



CONCORDANCE FOR 1: <PSE-1301-1>IMPLOD.COM.2,25-Jun-87 11:22:27
2: <PSE-1301-1>IMPOPS.COM.2,25-Jun-87 11:23:56

A - 1:3,1:3,1:3,1:3,1:3,1:3,1:7,1:7,1:8,1:8,1:8,1:10,
1:10,2:7,2:98,2:98,2:105,2:122
ABUS 1:68 1:78
ACCRES 2:105 2:104
ACT - 2:32
ACTIVA 2:23 2:23,2:23
ADDR - 2:4,2:4,2:4
AGAIN - 2:7
ALLOC - 2:23
ALLOW - 2:32
AMEMIO 1:68 1:73,1:73,1:73,1:78
AMPCom 1:70 1:78,1:81
AMPMAN 1:68 1:73,1:78,1:78,1:78
AMPROC 1:70 1:78,1:81,1:82,1:82,1:93
AMPSYS 1:68 1:73
AMPTRY 1:68 1:73
AN - 2:32
ANOM 2:78 2:98,2:107
ANSWER 2:112 2:110,2:111,2:112
ANYPRO 1:16 1:92,1:109,1:112,1:114,1:117,1:117,2:36,2:68
APROC 1:68 1:78
AR - 1:23
ARE - 2:7
ARG1 - 1:4,1:4,1:5,1:5,1:9,1:9,2:4,2:4,2:4,2:15,2:15,2:20,
2:20,2:20,2:20,2:20,2:21,2:21,2:21,2:21,2:21,2:21,2:27,
2:27,2:32,2:32,2:32,2:32,2:32,2:33,2:33,2:33,2:33,
2:33
ARG2 - 1:4,1:4,1:5,1:5,1:9,1:9,2:4,2:4,2:4,2:15,2:15,2:21,
2:21,2:21,2:21,2:21,2:27,2:27,2:33,2:33,2:33,2:33,
2:33
ARG3 - 1:4,1:4,1:5,1:5,2:4,2:4,2:4,2:15,2:15,2:27,2:27
ARG4 - 1:4,1:4,1:5,1:5,2:15,2:15,2:27,2:27
ARG5 - 1:4,1:4,2:15,2:15,2:27,2:27
ARGRES 2:105 2:94,2:108,2:111
ARGUME - 2:29
ARSTAC - 1:26
ASCICO 2:76
ASCICR 2:76 2:108
ASCIFF 2:76
ASCILF 2:76 2:88,2:108
ASCISP 2:76 2:96
ASSUME - 2:29
ATTNPA 1:24 1:11,1:33
BACK - 2:7,2:7
BAD 2:125 2:125,2:129
BASE 2:124 2:124,2:128,2:129,2:130
BBCANS 1:17 1:81
BBCBAD 1:71 1:95
BBCBAK 1:17 1:82,1:93
BBCFOR 1:17 1:78,1:78,1:81
BBCLOK - 1:70,1:73,1:81,1:81,1:82,1:83,1:84,1:93,1:94,1:95
BBCMAP 1:17 1:82,1:93,1:93,1:94,1:94,1:94,1:94
BBCMSK 1:17 1:17,1:82,1:82,1:93,1:93,1:93,1:94,1:94,1:94,1:94,
1:94,1:95
BBCODD 1:17 1:93,1:93,1:94

| | | |
|--------|------|--|
| BBCPAS | 1:17 | 1:78, 1:78, 1:81 |
| BBCRES | 1:17 | |
| BBCRST | 1:71 | 1:94 |
| BBCWIN | 1:17 | 1:28, 1:79, 1:80, 1:82, 1:93, 1:93, 1:93, 1:94, 1:94, 1:94, . 1:94, 1:94, 1:95 |
| BBCWMK | 1:17 | 1:82, 1:93, 1:94, 1:94, 1:94, 1:94, 1:94, 1:94, 1:94, 1:94, 1:95 |
| BD | - | 1:23 |
| BDSTAC | - | 1:26 |
| BEG | 1:23 | |
| BEGIN | - | 2:107, 2:110 |
| BESCLK | 1:3 | |
| BITSNK | 1:19 | 1:52, 1:75, 1:90 |
| BITTAB | 1:67 | 1:46, 1:54, 1:54, 1:62, 1:63, 1:64, 1:67, 1:74, 1:74, 1:74, 1:74, 1:77, 1:80, 1:81, 1:81, 1:82, 1:82, 1:93, 1:102, 2:56, 2:56, 2:131 |
| BLANK | - | 2:29 |
| BLOCK | - | 2:7 |
| BLT | 1:87 | 1:49 |
| BLTO2 | 1:87 | 1:87 |
| BLTO3 | 1:92 | 1:92 |
| BLTO4 | 1:92 | 1:92 |
| BLTO5 | 1:92 | 1:92 |
| BLTO6 | 1:92 | 1:92 |
| BLTO7 | 1:92 | 1:92 |
| BLT10 | 1:91 | 1:92 |
| BLT11 | 1:91 | 1:91 |
| BLT12 | 1:92 | 1:91 |
| BLTACT | 1:71 | 1:86, 1:87, 1:87, 1:90, 1:108, 1:114, 1:114, 2:60, 2:60, 2:61, 2:107, 2:107, 2:110, 2:110, 2:117, 2:117, 2:118, 2:118 |
| BLTADD | 1:71 | 1:87, 1:90, 1:114, 2:68, 2:115 |
| BLTBBO | 1:93 | 1:93 |
| BLTBB1 | 1:93 | 1:93 |
| BLTBB2 | 1:93 | 1:95 |
| BLTBB3 | 1:93 | 1:94 |
| BLTBB4 | 1:93 | 1:94 |
| BLTBB5 | 1:94 | 1:93 |
| BLTBB6 | 1:94 | 1:93 |
| BLTBB7 | 1:94 | 1:94, 1:94, 1:94, 1:94 |
| BLTBB9 | 1:93 | 1:93 |
| BLTBBC | 1:93 | 1:92 |
| BLTBBK | 1:95 | 1:93 |
| BLTBBQ | 1:95 | 1:93, 1:93, 1:93 |
| BLTBBR | 1:95 | 1:94 |
| BLTBBS | 1:94 | 1:93 |
| BLTBBT | 1:95 | 1:94 |
| BLTBBY | 1:95 | 1:95 |
| BLTBBZ | 1:94 | 1:94 |
| BLTBCQ | 1:86 | 1:85, 1:95 |
| BLTBFA | 1:71 | 1:87, 1:87, 1:114, 2:68, 2:109, 2:116 |
| BLTBFM | 1:71 | 1:87, 1:114, 2:116 |
| BLTBGO | 1:86 | 1:86 |
| BLTBIT | 1:90 | 1:86 |
| BLTBMK | 1:71 | 1:86, 2:109, 2:112 |
| BLTBMQ | 1:86 | 1:85, 1:88, 1:88, 1:88, 1:90, 1:95 |
| BLTBPR | 1:89 | 1:86, 1:90 |
| BLTBU1 | 1:85 | 1:85 |
| BLTBU4 | 1:86 | 1:86, 1:86, 1:94 |

BLTBUD 1:85 1:92
BLTBUE 1:86 1:86, 1:91
BLTBUF 1:71 2:68
BLTBUG 1:86 1:86
BLTBUS 1:86 1:85
BLTBUT 1:86 1:86
BLTBUZ 1:86 1:86
BLTC1 1:88 1:89
BLTC2 1:89 1:91
BLTC3 1:88 1:85, 1:88, 1:88, 1:88, 1:91, 1:91, 1:91, 1:91
BLTC4 1:88 1:89
BLTC6 1:89 1:89, 1:89, 1:89
BLTC7 1:89 1:91
BLTCC 1:90 1:88
BLTCC1 1:90 1:90, 1:90, 1:90
BLTCCK 1:71, 1:98 1:90, 1:98, 1:107, 1:114, 2:107, 2:107
BLTCE 1:90 1:90, 1:90, 1:90
BLTCE1 1:90 1:88
BLTCE2 1:90 1:90, 1:90
BLTCEQ 1:88 1:90
BLTCF 1:88 1:88
BLTCOM 1:89 1:87
BLTCTL 1:71 1:86, 1:86, 1:89, 1:93, 1:94, 1:94
BLTDDT 1:71 1:87
BLTDID 1:71 1:95
BLTDO 2:117 2:107, 2:110, 2:117
BLTDON 1:71 1:86, 1:87, 1:88, 1:90
BLTDPI 1:71 1:86, 1:89, 1:114, 2:68, 2:115
BLTDPM 1:71 1:89, 1:90, 1:92, 1:114, 2:115
BLTDSK - 2:113
BLTDTY 1:71 1:87, 1:87, 1:114, 2:68, 2:115
BLTEDF 1:71 1:71, 1:71, 1:91, 2:113
BLTEDK 1:71 1:71, 1:71, 1:86
BLTEDQ 1:71 1:71, 1:71, 1:86, 2:113
BLTEND 1:90 1:90
BLTEOR 1:89 1:86, 1:87, 1:90
BLTERS 1:71, 1:98 1:98, 1:108, 1:110, 1:110, 1:111, 1:111, 2:109, 2:112,
2:112
BLTESF 1:71 1:71, 2:113
BLTESK 1:71 1:71, 2:113
BLTESQ 1:71 1:71, 2:113
BLTETO 1:71 1:71, 1:87, 2:112
BLTFAK 1:71 1:87, 1:114
BLTFDN 1:90 1:87
BLTFER 1:91 1:92, 1:92, 1:93
BLTFEX 1:71 1:87, 1:103, 1:104, 1:104, 1:114
BLTFGO 1:87 1:87
BLTFUL 1:71 1:87, 1:89, 1:90, 1:114, 2:107, 2:107
BLTGET 2:117 2:107, 2:107, 2:110, 2:110, 2:118
BLTLOG 1:71 1:89, 1:89, 1:117, 1:117, 2:115
BLTLOK - 1:71, 1:73, 1:87, 1:89, 1:114, 1:114, 1:114, 2:68, 2:68, 2:109,
2:112, 2:117, 2:117, 2:118, 2:118
BLTM1 1:91 1:91
BLTM2 1:91 1:91
BLTM3 1:91 1:91
BLTMAX 1:16 1:87, 1:87, 1:87
BLTME 1:91 1:92
BLTME1 1:91 1:91

