT	40
41	3760	075111*	5670	JMP I	RDWAIT	/TAKE IMMEDIATE RETURN IF SO	41
42	3761	075112*	2270	ISZ	RDWAIT	/TAKE DOUBLE-SKIP RETURN ON DONE FLAG RAISE	42
43	3762	075113*	2270	RDTRAN,	ISZ	RDWAIT	43
44	3763	075114*	5670	JMP I	RDWAIT	/RETURN EITHER WAY	44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	3765	/	MPSCC REGISTER READ ROUTINE. CALLED WITH IN-LINE ARGUMENT OF EVEN/ODD FIELD		1
2	3766	/	IN BITS[6-8] AND SPECIFIED REGISTER IN BITS[9-11]. THE CONTENTS OF THE		2
3	3767	/	SPECIFIED REGISTER ARE RETURNED IN THE AC.		3
4	3768				4
5	3769	075115* 0000	COMREAD, .-. .	/MPSCC CHIP REGISTER READ ROUTINE	5
6	3770	075116* 1715	TAD I COMREAD	/GET THE IN-LINE ARGUMENT	6
7	3771	075117* 2315	ISZ COMREAD	/BUMP PAST IT	7
8	3772	075120* 3020	DCA CTMP1	/SAVE IT	8
9	3773	075121* 1020	TAD CTMP1	/GET IT BACK	9
10	3774	075122* 0366	AND L70/(70)	/JUST FIELD BITS	10
11	3775	075123* 1122	TAD ZCDF/[CDF]	/FORM CDF INSTRUCTION	11
12	3776	075124* 3325	DCA .+1	/STORE IN-LINE	12
13	3777	075125* 7402	HLT+.-.	/WILL BE CDF XX INSTRUCTION	13
14	3778	075126* 1020	TAD CTMP1	/GET ARGUMENT AGAIN	14
15	3779	075127* 0104	AND Z7/[7]	/JUST REGISTER BITS	15
16	3780	075130* 1336	TAD L4000/(4000)	/ADD ON READ ACCESS BIT	16
17	3781	075131* 6366	MPSCC	/SETUP TO READ THE SPECIFIED REGISTER	17
18	3782	075132* 7200	CLA	/CLEAN UP	18
19	3783	075133* 6366	MPSCC	/READ THE REGISTER	19
20	3784	075134* 6271	CDF 70	/BACK TO OUR FIELD	20
21	3785	075135* 5715	JMP I COMREAD	/RETURN	21
22	3786				22
23	3787	075136* 4000	L4000, 4000	/CONSTANT 4000	23
24	3788				24
25	3789	/	MPSCC REGISTER LOAD ROUTINE. CALLED WITH IN-LINE ARGUMENTS OF EVEN/ODD FIELD		25
26	3790	/	IN BITS[6-8] AND SPECIFIED REGISTER IN BITS[9-11] OF THE FIRST ARGUMENT,		26
27	3791	/	FOLLOWED BY VALUE TO LOAD INTO THE SPECIFIED REGISTER IN THE SECOND ARGUMENT.		27
28	3792				28
29	3793	075137* 0000	COMLOAD, .-. .	/COMMUNICATIONS CHIP REGISTER LOAD ROUTINE	29
30	3794	075140* 7200	CLA	/CLEAN UP	30
31	3795	075141* 1737	TAD I COMLOAD	/GET THE FIRST ARGUMENT	31
32	3796	075142* 2337	ISZ COMLOAD	/BUMP PAST IT	32
33	3797	075143* 3020	DCA CTMP1	/STASH IT	33
34	3798	075144* 1737	TAD I COMLOAD	/GET THE SECOND ARGUMENT	34
35	3799	075145* 2337	ISZ COMLOAD	/BUMP PAST IT	35
36	3800	075146* 3021	DCA CTMP2	/STASH IT	36
37	3801	075147* 1020	TAD CTMP1	/GET THE FIRST ARGUMENT	37
38	3802	075150* 0366	AND L70/(70)	/JUST FIELD BITS	38
39	3803	075151* 1122	TAD ZCDF/[CDF]	/FORM CDF INSTRUCTION	39
40	3804	075152* 3353	DCA .+1	/STORE IN-LINE	40
41	3805	075153* 7402	HLT+.-.	/WILL BE CDF XX INSTRUCTION	41
42	3806	075154* 1020	TAD CTMP1	/GET THE FIRST ARGUMENT AGAIN	42
43	3807	075155* 0104	AND Z7/[7]	/JUST REGISTER BITS	43
44	3808	075156* 7440	SZA	/DON'T NEED TO POINT IF ACCESSING REGISTER 0	44
45	3809	075157* 6366	MPSCC	/POINT TO DESIRED REGISTER NEXT TIME	45
46	3810	075160* 7200	CLA	/CLEAN UP	46
47	3811	075161* 1021	TAD CTMP2	/GET THE NEW REGISTER CONTENTS	47
48	3812	075162* 6366	MPSCC	/LOAD THE NEW VALUE	48
49	3813	075163* 7200	CLA	/CLEAN UP	49
50	3814	075164* 6271	CDF 70	/BACK TO OUR FIELD	50
51	3815	075165* 5737	JMP I COMLOAD	/RETURN TO CALLER	51
52	3816				52
53	3817	075166* 0070	L70, 70	/CONSTANT 0070	53
54	3818				54
55	3819	075167* 0000	ZBLOCK 11	/EMPTY SPACE	55
56					56
57					57
58					58