| | | |
|---------|------------|---|
| BLTMK | - | 2:113 |
| BLTMSK | 1:89 | 1:86, 1:87, 1:88 |
| BLTMYB | 1:90 | 1:89, 1:89 |
| BLTMYC | 1:21 | 1:39, 1:45, 1:50, 1:51, 2:60, 2:96, 2:96, 2:119, 2:126, 2:126 |
| BLTMYM | 1:21 | 1:91, 1:91 |
| BLTMYR | 1:21 | 1:89, 1:91, 2:96, 2:96 |
| BLTMYS | 1:89 | 1:90 |
| BLTNLC | 1:71 | 1:87, 1:89, 1:109, 1:112, 1:117, 2:115 |
| BLTPCH | 1:92 | 1:87 |
| BLTPCO | 1:92 | 1:92, 1:92 |
| BLTPCS | 1:92 | 1:92 |
| BLTPOK | 1:71 | 1:90 |
| BLTPRM | 1:95 | 1:88, 1:91 |
| BLTPRO | 1:71 | 1:86, 1:89, 1:94, 1:95 |
| BLTPRS | 1:95 | 1:85, 1:93 |
| BLTPRX | 1:95 | 1:95, 1:95 |
| BLTRAT | 1:16 | 1:89, 1:114, 2:68, 2:115 |
| BLTRUN | 1:71 | 1:86, 1:93, 1:94, 2:61 |
| BLTSDF | 1:71, 1:98 | 1:88, 1:88, 1:98, 1:107, 1:110, 1:110, 1:114, 2:110, 2:110 |
| BLTSIZ | 1:71 | 1:87, 1:90, 1:114, 2:68, 2:109, 2:115 |
| BLTSPA | 1:71 | 1:89, 2:107, 2:110 |
| BLTSPM | 1:71 | 1:86, 1:92, 1:114, 2:68, 2:115 |
| BLTST | 1:71 | 1:86, 1:86, 1:87, 1:88, 1:89, 1:93, 1:94, 1:108, 1:110, 1:114, 1:114, 2:60, 2:68, 2:68, 2:107, 2:110, 2:110, 2:112, 2:117, 2:117, 2:117, 2:118, 2:118 |
| BLTSTB | 1:16 | 1:46 |
| BLTSTY | 1:71 | 1:87, 1:87, 1:114, 1:114, 2:68, 2:115 |
| BLTTCH | 1:87 | 1:92, 1:92, 1:92, 1:92 |
| BLTTO | 1:71 | 1:87, 1:89, 1:114, 2:68, 2:115 |
| BLTTOG | 1:89 | 1:86, 1:90 |
| BLTX | 1:89 | 1:87, 1:87 |
| BSADIL | 1:25 | 1:64, 1:73, 1:79, 1:80, 1:81, 1:93, 1:102 |
| BSADIT | 1:3 | 1:11, 1:25 |
| BSADM | 1:25 | 1:73, 1:81 |
| BSADMT | 1:3 | 1:11, 1:25 |
| BSADRM | 1:25 | 1:25, 1:81, 1:81 |
| BSADRS | 1:25 | 1:11, 1:25, 1:64, 1:74, 1:80, 1:81, 1:81, 1:93, 1:102, 1:102 |
| BSMAPM | 1:69 | 1:81, 2:38, 2:38, 2:39, 2:40, 2:43 |
| BSMAPS | 1:69 | 1:11, 1:74 |
| BSMPTB | 1:3 | 1:11, 1:69 |
| BSMPTM | 1:3 | 1:11, 1:69 |
| BUFEND | 1:97 | 1:104 |
| BUFFLG | 1:72 | 2:41 |
| BUFSIZ | 2:78 | 2:78, 2:90, 2:91 |
| BUSCON | 1:70 | 1:27 |
| BUSFIX | 1:70 | 1:27 |
| BUSINC | 1:17 | 1:102 |
| BUSKIL | 1:25 | 1:33, 1:64, 1:73, 1:73, 1:73 |
| BUT | - | 2:32 |
| CABORT | 1:96 | 1:107, 1:111 |
| CARRY | - | 1:31, 1:35, 2:70 |
| CCADDR | 1:97 | 1:107, 1:107 |
| CCKTAB | 1:68 | 1:88, 1:88 |
| CCCLEAR | 2:76 | |
| CCPIEC | 1:97 | |
| CCSIZE | 1:97 | |
| CCTYPE | 1:97 | 1:107 |

CD - 1:23
CDOWN 2:76
CDSTAC - 1:26
CFORHA 1:97 1:97
CFORHS 1:97
CFORI 1:97
CFORL 1:97
CFORMI 1:97
CHANGE - 2:6,2:7
CHECK - 2:7,2:29
CHKH 1:97 1:103
CILBUF 1:29 1:29
CILCNT 1:29 1:42,1:42,1:43
CILEND 1:29 1:41,1:41
CILLOC 1:29 1:43
CILNUM 1:16 1:29,1:29
CILOPS 1:29 1:41,1:41
CILOVF 1:29 1:42,1:42
CILPRO 1:29 1:42,1:42,1:43
CILREG 1:29 1:43
CKLOCK - 1:58
CKPASS 1:16 1:12,1:12,1:12,1:12,1:66,1:90
CKSUB 1:65 1:52,1:60,1:60,1:66,2:36,2:45
CKSUM 1:28 1:12,1:60,1:118,2:37,2:44,2:44,2:45,2:50,2:51
CLOKRT 1:19 1:36,2:62,2:62
CLST 1:5 1:5,1:6,1:6,1:12
CMAP 1:72 2:38,2:51
CMARK 1:97 1:97,1:106,1:107,1:112,1:112
CMMBEG 1:97
CMMBIT 1:97
CODE - 1:13,1:14,1:14,1:69,2:13,2:34
CODEPA 1:5 1:13,2:34
COMAR 1:70 1:12,1:76,1:77,1:77
COMMA 2:104 2:101,2:104
COMMON - 2:7
COMPTR 1:28 1:47,1:47,1:59,1:59
COMREL 1:28 1:48,1:60,1:61,1:61,1:118,2:51
COMSTS 1:16 1:12
COMTST 1:28 1:59,1:59,1:59,1:59
CONCOM 1:27 1:27,1:75
CONSOL 1:22 1:45,1:52,1:105,2:128
CONTAB 1:27 1:27,1:27,1:49,1:63,1:63,1:63
CONTLO 1:12 2:139
CORETU - 2:92
COUBUS 1:70 1:79,1:79
COUCON 1:70 1:27
COUFIX 1:70 1:27
COUNT - 2:4,2:4
COUPLR 1:17 1:78
COUTAB 1:70 1:70,1:79,1:79,1:79,1:91,1:93,1:95
COUTBL 1:70 1:70,1:79
CPAGE - 2:35
CPCORE 1:96 1:96,1:107,1:112
CPDH 1:97
CPDL 1:97 1:107
CPKADD 1:97
CPKSIZ 1:97
CPKSSF 1:97

CPLINE 1:97
CPMASK 1:96 1:106
CPSATD 1:97 1:104
CPSATS 1:97 1:104
CPSIZE 1:97
CR 2:107 2:100,2:106,2:108
CRASH - 1:37
CRIGHT 2:76
CRLFTX 2:108 2:107
CRPBIT 1:97
CRPCNT 1:97
CSETPUP 1:96 1:96,1:101,1:106,1:112
CSRFLG 1:97
CTYPE 1:97 1:97
CUMBIT 2:79
CUP 2:76
CURADD 2:80 2:104,2:106,2:110,2:115
CURREN - 2:7,2:7
CWPCIN 1:12 1:11,1:63
D2FB 2:78 2:82,2:83,2:119,2:119,2:119
D2FL - 2:78,2:82,2:83,2:119,2:119,2:119
D2FPOK 2:78 2:119
DATA - 1:97
DD - 2:76
DDACCU 2:80 2:98,2:104,2:104,2:104,2:105,2:105,2:107
DDIENT 2:94 2:94
DDIVAR 2:80
DDPOKE 2:78 2:107,2:110,2:117
DDRESE 2:93 2:93,2:94
DDSTAC - 2:94
DDT - 2:34,2:34
DDTATT 1:33 1:15,1:33
DDTBFI 2:80 2:107,2:109
DDTBUF 2:80 2:107,2:112,2:116
DDTCOD 2:34 2:82,2:91
DDTDIS 2:100 2:98,2:103
DDTGET 2:119 2:98,2:120
DDTINI 2:92 2:92
DDTLOD 2:34 2:34
DDTLOK - 2:78,2:94,2:94
DDTOUT 2:119 2:102,2:119,2:121,2:121
DDTPOL 2:94 2:81
DDTPRO 2:80 2:93,2:110,2:115
DDTSP 2:76 2:94,2:94
DDTSTE 2:96 2:96,2:120
DDTVAR 2:34 2:76,2:80
DDTVST 2:34,2:34 2:34,2:34,2:34
DDVARE 2:80 2:93
DDVARS 2:80 2:93
DEBUGM 1:24 1:45,1:48,2:60,2:70,2:82,2:107,2:119,2:126
DECIMA 2:104 2:101,2:104
DECNUM 2:80 2:104,2:104,2:105,2:105,2:105
DEFATN - 1:10,1:10,1:15
DEFBEG - 1:10,1:13
DEFBUS 1:3 1:13
DEFENT 1:10 2:35
DEFILL - 1:10,1:10
DEFINT 1:8 1:8,1:8,1:8,1:8,1:13

DEFIO 1:7 1:13
DEFLIN 1:8 1:13, 1:15
DEFLOO - 1:10, 1:10, 2:125
DEFMO 2:126 2:126, 2:126
DEFM1 1:6 1:6, 1:6, 1:6, 1:6
DEFMAC 1:10 1:10, 1:10, 1:10, 1:10, 1:10
DEFMEM 1:4 1:13, 1:14
DEFPAG - 1:14, 1:14, 1:14, 1:14, 1:14, 1:14, 1:25, 1:69, 2:34,
2:34, 2:34
DEFPCN - 1:10, 1:10
DEFREL 1:7 1:13
DEFRLD 1:9 1:13, 1:15
DEPOSI 2:107 2:107
DERRPR 2:109 2:107, 2:107, 2:109
DESVPA 1:5
DEVINC 1:17 1:72, 1:102, 1:102, 1:116
DEVTYP 1:17 1:102
DHALT 1:29 1:39, 2:126
DHPASS - 1:37, 1:39, 2:126
DIGIT 2:105 2:102, 2:102, 2:102, 2:102, 2:102, 2:102, 2:102, 2:102,
2:102, 2:105
DINIT 1:29 1:42
DISMIS 2:22 2:22
DISPLO 2:98 2:95
DLST 1:5 1:5, 1:6
DMAP 2:80 2:93, 2:110, 2:115
DO1.6 2:136 2:136
DO25.6 2:136 2:136
DOCLOC 2:136 2:136, 2:136, 2:136
DOINIT 2:133 2:133
DON - 2:6, 2:7
DOT 2:104 1:11, 1:11, 2:101
DSLEEP 2:94 2:94, 2:117, 2:118, 2:119, 2:120, 2:120
DSPFLA 2:78
DSPPOK 2:78 2:89
DSTAND 1:50 1:13, 1:33
DSTH 1:97 1:106, 1:107
DSYSSP 2:76 2:94, 2:94
DUMMY 1:14 1:11, 1:12, 1:12, 1:12, 1:17, 1:29, 1:96, 2:7, 2:30, 2:30, 2:35,
2:76, 2:78
DXTCHK 2:82 2:81, 2:83, 2:84
DXTFLA 2:78 2:82, 2:83
EOOO - 1:13
E1 - 1:10, 1:10
E100 - 1:13
E2 - 1:10, 1:10
E200 - 1:13
E3 - 1:10, 1:10
E4 - 1:10, 1:10
EMTPID 1:18
EMTY 2:128
END 1:23 2:7, 2:107, 2:111
ENDI 1:96 1:100, 1:100, 1:104
ENDINI 2:18
ENDO 1:96 1:103
ENTER - 2:6, 2:7
ENTLIS 1:10, 1:10 1:11
EQUAL - 1:34, 1:52, 1:55, 1:60, 1:60, 1:61, 1:74, 1:75, 1:79, 1:79,

1:79, 1:102, 1:102, 1:102, 1:102, 1:103, 2:36, 2:36, 2:38,
2:45, 2:56, 2:64, 2:65, 2:82, 2:83, 2:87, 2:92
ERRPRI 2:112 2:109, 2:112, 2:113
ETAB 2:113 2:113, 2:113
EXAMIN 2:110 2:111
EXIT 2:19, 2:31 2:19, 2:27, 2:32
F.IILLE 2:13
F.LOCK 2:13
F.QUEU 2:13
F.RQUE 2:13
F.RUNN 2:13
F.WAIT 2:13
FOOO - 1:13
F1 - 1:50
F100 - 1:13
F2 - 1:95
F200 - 1:13
F2DB 2:78 2:82, 2:82, 2:120, 2:120
F2DL - 2:78, 2:82, 2:82, 2:120, 2:120, 2:120
F2DPOK 2:78 2:120
F2TPOK 2:78 2:89
FADTEX 2:97 2:97
FALSE - 2:17, 2:17, 2:29, 2:29
FCB - 2:13, 2:13
FCBCCHA 2:13
FCBCOD 2:13
FCBREL 2:13
FCBSP 2:13
FCBSTA 2:13
FCBSTK 2:13
FCBTIM 2:13
FIELD 2:32
FINDDT 2:81 2:81, 2:81
FINDE1 2:131 2:131
FINDE2 2:131 2:131
FINDEV 2:131
FINDEX 2:131 2:131
FIXJIF 2:66 2:50, 2:51, 2:65, 2:66
FIXTAB 1:27 1:63, 1:63, 1:63, 1:63
FNDCLK 1:64 1:35, 1:53, 1:64
FOO - 2:78, 2:85, 2:89, 2:89, 2:93, 2:93, 2:104, 2:105, 2:105, 2:105, 2:105, 2:105, 2:120
FORK 2:23
FORMER 2:114 2:113
FTRAPV 1:15
FVSTAR 2:87 2:85
G - 1:107
GET - 2:7
GO - 2:7, 2:7
GOSJ6 2:128 2:126, 2:126
HALTUS 1:50 1:33, 1:35
HDCBIT 2:78
HERALD 2:95 2:94
HEXLET 2:105 2:102, 2:102, 2:102, 2:102, 2:102, 2:105
HEXOUT 2:121 2:121, 2:122
HMODID 1:96 1:102
HOTLIM 1:24 1:68, 2:36, 2:68
HRDOFF 1:96 1:103, 1:105

HTSBIT 2:79

I - 1:23
IBUFL 1:20 1:20, 1:40, 2:63, 2:63
ID - 1:23
IDDTOU 2:119 2:96, 2:96, 2:106
IDLEC 2:124 2:125
IDSTAC - 1:26
IHOTLM 1:68 2:36
ILIS - 1:3, 1:3, 1:3
ILLBUF 1:41, 1:41 1:41, 1:41, 1:44
ILLC 1:37 1:37
ILLCNT 1:41 1:40, 1:42, 2:63
ILLCOP 1:43 1:41, 1:43, 1:44
ILLEND 1:41, 1:41 1:41, 1:41
ILLOPA 1:24 1:11, 1:37
ILLOPO 1:24 1:41
ILLPO 1:37 1:53
ILLP1 1:37 1:53
ILOCKT 1:73 1:73, 1:75
ILOPTX 2:97 2:97
IMEMT 2:37 2:37, 2:50
IMP.PK - 1:15
IN - 2:6, 2:7
INBAS2 2:129 2:135
INBASE 2:129
INDIRE - 2:68, 2:68
INDVAR 1:6 1:11
INI - 1:5, 1:7, 1:7, 1:7, 1:7
INICON 1:72 1:27, 2:66
INIFIX 1:72 1:27, 2:66
INIRAT 1:16 1:77
INITFO 2:18
INITLI 2:134 2:6, 2:134
INITPI 2:135 2:6, 2:135
INNER 1:10 1:10
INTABL 1:8 1:53
INTER 1:53 1:52
INTIME 1:28 1:51, 2:37, 2:49, 2:50, 2:50, 2:50, 2:50, 2:52, 2:52, 2:52,
2:52, 2:54, 2:55, 2:81, 2:81, 2:81, 2:81, 2:92, 2:92
IOBASE 1:69 1:11, 1:69, 1:102, 2:56, 2:131
IOBTAB 1:7 1:11, 1:69
IOCON 1:72 1:27, 2:58
IOCTAB 1:70 1:70, 1:79, 1:79
IOCTBL 1:70 1:79
IOFIX 1:72 1:27, 2:58, 2:58, 2:58
IOKILL 1:69 2:56
IOMASK 1:17 1:82, 1:102, 2:56
IRET 1:19 1:37, 1:37, 1:37, 1:37
IS - 2:6, 2:7
ISTACK - 1:33, 1:35, 1:37
IT - 2:7
ITABLE 2:7 2:7
ITEXTO 2:121 2:94, 2:96, 2:105, 2:107, 2:112, 2:113
IX 1:19 1:33, 1:33, 1:34, 1:34, 1:35, 1:35, 1:36, 1:36, 1:37, 1:37,
1:37, 1:38, 1:40, 1:40
JLOOP 1:12 1:11, 1:45
JPOLL 1:24 1:45, 2:81
JTIME 1:22 1:35, 1:36, 2:62