1	3852	075230*	0000	REGINIT, .-. .		/REGISTER INITIALIZE ROUTINE	1
2	3853	075231*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	2
3	3854	075232*	0000	A^10+0		/0A	3
4	3855	075233*	0030	030		/CHANNEL RESET	4
5	3856	075234*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	5
6	3857	075235*	0010	B^10+0		/0B	6
7	3858	075236*	0030	030		/CHANNEL RESET	7
8	3859	075237*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	8
9	3860	075240*	0002	A^10+2		/2A	9
10	3861	075241*	0020	020		/LOW-ORDER BITS ARE AFFECTED, NO DMA	10
11	3862	075242*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	11
12	3863	075243*	0012	B^10+2		/2B	12
13	3864	075244*	0000	000		/INTERRUPT VECTOR IS ALL ZEROES	13
14	3865	075245*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	14
15	3866	075246*	0004	A^10+4		/4A	15
16	3867	075247*	0104	104		/16X CLOCK RATE, 1 STOP BIT ASYNCHRONOUS	16
17	3868	075250*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	17
18	3869	075251*	0001	A^10+1		/1A	18
19	3870	075252*	0022	022		/RECEIVE AND TRANSMIT INTERRUPT ENABLE	19
20	3871	075253*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	20
21	3872	075254*	0011	B^10+1		/1B	21
22	3873	075255*	0004	004		/CONDITION AFFECTS VECTOR, NO B INTERRUPTS	22
23	3874	075256*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	23
24	3875	075257*	0003	A^10+3		/3A	24
25	3876	075260*	0301	301		/8 BITS, ENABLE RECEIVER	25
26	3877	075261*	4542	JMS I	PCOMLOAD/[COMLOAD]	/LOAD REGISTER	26
27	3878	075262*	0005	A^10+5		/5A	27
28	3879	075263*	0150	150		/8 BITS, ENABLE TRANSMITTER	28
29	3880	075264*	5630	JMP I	REGINIT	/RETURN	29
30							30
31							31
32							32
33							33
34							34
35							35
36							36
37							37
38							38
39							39
40							40
41							41
42							42
43							43
44							44
45							45
46							46
47							47
48							48
49							49
50							50
51							51
52							52
53							53
54							54
55							55
56							56
57							57
58							58

1	3994	/	WHILE THERE IS NO MORE ACTUAL CODE, THE FOLLOWING LINES ARE PRESENT TO	1
2	3995	/	CALCULATE THE MOVE LENGTH FOR THE CP-INTERRUPT HANDLER USED DURING THE	2
3	3996	/	LOOPBACK TEST IN MAIN MEMORY. THIS CALCULATION IS PROBABLY INCORRECT AND	3
4	3997	/	MOVES TOO MUCH CODE. THE ONLY POSSIBLE PROBLEM IS INTERACTION WITH APT	4
5	3998	/	DOWNLOADED PROGRAM CONVENTIONS.	5
6	3999			6
7	4000	075424* 0000	ZBLOCK .&7600+200-. /EMPTY SPACE IN THE REST OF THE PAGE	7
8	4001			8
9	4002	/	THIS IS WHERE THE STARTUP SECTOR CODE IS READ IN AS A CALLED SUBROUTINE.	9
10	4003			10
11	4004	075600* 0000	CALLSUB,.-. /TEST SUBROUTINE LOCATION	11
12	4005			12
13	4006	5601	CPEND2= . /MOVED CP INTERRUPT HANDLER CODE ENDS HERE?	13
14	4007			14
15	4008	075601* 0000	TSTBUFF,ZBLOCK 4 /KEYBOARD TEST BUFFER	15
16	4009			16
17	4010	5605	TSTBEND=. /END OF KEYBOARD TEST BUFFER	17
18	4011			18
19	4012	*6000	RELOC 6000 /EMPTY SPACE	19
20	4013			20
21	4014	6000	APTSTRT=. /APT ROUTINE STARTS HERE	21
22	4015			22
23	4016	*6400	RELOC 6400 /EMPTY SPACE	23
24	4017			24
25	4018	6400	WRIBUFF=. /WRITE BUFFER FOR SETUP	25
26	4019			26
27	4020	*6500	RELOC 6500 /EMPTY SPACE	27
28	4021			28
29	4022	6500	APTROUT=. /APT ROUTINE LOADS HERE	29
30	4023			30
31	4024	*6520	RELOC .+20 /EMPTY SPACE	31
32	4025			32
33	4026	6520	APTRO2= . /SECONDARY APT ADDRESS HERE	33
34	4027			34
35	4028	*5505	RELOC /UN-FOOL THE ASSEMBLER	35
36	4029			36
37	4030	075505 0000	ZBLOCK ROMGO-. /EMPTY SPACE	37
38	4031			38
39	4032	\$	/THAT'S ALL FOLK!	39
40				40
41				41
42				42
43				43
44				44
45				45
46				46
47				47
48				48
49				49
50				50
51				51
52				52
53				53
54				54
55				55
56				56
57				57
58				58

1		1
2		2
3	NO ERRORS DETECTED	3
4		4
5	NO LINKS GENERATED	5
6		6
7	8K MEMORY UTILIZED	7
8		8
9	3 FILES CREATED	9
10		10
11	700 SYMBOLS	11
12		12
13		13
14		14
15		15
16		16
17		17
18		18
19		19
20		20
21		21
22		22
23		23
24		24
25		25
26		26
27		27
28		28
29		29
30		30
31		31
32		32
33		33
34		34
35		35
36		36
37		37
38		38
39		39
40		40
41		41
42		42
43		43
44		44
45		45
46		46
47		47
48		48
49		49
50		50
51		51
52		52
53		53
54		54
55		55
56		56
57		57
58		58