| | | |
|---------|-----------|---|
| JUMP | - | 2:7,2:7 |
| KERLIM | 1:24 | 1:68 |
| L | - | 1:23 |
| L\$1.6 | 1:24 | 1:11,2:136 |
| L\$25.6 | 1:24 | 1:11,2:136 |
| L1 | - | 1:8,1:8,1:8,1:8,1:8,1:8,1:8 |
| L2 | - | 1:8,1:8,1:8,1:8,1:8,1:8,1:8 |
| L3 | - | 1:8,1:8,1:8,1:8,1:8,1:8,1:8 |
| L4 | - | 1:8,1:8,1:8,1:8,1:8,1:8,1:8 |
| LASTNU | 2:80 | 2:121 |
| LBASE | 2:124 | |
| LBEG | - | 1:23 |
| LC | - | 1:23 |
| LCILBU | - | 1:29,1:29,1:41,1:43,1:43,1:44,1:44 |
| LCKTAB | 1:68 | 1:91 |
| LCLOCK | 1:22 | 1:30,1:30,1:35,1:36,1:36,1:45,1:46,1:46,1:47,1:64, 1:64,1:116,2:66,2:128 |
| LCLSTK | 1:14 | 1:23 |
| LCODE | 1:14 | 1:11,1:12,1:24,1:30,1:99,2:16,2:28,2:64,2:81,2:81, 2:90,2:121,2:125 |
| LCOMAR | 1:12 | 1:76,1:77 |
| LCOML | 1:27 | 1:75 |
| LCSTAC | - | 1:26 |
| LEFT | - | 2:7 |
| LEN | - | 1:23,1:23 |
| LEVEL1 | 1:33 | 1:13,1:15 |
| LEVEL2 | 1:34,1:34 | 1:13 |
| LF | 2:106 | 2:100,2:106 |
| LIKE | - | 2:32 |
| LIM | - | 1:5,1:6 |
| LIMEMT | 2:37 | 2:50 |
| LIMIT | - | 1:13,1:14,1:14,1:14,1:14,1:14,1:14,1:25,1:69,2:34, 2:34,2:34,2:34 |
| LIMTAB | 1:68 | 1:6,1:11,1:118,2:50 |
| LIS | - | 1:7,1:7 |
| LIS1 | - | 1:7,1:7 |
| LIST | - | 1:8,1:8 |
| LK | - | 1:23 |
| LKERCK | 1:24 | 1:12,1:52,1:73,1:77 |
| LKEREN | 1:12 | 1:24 |
| LKERP | 1:25 | |
| LKPATC | 1:25 | |
| LKPLEN | 1:16 | 1:25,1:25 |
| LKSTAC | - | 1:51 |
| LLOCKT | 1:73 | 1:75 |
| LMAP | 1:20 | 1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:11,1:51,1:88, 1:89,1:89,2:38,2:39,2:41,2:64,2:133,2:134 |
| LMAPTB | 1:6 | 1:11,1:20 |
| LMDCON | 1:27 | 1:28,1:57,1:57 |
| LMMBAS | 1:69 | 2:43 |
| LOCALC | 1:24 | 1:12,2:36,2:36,2:68 |
| LOCBEG | - | 1:10,1:24 |
| LOCBLT | 2:68 | 2:60,2:61,2:68 |
| LOCCON | 1:72 | 1:27 |
| LOC CST | 1:14 | 1:14,1:14,1:14 |
| LOCDEF | - | 1:28,1:28,1:28,1:28,1:70,1:70,1:70,1:71,1:71,1:72, 1:72,1:72,1:72,1:98,2:78,2:78,2:78,2:78 |
| LOCEND | 1:14 | 1:24,1:52,1:68,2:36 |

LOCFAD 1:18 1:37,2:96,2:97,2:120
LOCFIX 1:72 1:27,2:60,2:60
LOCILL 1:20 1:40,1:40,2:63,2:63,2:63
LOCILLO 1:18 1:37,2:96,2:97,2:120
LOCIPT 1:20 1:40,1:40,2:63
LOCKDE 2:9
LOCOPE 2:80 2:107,2:108,2:111
LOCQUT 1:18 1:32,2:96,2:97,2:120
LOCSSST 1:14 1:14,1:14
LOCSTS 1:27 1:48
LOCVST 1:14 1:14
LOCZEL 1:22 1:51
LOCZER 1:22 1:22,1:51
LOD - 1:5,1:6,1:6,1:12
LODINT 1:8 1:19
LOOP1 2:128 2:126,2:127
LOOP2 2:128 2:128
LOOP3 2:128 2:128
LOOP4 2:128
LOOP5 2:126 2:126
LOOPE 2:127 2:126,2:126,2:126,2:127,2:127,2:127
LOOPM 2:126 2:128
LOOPMV 2:126 2:19,2:20,2:31,2:32,2:125
LOWMSK 1:96 1:103,1:103,1:103,1:103
LPSBTA - 2:85
LST - 1:5,1:5
LSTACK - 1:45,2:62,2:126,2:126,2:128
LSTKLN 1:14 1:14,1:23,2:62,2:62,2:126,2:126,2:126,2:126
LSYSFC - 2:15
LTIME 1:22 1:36,1:46,1:64,2:66,2:128
LVARS 1:14 1:11,1:19
M% - 2:7
M%LOCA 2:6
MO - 1:11,1:14,1:14,1:28,1:38,1:39,1:45,1:49,1:49,1:57,
1:57,1:57,1:91,1:91,1:92,2:64,2:65,2:65,2:84,2:94,
2:131,2:133,2:133
M1 - 1:6,1:14,1:29,1:34,1:39,1:40,1:47,1:57,1:58,1:58,1:60,
1:60,1:60,1:67,1:68,1:68,1:68,1:88,1:88,2:9,2:34,2:37,
2:37,2:41,2:44,2:44,2:45,2:45,2:45,2:49,2:50,2:50,
2:50,2:50,2:50,2:51,2:51,2:51,2:52,2:52,2:53,2:54,
2:55,2:84,2:84,2:94,2:94,2:126,2:133
M2 - 1:14,1:47,1:47,1:47,1:48,1:48,1:60,1:60,1:67,1:74,
1:80,1:81,1:81,1:81,1:91,1:114,1:118,2:34,2:45,
2:49,2:52,2:109,2:115,2:116,2:117,2:117,2:117,2:117,
2:117,2:118,2:118,2:118,2:126,2:129,2:129,2:133
M3 - 1:49,1:49,1:51,1:90,1:91,1:91,1:101,1:101,1:106,1:106,
1:106,1:107,1:110,1:110,1:111,1:115,1:115,1:116,2:9,
2:34,2:50,2:84,2:94,2:107,2:109,2:109,2:109,2:109,
2:110,2:112,2:112,2:115,2:115,2:115,2:115,2:115,2:115,
2:115,2:115,2:116,2:116,2:117,2:117,2:117,2:117,2:117,
2:117,2:118,2:118,2:119,2:120,2:133
MAGMOD 1:96 1:102
MAP - 1:6,1:14,1:69,2:6,2:16,2:28,2:34
MAPBIT 2:80 2:110,2:110,2:115
MAPCOD - 2:126,2:126
MAPCOM 1:20 1:47,1:47,1:49,1:51,1:55,1:55,1:59,1:59
MAPDDT - 2:81,2:116,2:126
MAPMSK 1:17 1:58,2:127

MAPREL - 1:45, 1:48, 1:49, 1:49, 1:75, 1:103, 1:105, 2:45, 2:64, 2:65,
 2:65, 2:117, 2:117, 2:118, 2:118, 2:131
 MAPV2 - 2:126, 2:127, 2:128, 2:129, 2:133
 MAPVAR - 1:34, 1:39, 1:49, 2:63, 2:81, 2:109, 2:112, 2:117, 2:118, 2:126,
 2:126, 2:133
 MAXSTR 1:7 1:11, 1:24
 MBLKS 2:124 2:129, 2:130
 MBUSY 1:96 1:103, 1:104
 MD - 1:23
 MDSTAC - 1:26
 MEMCON 1:72 1:27, 2:60
 MEMDIS 1:77 1:75, 1:77
 MEMFIX 1:72 1:27
 MEMKIL 1:25 1:47, 1:54, 1:77
 MEMRAT 1:16 1:54
 MEMSEG 1:28 1:55, 1:55, 1:67, 1:74
 MEMTOT 1:28 1:55, 2:38
 MEMTST 1:67 1:60, 1:67, 2:38, 2:47, 2:53
 MENDF 1:96 1:103
 MESSID 1:97
 MIDH - 1:97
 MINPRO 2:66 1:11, 1:11, 2:66
 MLIMX 1:6 1:6
 MLIS - 1:3, 1:3, 1:3
 MM - 1:23
 MMAXST 1:24 1:11, 1:36
 MMLIMS 1:6 1:11, 1:68
 MMSTAC - 1:26
 MNCODE 1:6 1:11, 1:11
 MNOVAR 1:6 1:11
 MNVARS 1:6 1:6, 1:11
 MODID 1:96 1:102, 1:102
 MP - 2:4, 2:4, 2:4, 2:4, 2:4
 MSGBIT 2:78
 MSTRIP 1:7 1:13
 MYPROC 1:19 1:79, 1:80, 1:89, 1:91
 MYSEGS 1:22 1:54, 1:55, 2:41
 N - 2:27, 2:27, 2:27
 NAME 2:9, 2:16, 2:20, 2:30, 2:32 1:5, 1:5, 1:5, 1:5, 1:5, 1:5, 1:5, 1:5,
 1:5, 1:5, 1:5, 1:5, 1:5, 1:5, 1:5, 1:5, 1:12, 1:12, 1:12, 1:23, 1:23,
 1:23, 1:23, 2:6, 2:6, 2:6, 2:6, 2:6, 2:9, 2:9, 2:15, 2:15,
 2:15, 2:16, 2:16, 2:16, 2:16, 2:19, 2:19, 2:20, 2:22, 2:22,
 2:23, 2:23, 2:23, 2:23, 2:23, 2:27, 2:27, 2:27, 2:27, 2:27,
 2:27, 2:28, 2:28, 2:28, 2:30, 2:30, 2:31, 2:32
 NCODEM 1:69 1:11
 NCODEP 1:69 1:11, 1:88, 1:88, 1:89, 1:89, 1:89, 2:38, 2:41, 2:41, 2:44,
 2:45, 2:45, 2:47, 2:47, 2:65
 NDVARS 1:69 1:11, 2:41
 NEQUAL - 2:52
 NETH 1:97
 NEW - 2:7, 2:7
 NEWCOM 1:22 1:54, 1:59, 1:59
 NEWPKC 1:1 1:109, 1:112, 1:117
 NFH 1:96 1:101
 NIBUF 1:20 1:20, 1:40, 2:63
 NLOCST 1:26 1:63, 1:63
 NMSEG 1:4 1:16, 1:22, 1:25, 1:28, 1:47, 1:47, 1:48, 1:54, 1:55, 1:60,
 1:61, 1:61, 1:67, 2:39, 2:40, 2:43

NOPID 1:18 1:24
NOPIDS 2:125 2:125
NOPTVM 1:69 1:11
NOSAVE - 1:32, 1:33, 1:36, 1:44, 1:44, 1:51, 2:81, 2:125, 2:125
NOTEEXT 2:114 2:96, 2:97, 2:113, 2:113
NOTRAP - 1:57, 1:57, 1:58, 1:59, 1:59, 1:59
NOVARP 1:69 1:11, 2:39
NOW - 2:7
NPROC 1:16 1:70
NQUIT - 1:40, 1:49, 1:56, 1:58, 1:58, 1:59, 2:45
NREQUP 1:69 2:40, 2:41
NSEGS 1:69 1:69, 1:72, 2:56
NSITRY 1:100 1:100
NSPARM 1:69 1:11
NSPARP 1:69 1:11, 2:41, 2:47, 2:50
NSUER 1:20 1:20, 1:40, 1:43
NUMOUT 2:121 2:96, 2:96, 2:106, 2:112, 2:113, 2:121
NVARSM 1:69 1:11
NVARSP 1:69 1:11
NXISTX 2:114 2:113
OF - 2:7
OFF - 2:7
OFFDIS 1:24 1:63
OLD - 2:7
OLDP 1:19 2:128, 2:128
OLST 1:5 1:5, 1:6
ONCE - 2:7
ONE - 2:7, 2:7
OPTV - 1:13, 1:14
OPTVPA 1:5 1:14, 1:14, 1:14, 1:14, 1:25, 1:69, 2:34, 2:34, 2:34
ORG - 2:7, 2:7, 2:7
OUR - 2:7, 2:7, 2:7
OUTER 1:10 1:10
OVRBIT 2:79 2:98, 2:107
OVRRID 2:100 2:98
P 1:6 1:6, 1:6, 1:6, 1:6
P\$ - 2:27, 2:27
P\$1.6 1:13 1:11
P\$25.6 1:13 1:11
P\$LOOP - 1:10, 1:11
P\$PCNT - 1:10, 1:11
P% - 2:7, 2:7
PACKM - 1:89, 1:89, 1:91, 1:100, 1:100, 2:54, 2:115, 2:131
PAGE - 1:11, 1:11, 1:12, 1:12, 1:12, 1:12, 1:12, 1:12, 1:17, 1:19,
1:23, 1:24, 1:28, 1:29, 1:29, 1:30, 1:68, 1:70, 1:73, 1:96,
1:98, 1:99, 1:101, 2:7, 2:7, 2:7, 2:7, 2:7, 2:7, 2:7, 2:7, 2:7,
2:7, 2:7, 2:9, 2:9, 2:16, 2:16, 2:28, 2:28, 2:35, 2:35, 2:35,
2:64, 2:66, 2:70, 2:70, 2:76, 2:76, 2:78, 2:78, 2:78, 2:80,
2:80, 2:81, 2:82, 2:90, 2:91, 2:121, 2:124, 2:124, 2:125
PAGEBC 1:28 1:68, 2:34, 2:37, 2:37
PAKTYP 1:97 1:104
PATCHI - 2:7
PATTN - 1:10, 1:11
PC - 2:94
PCHALT 1:18 1:18, 1:32, 1:37, 1:37, 1:39, 1:50, 1:50, 1:71, 1:85, 1:86,
1:89, 1:93, 1:94, 2:126, 2:126
PCRUN 1:18 1:18, 1:50, 1:51, 1:71, 1:86, 1:86, 1:89, 1:94, 1:94
PCSTEP 1:18 1:85

| | | |
|--------|------------|--|
| PDBBLK | 2:14 | 2:14 |
| PDBCHA | 2:26 | |
| PDBLIM | 2:14 | |
| PDBOFF | 2:14 | |
| PDBPID | 2:26 | |
| PDBPKO | 2:14 | |
| PDBRAT | 2:14 | |
| PFX | 2:80 | 2:98, 2:104, 2:105, 2:106, 2:109, 2:110 |
| PG | - | 1:6, 1:6, 1:11, 1:11, 1:11, 2:7, 2:7, 2:7, 2:7, 2:7 |
| PG\$ | - | 1:5, 1:5 |
| PGINIT | 1:28 | 1:11, 2:37, 2:64 |
| PHYSIC | - | 1:13, 1:14, 1:14, 1:69, 2:34, 2:34 |
| PID | 1:19 | 1:74, 1:90, 2:21, 2:33, 2:88, 2:89, 2:119, 2:120, 2:136 |
| PID3 | 1:24 | 1:74 |
| PIDGET | 1:21 | 1:21, 1:74, 1:74, 2:56, 2:128 |
| PIDRCL | 1:17 | 1:74, 1:74, 2:56, 2:56 |
| PIDSTO | 1:17 | 1:74 |
| PIDTOT | 1:21 | 1:74, 2:56 |
| PILEND | 1:29 | 1:41 |
| PILLOP | - | 1:10, 1:11 |
| PILLOV | 1:44, 1:44 | 1:24, 1:44 |
| PILNUM | 1:29 | 1:29, 1:44 |
| PILOPS | 1:29 | 1:41 |
| PILOPV | 1:29 | 1:44, 1:44 |
| PKCACT | 1:98 | 1:101, 1:101, 1:101, 1:106, 1:107, 1:107, 1:109, 1:112, 1:116, 1:117 |
| PKCADD | 1:98 | 1:105, 1:107, 1:108, 1:108, 1:108, 1:110, 1:112, 1:117 |
| PKCBFA | 1:98 | 1:108, 1:110, 1:114 |
| PKCBLT | 1:114 | 1:109, 1:112 |
| PKCCLA | 1:98 | 1:98, 1:116 |
| PKCCLL | 1:98 | 1:116 |
| PKCCLR | 1:115 | 1:108, 1:111, 1:115 |
| PKCDON | 1:98 | 1:108, 1:110, 1:114 |
| PKCETY | 1:98 | 1:109, 1:111, 1:111, 1:112 |
| PKCEXT | 1:98 | 1:101, 1:115, 1:117 |
| PKCFHA | 1:98 | 1:106 |
| PKCFHO | 1:98 | 1:106 |
| PKCFID | 1:98 | 1:106, 1:116 |
| PKCFIM | 1:98 | 1:106 |
| PKCFLN | 1:98 | 1:106, 1:106, 1:107 |
| PKCIC | 1:106 | 1:104, 1:104 |
| PKCIID | 1:101 | 1:115 |
| PKCIST | 1:98 | 1:104, 1:106, 1:108, 1:109 |
| PKCITB | 1:101 | 1:101, 1:116, 1:116 |
| PKCITL | 1:101 | 1:116 |
| PKCLEN | 1:98 | 1:108, 1:108, 1:108, 1:110, 1:112, 1:117 |
| PKCLHA | 1:98 | 1:101, 1:106, 1:106, 1:107 |
| PKCLHO | 1:98 | 1:101, 1:106, 1:107 |
| PKCLID | 1:98 | 1:106, 1:107, 1:110, 1:115 |
| PKCLIM | 1:98 | 1:106, 1:106, 1:107, 1:107 |
| PKCLLN | 1:98 | 1:106, 1:106 |
| PKCLMK | 1:97 | 1:108, 1:114 |
| PKCLOK | - | 1:98, 1:101, 1:101, 1:116, 1:117 |
| PKCLTO | 1:96 | 1:107, 1:108 |
| PKCMAX | 1:98 | 1:112, 1:112 |
| PKCMYI | 1:98 | 1:104, 1:116 |
| PKCNIM | 1:98 | 1:104 |
| PKCOC | 1:110 | 1:103 |

PKCORE - 1:15
PKCOST 1:98 1:110, 1:110, 1:112
PKCOTM 1:98 1:112, 1:112
PKCRAT 1:96 1:101, 1:117
PKCRCR 1:98 1:101, 1:101, 1:104, 1:108, 1:115, 1:116, 1:116
PKCSRF 1:98 1:111, 1:112, 1:112, 1:112, 1:117
PKCSSF 1:98 1:101, 1:107, 1:108, 1:109, 1:110, 1:111, 1:117
PKCST 1:98 1:101, 1:101, 1:101, 1:104, 1:106, 1:107, 1:107, 1:107, 1:108,
1:108, 1:109, 1:109, 1:110, 1:110, 1:110, 1:110, 1:111, 1:112,
1:112, 1:112, 1:114, 1:115, 1:115, 1:116, 1:117
PKCSTD 1:96 1:110, 1:113
PKCTIM 1:98 1:101, 1:115
PKCTMK 1:97 1:107, 1:109, 1:114
PKCTMX 1:96 1:104, 1:108
PKCTRY 1:96, 1:96 1:116
PKCTYH 1:98 1:101
PKCTYP 1:98 1:107, 1:112, 1:117
PKCUPT 1:115 1:101, 1:107, 1:108, 1:110, 1:113, 1:117
PKTH 1:97 1:97, 1:106, 1:106, 1:106, 1:107, 1:107
PLRCOM 1:98 1:109, 1:109, 1:112, 1:112, 1:117
PLRPKC 1:98 1:109, 1:112, 1:117, 1:117
POKE 2:21, 2:33 2:21, 2:33
POLBLT 1:49 1:45, 1:46
POLLER 2:81 2:81, 2:81
POLRLD 1:9, 1:101 1:89, 1:101
POLTIM 2:78 2:82
POSITI - 2:7, 2:7
POST 2:19
PREFIX 2:80 2:98, 2:104, 2:105, 2:107, 2:110, 2:110
PROAMP 1:68 1:78
PROC - 2:19
PROCBT 1:19 1:42, 1:45, 1:49, 1:49, 1:52, 1:58, 1:63, 1:63, 1:75, 1:78,
1:92, 2:58, 2:58, 2:60, 2:70, 2:96, 2:107, 2:119, 2:126, 2:128
PROCD 1:22 1:78, 1:79, 1:82, 1:82, 1:84, 1:84
PROCES 2:15
PROCEX 1:70 1:79, 1:79, 1:92, 2:61, 2:115
PROCNO 1:19 1:50, 1:52, 1:79, 1:84, 1:84, 1:92, 1:95, 2:62
PROHLT 1:68 1:78
PROHNG 1:68 2:59
PROIOR 1:70 1:78, 2:61
PROKIL 1:68 1:82, 2:61
PRRATE 1:16 1:90, 2:60
PRTIME 1:72 1:90, 2:60, 2:61
PSBACT 2:76 2:87, 2:89
PSBBRK 2:76 2:87
PSBCTL 2:76 2:87, 2:87, 2:89, 2:89
PSBDAT 2:76 2:88, 2:89
PSBECH 2:76
PSBFRE 2:76 2:87, 2:88, 2:89
PSBINT 2:76
PSBONE 2:76 2:86
PSBOUT 2:76 2:87, 2:89, 2:89
PSBOVR 2:76
PSBSTA 2:76 2:87, 2:87, 2:89
PSBTAB 2:86 2:85, 2:86
PSBTWO 2:76 2:86
PSDATA 1:98 1:101, 1:111
PSE 1:1 1:15, 1:29, 1:41, 1:44, 1:96, 2:34

PSETUL 1:98 1:106,1:106,1:111,1:111
 PSEUP 1:98 1:98,1:106,1:111,1:112
 PTPRDR 1:18 1:34
 PTR 1:1,1:18 1:18,1:29,1:34,2:13,2:13,2:13,2:26
 PTRAD 1:34 1:34
 PTRB 1:29 1:34
 PTRBC 1:29
 PTRBL 1:29 1:29,1:34,1:34
 PTRCS 1:29
 PTRDN 1:29
 PTRFB 1:29
 PTRFLG 1:29
 PTRIP 1:29 1:34,1:34
 PTROP 1:29 1:34
 PTRSR 1:29
 PTRST 1:29
 PUT - 2:7
 Q1 1:30 1:30
 Q50 1:30 1:51,1:53
 Q70 1:30 1:53
 QQHAD 1:19 1:32,2:62
 QQHCT 1:19 1:32,2:62,2:62
 QQHPC 1:19 1:32,2:62
 QSUBR 1:32 1:31,1:31,1:32,1:32
 QUIT - 1:34,1:39,1:39,1:41,1:42,1:42,1:42,1:44,1:47,1:47,
 1:47,1:47,1:49,1:50,1:50,1:50,1:50,1:51,1:52,1:55,
 1:56,1:57,1:57,1:58,1:59,1:60,1:60,1:63,1:63,1:64,
 1:64,1:65,1:65,1:66,1:74,1:74,1:76,1:77,1:77,1:80,
 1:81,1:81,1:81,1:82,1:82,1:84,1:84,1:88,1:102,1:103,
 2:52,2:52,2:54,2:54,2:55,2:55,2:56,2:85
 QUITAD 1:19 1:30,1:30,1:32,2:52
 QUITFL 1:22 1:30,1:31,1:32
 QUITPC 1:19 1:30,1:30,1:30,1:31,1:31,1:31
 QUITRT 1:19 1:30,1:31,2:62,2:62
 QUITST 1:19 1:30,1:31
 QUITTM 1:19 1:30,1:30,1:31,1:31
 QUITV 1:19 1:30,1:84
 QUITXT 2:114 2:97,2:113
 QUTPAT - 1:85,1:88,1:88,1:88,1:90,1:90,1:90,1:90,1:93,1:93,
 1:93,1:94,1:94,2:85,2:89,2:126,2:126,2:126,2:127,2:128,
 2:128
 QX 1:19 1:30,1:30,1:31,1:31,1:31,1:31,1:31
 R - 1:102
 RO - 1:42,1:50,1:55,1:56,1:85,1:89,1:94,1:95,1:101,1:107,
 1:108,1:109,1:110,1:112,1:117,2:84,2:128
 R1 - 1:9,1:30,1:30,1:30,1:30,1:30,1:30,1:30,1:31,1:32,1:32,
 1:32,1:32,1:33,1:33,1:33,1:33,1:33,1:34,1:34,1:34,
 1:34,1:35,1:35,1:35,1:35,1:36,1:36,1:36,1:36,1:36,
 1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,
 1:37,1:38,1:38,1:38,1:38,1:38,1:38,1:39,1:39,1:39,
 1:39,1:39,1:40,1:40,1:40,1:40,1:40,1:40,1:40,1:40,
 1:40,1:41,1:41,1:43,1:43,1:43,1:44,1:45,1:45,1:45,
 1:46,1:46,1:46,1:46,1:47,1:47,1:47,1:47,1:47,1:48,
 1:49,1:49,1:50,1:50,1:50,1:50,1:50,1:50,1:50,1:50,
 1:50,1:50,1:51,1:51,1:51,1:51,1:51,1:51,1:51,1:51,
 1:51,1:51,1:52,1:52,1:52,1:52,1:52,1:52,1:52,1:52,
 1:52,1:52,1:53,1:54,1:54,1:54,1:55,1:55,1:55,1:55,
 1:55,1:55,1:56,1:56,1:56,1:56,1:57,1:58,1:58

2:136, 2:136, 2:136
1:9, 1:30, 1:30, 1:31, 1:31, 1:31, 1:31, 1:31, 1:31, 1:31, 1:32, 1:32,
1:32, 1:32, 1:32, 1:33, 1:33, 1:33, 1:34, 1:35, 1:36, 1:36,
1:36, 1:36, 1:36, 1:36, 1:36, 1:37, 1:37, 1:37, 1:37, 1:37, 1:37,
1:37, 1:38, 1:38, 1:38, 1:39, 1:39, 1:39, 1:39, 1:40, 1:40, 1:41,
1:41, 1:43, 1:47, 1:47, 1:50, 1:51, 1:51, 1:51, 1:51, 1:51,
1:51, 1:52, 1:52, 1:52, 1:52, 1:52, 1:52, 1:52, 1:54, 1:54,

R3

1:88,1:88,1:89,1:89,1:89,1:89,1:89,1:90,1:90,
1:90,1:90,1:90,1:90,1:90,1:90,1:90,1:90,1:90,
1:91,1:91,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,
1:92,1:92,1:92,1:93,1:93,1:93,1:93,1:93,1:94,1:94,
1:95,1:95,1:95,1:100,1:100,1:101,1:101,1:103,1:103,
1:103,1:104,1:104,1:104,1:105,1:106,1:106,1:106,
1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:107,
1:107,1:107,1:107,1:107,1:107,1:107,1:107,1:107,1:108,
1:108,1:110,1:110,1:110,1:111,1:111,1:111,1:111,1:111,
1:111,1:112,1:112,1:112,1:112,1:112,1:112,1:112,1:112,
1:112,1:112,1:112,1:114,1:114,1:114,1:115,1:115,1:116,
1:116,1:116,2:15,2:15,2:18,2:18,2:18,2:38,2:38,
2:38,2:39,2:40,2:40,2:40,2:40,2:41,2:43,2:56,
2:56,2:58,2:58,2:58,2:60,2:60,2:60,2:60,2:61,2:62,
2:62,2:68,2:68,2:98,2:98,2:104,2:104,2:104,2:104,2:106,
2:107,2:107,2:110,2:110,2:110,2:110,2:110,2:110,
2:110,2:110,2:110,2:115,2:115,2:115,2:115,2:126,2:126,
2:126,2:127,2:129,2:129,2:129,2:130,2:131,2:131,2:131,
2:131,2:131,2:131,2:131,2:131,2:134,2:134,2:134,
2:134,2:135

R4 - 1:13,1:13,1:13,1:30,1:30,1:31,1:31,1:31,1:31,
1:31,1:31,1:32,1:33,1:33,1:33,1:34,1:34,1:34,1:34,
1:35,1:35,1:35,1:35,1:36,1:36,1:36,1:36,1:36,1:36,
1:37,1:38,1:38,1:38,1:38,1:38,1:38,1:38,1:38,1:38,
1:38,1:38,1:39,1:39,1:39,1:39,1:39,1:39,1:39,1:39,
1:39,1:39,1:39,1:39,1:39,1:40,1:41,1:41,1:41,1:41,
1:42,1:44,1:44,1:44,1:44,1:47,1:47,1:47,1:47,1:47,
1:47,1:48,1:48,1:48,1:48,1:48,1:48,1:49,1:49,
1:49,1:49,1:49,1:49,1:49,1:49,1:49,1:49,1:50,1:51,
1:53,1:62,1:62,1:62,1:62,1:62,1:63,1:63,1:63,1:63,
1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,
1:63,1:63,1:63,1:63,1:63,1:64,1:64,1:64,1:64,1:64,
1:64,1:64,1:64,1:65,1:65,1:65,1:66,1:66,1:66,1:66,
1:73,1:74,1:74,1:75,1:78,1:78,1:78,1:78,1:78,1:78,
1:78,1:78,1:78,1:78,1:78,1:79,1:79,1:79,1:79,1:79,
1:80,1:80,1:80,1:80,1:81,1:81,1:81,1:81,1:82,1:82,
1:82,1:82,1:82,1:82,1:82,1:82,1:84,1:84,1:84,1:84,
1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,
1:84,1:85,1:85,1:85,1:87,1:87,1:87,1:88,1:88,1:88,
1:90,1:93,1:94,1:95,1:106,1:106,1:106,1:106,
1:107,1:107,1:108,1:108,1:109,1:109,1:109,1:109,
1:109,1:109,1:110,1:112,1:112,1:114,1:114,1:114,2:38,
2:38,2:39,2:40,2:40,2:40,2:40,2:40,2:41,2:43,2:45,
2:52,2:52,2:55,2:56,2:56,2:56,2:58,2:58,2:58,2:58,
2:58,2:58,2:58,2:58,2:58,2:61,2:61,2:61,2:61,2:61,
2:61,2:61,2:61,2:63,2:63,2:70,2:85,2:85,2:85,2:87,
2:87,2:87,2:87,2:87,2:88,2:89,2:89,2:89,2:89,2:89,
2:98,2:107,2:110,2:110,2:112,2:112,2:112,2:121,2:121,
2:121,2:122,2:125,2:126,2:126,2:127,2:127,2:127,2:127,
2:127,2:127,2:128,2:128,2:128,2:128,2:128,2:128,2:128,
2:128,2:128

R5 - 1:30,1:30,1:30,1:30,1:30,1:30,1:30,1:31,1:31,
1:31,1:31,1:31,1:39,1:39,1:39,1:39,1:39,1:43,
1:43,1:43,1:43,1:45,1:47,1:47,1:47,1:47,1:48,1:48,
1:48,1:48,1:49,1:49,1:49,1:49,1:49,1:49,1:55,
1:55,1:55,1:56,1:57,1:57,1:57,1:57,1:57,1:57,1:57,
1:57,1:58,1:58,1:58,1:58,1:64,1:64,1:64,1:64,1:65,
1:65,1:65,1:65,1:65,1:66,1:66,1:66,1:66,1:66,

1:105, 1:105, 1:106, 1:106, 1:107, 1:107, 1:107, 1:107,
1:107, 1:107, 1:107, 1:107, 1:107, 1:108, 1:108, 1:108,
1:108, 1:108, 1:108, 1:108, 1:108, 1:108, 1:109, 1:109,
1:109, 1:109, 1:110, 1:110, 1:110, 1:110, 1:110, 1:110,
1:110, 1:110, 1:110, 1:110, 1:111, 1:111, 1:111, 1:111,
1:112, 1:112, 1:112, 1:112, 1:112, 1:112, 1:112, 1:112,
1:112, 1:112, 1:114, 1:114, 1:114, 1:114, 1:114, 1:114,
1:114, 1:114, 1:114, 1:114, 1:114, 1:114, 1:114, 1:114,
1:115, 1:115, 1:115, 1:115, 1:115, 1:115, 1:116, 1:116,
1:116, 1:116, 1:116, 1:116, 1:116, 1:116, 1:116, 1:116,
1:116, 1:116, 1:116, 1:116, 1:116, 1:116, 1:117, 1:117,
1:117, 1:117, 1:117, 1:117, 1:117, 2:16, 2:21, 2:28,
2:28, 2:33, 2:39, 2:39, 2:39, 2:41, 2:41, 2:41, 2:41,
2:41, 2:42, 2:42, 2:42, 2:42, 2:42, 2:42, 2:43, 2:43,
2:43, 2:43, 2:43, 2:43, 2:44, 2:44, 2:44, 2:44, 2:45, 2:45,
2:45, 2:47, 2:47, 2:49, 2:49, 2:49, 2:50, 2:50, 2:50, 2:50,
2:50, 2:50, 2:51, 2:51, 2:51, 2:52, 2:52, 2:52, 2:53,
2:53, 2:54, 2:54, 2:55, 2:55, 2:55, 2:55, 2:56, 2:56, 2:58,
2:58, 2:58, 2:58, 2:58, 2:58, 2:58, 2:59, 2:60, 2:64, 2:64,
2:64, 2:64, 2:64, 2:64, 2:65, 2:65, 2:66, 2:66, 2:66, 2:66,
2:66, 2:66, 2:66, 2:68, 2:68, 2:68, 2:68, 2:68, 2:68, 2:68,
2:68, 2:68, 2:68, 2:68, 2:81, 2:81, 2:82, 2:82, 2:82,
2:82, 2:84, 2:84, 2:84, 2:84, 2:85, 2:85, 2:85, 2:87, 2:87,
2:87, 2:88, 2:89, 2:89, 2:89, 2:89, 2:89, 2:89, 2:90, 2:90,
2:90, 2:90, 2:90, 2:90, 2:90, 2:91, 2:91, 2:91, 2:91, 2:91,
2:91, 2:91, 2:91, 2:91, 2:93, 2:93, 2:93, 2:93, 2:93, 2:93,
2:93, 2:94, 2:94, 2:94, 2:94, 2:96, 2:96, 2:98, 2:98, 2:98,
2:98, 2:98, 2:98, 2:104, 2:105, 2:105, 2:105, 2:105, 2:105,
2:105, 2:105, 2:105, 2:106, 2:106, 2:106, 2:106, 2:110, 2:110,
2:112, 2:112, 2:112, 2:113, 2:115, 2:115, 2:115, 2:115, 2:115,
2:115, 2:115, 2:115, 2:118, 2:120, 2:120, 2:129, 2:129, 2:129,
2:129, 2:129, 2:129, 2:131, 2:133, 2:133, 2:133, 2:134, 2:134,
2:134, 2:134, 2:136, 2:136
RBFGET 2:90 2:82, 2:87, 2:91
RBFLEN 1:96 1:98, 1:98, 1:105
RBFLOK 2:78 2:85, 2:85, 2:90, 2:90, 2:91, 2:91, 2:93, 2:93, 2:93
RBFPUT 2:90 2:82, 2:88, 2:88, 2:90
RBUFE 2:78 2:90, 2:90, 2:91, 2:93, 2:93
RBUFF 2:78
RBUFLE 2:78 2:78, 2:78
RBUFS 2:78 2:85, 2:90, 2:91, 2:91, 2:93, 2:93
RC - 1:23
RCKCON 1:71 1:27, 1:92, 1:92, 1:92, 1:114, 2:61
RCKTAB 1:68 1:88
RCLOCK 1:36 1:35, 1:36, 2:62
RCSTAC - 1:26
RE - 2:7
REL - 1:13, 1:14
RELCOD 1:14 1:11, 1:12, 1:68, 1:73, 1:101, 2:35, 2:35, 2:66, 2:70
RELINI 1:7, 1:7 1:11, 2:35
RELLOD 1:14 1:13, 1:69, 2:35
RELTAB 1:7, 1:7 1:11, 2:35
RELTYP - 1:118, 1:118, 2:35, 2:45, 2:51
RELVAR 1:14 1:12, 1:70, 1:98
RELVST 1:14 1:13, 1:14, 1:14, 2:35
REMemb - 2:7, 2:7, 2:7, 2:7
REQUES - 2:7
RFAIL 1:44 1:24, 1:24

RK - 1:23
RKELIM 1:68 1:68
RKEPAS 1:68 1:60, 1:88
RKERCK 1:68 1:12, 1:60
RKEREN 1:12 1:68
RKERP 1:69
RKPATC 1:69
RKPLEN 1:16 1:69, 1:69
RKSTAC - 1:26
RLDDEV 1:98 1:102, 1:102, 1:116
RLDINB 1:98 1:103, 1:103, 1:104, 1:104, 1:104, 1:104, 1:105, 1:105, 1:105
RLDINS 1:9, 1:9 1:95
RLDOTB 1:98 1:103, 1:103, 1:103, 1:103, 1:103, 1:103, 1:103
RLDSUB 1:102 1:101, 1:105
RLDTYP 1:97 1:101, 1:104
RSUCCE 1:44 1:44
RTCADD 1:17 1:3, 1:64, 1:74, 1:116
RTCPDS 1:17
RTCSWS 1:17 1:116
RTCTEM 1:17
RTRYAD 1:19 1:31, 2:62
RTRYPC 1:19 1:31, 2:62
RUBOUT 2:105 2:98, 2:99, 2:103, 2:105, 2:108
RUBTXT 2:106 2:105
S - 2:7
SAROO 2:60 2:35
SARPCN 2:66 2:62, 2:67
SARPOL 2:64 2:62, 2:65
SARWDG 2:70 2:62, 2:70
SAVEPA - 2:7, 2:7, 2:7, 2:9, 2:16, 2:28
SBAD 1:62 1:51, 1:55, 1:59, 1:62, 1:62, 1:118, 2:58, 2:59
SBDOO 1:73 1:26
SBDCLR 1:77 1:75, 1:77
SBDQCH 1:76 1:73, 1:76
SBDTIM 1:77 1:73, 1:75, 1:77
SCDOO 1:78 1:26
SCDBBC 1:82 1:80, 1:83
SCDBUS 1:22 1:79, 1:80, 1:80, 1:81, 1:81
SCDSET 1:84 1:82, 1:83, 1:84
SCDTAB 1:82 1:82
SCDTST 1:80 1:79, 1:81
SCLEAR 1:63 1:62, 1:63, 2:58, 2:66
SCLROK 1:62 1:55, 1:61, 1:62, 1:74, 1:79, 2:36, 2:39
SDBBLK 2:26 2:26
SEGCON 1:28 1:27, 1:78, 2:61
SEGFIX 1:28 1:27
SEGINC 1:72 1:72, 1:102
SEGMSK 1:72 1:102
SENDST 1:115 1:110, 1:111
SEQH - 1:97
SETBLT 2:115 2:109, 2:110, 2:110, 2:116
SETDEP 2:109 2:107, 2:107, 2:109
SETUP 1:51
SFIXIT 1:63 1:62, 1:63, 1:79
SFXBAD 1:62 1:55, 1:61, 1:62, 1:74, 1:75, 1:79, 1:79, 2:36, 2:38, 2:64
SIDOO 2:56 2:35
SIDFLG 1:29 2:58
SIGN 1:17 1:47, 1:71, 1:98, 2:93

SJ2 1:45
SJ6 1:45 2:128
SJIF 1:35 1:13
SLASH 2:110 2:101,2:106,2:111
SLCOO 2:36 2:35
SLEEP 2:20,2:32
SLFLK - 1:28,1:39,1:58,1:58,1:58,2:39,2:39,2:127,2:127,2:127
SLF PTR 1:28 1:34,1:58,1:58,1:114,2:126,2:126,2:126,2:131
SLKOO 1:52
SLP - 2:20,2:20,2:32,2:32
SLPENT 2:20,2:32
SLSTAC 1:23 1:45,1:89,2:64
SLSTKL 1:14 1:23
SMDOO 1:54 1:26
SMDBLK 1:28 1:57,1:57,1:57,1:58,1:58,1:58
SMDBUC 1:28 1:56
SMDCON 1:27 1:27,1:57,1:57
SMDFLG 1:22 1:54,1:55,1:59
SMDTIM 1:22 1:54,1:55
SMDTS2 1:57 1:56,1:59
SMDTST 1:56 1:54,1:56
SMMOO 2:38 2:35
SMMBAS 1:69 1:69,2:43
SMMCHE 2:45 2:43,2:47,2:48
SMMCOP 2:52 2:49,2:50,2:50,2:52
SMMFIX 2:49 2:38
SMMFRE 1:22 2:42,2:47,2:49
SMMFTY 1:22 2:42,2:47,2:47,2:47
SMMINS 1:69 2:43
SMMOK 1:22 2:42,2:43,2:45,2:49,2:50
SMMQCH 2:54 2:45,2:54
SMMQFX 2:55 2:54,2:54,2:55
SMMSCA 2:47 2:43,2:43,2:48
SMMSEA 2:41 2:38
SMMSSMA 2:52 2:39,2:39,2:53
SMMSPA 1:22 2:42,2:43,2:43,2:46,2:49,2:50
SNAP 1:21 1:39,1:51
SNAPBG 1:19 1:19,1:39
SNAPIL 1:39 1:37,1:40
SNAPLN 1:19 1:39
SNAPLO 1:38 1:36,1:36,1:40
SOKAY 1:63 1:53,1:62,1:63,1:118,2:59,2:62
SOMETH 2:80 2:105,2:105,2:107
SP - 1:33,1:35,1:37,1:45,1:45,1:45,1:49,1:89,2:64,2:64,
2:65,2:84,2:84,2:84,2:84,2:85,2:94,2:94,2:94,
2:94
SPACE 2:104 2:7,2:100,2:104
SRCOO 1:118 1:26
SRCH - 1:97
SRKOO 1:60 1:26
SRKKER 1:22 1:60,1:60,1:60,1:60,1:60,1:61
SRKREL 1:22 1:60,1:61,1:61
SS2BIT 2:79
SS4BIT 2:79
STACK 1:23,1:23 1:23,1:23,1:23,1:23,1:23,1:23,1:23,1:23,
1:23,1:23,1:23,2:76,2:77
STAGE. - 1:1
STAGEK - 1:15

STAGEP 2:128
 STARTI 1:96 1:100
 STARTO 1:96 1:103
 STARTU 2:20,2:32
 STATD 1:96
 STATIM 1:96 1:103,1:104
 STATOM 1:96 1:103,1:103
 STETAB 2:96 2:96,2:97
 STGCON 1:28 1:27
 STGCYC 1:16 1:16,1:16,1:16,1:16,1:49
 STGFIX 1:28 1:27
 STGPAS 1:16 1:60,1:68,1:88
 STGRAT 1:16 1:16,1:36,1:45,1:46,1:64
 STGTIC 1:16 1:16
 STGTIM 1:70 1:73,1:77
 STIM2 1:19 1:47,2:70
 STIME 1:19 1:47,1:48,1:48,1:49,1:49,1:54,1:55,1:73,1:77,1:87,
 1:89,1:90,1:101,1:112,1:114,1:115,2:60,2:68,2:70,2:82,
 2:115
 STKPAS 2:12 2:62,2:126,2:126
 STRIP 2:27 2:26
 STVEC - 2:22,2:22,2:22,2:22,2:22,2:22
 SUCCEE - 1:54,1:60
 SUM 2:80 2:98,2:104,2:105,2:107
 SVTIME 1:22 1:48,1:48,1:48
 SW 1:7 1:7,1:7
 SYSFCB 2:13 2:13
 SYSUQ 1:32 1:52
 SYTIME 1:72 1:47
 SYTIME 1:28 1:47,1:48
 T - 2:6,2:7
 T2FPOK 2:78 2:88
 TAB - 1:5
 TEMP - 2:7,2:7,2:7,2:7,2:9,2:9,2:16,2:16,2:28,2:28
 TEMP1 1:21 1:87,1:89,2:61,2:61
 TEMP2 1:21 1:85,1:86,1:86,1:87,1:90,1:91,1:92,1:92,1:93
 TEMP3 1:21 1:85,1:88,1:88,1:89,1:91,1:93,1:95
 TEMP4 1:21 1:93,1:93,1:94,1:95,2:131,2:131
 TESTSW 1:7
 TEXTOU 2:121 2:96,2:113,2:121
 TF3 - 1:85
 TG - 1:91,1:91
 THE - 2:7,2:7
 THIS - 2:6,2:7,2:7
 TIM - 1:7,1:7,1:7,1:7
 TL - 1:91
 TLIMIT 1:28 1:68,1:118,2:37,2:50
 TNE - 1:92
 TNF1 - 1:86,1:86
 TNO - 1:87
 TNZ - 1:92,1:92,1:95
 TO - 2:7,2:7,2:7
 TOPNTR 1:28 2:37
 TOTAL - 1:13,1:14
 TOTSTB 1:16 1:45,1:48,1:63,1:112
 TOTSTS 1:16 1:16,1:16,1:22,1:27,1:47
 TOTXT 2:114 2:112
 TR - 1:86,1:86,1:86,1:87,1:88,1:91,1:91,1:92,1:94,2:128

TRAP - 1:9,1:31,1:33,1:35,1:35,1:35,1:35,1:35,1:38,1:47,1:48,
1:50,1:52,1:53,1:55,1:66,1:66,1:75,1:76,1:80,1:81,
1:82,1:84,1:87,1:91,1:92,1:93,1:95,2:36,2:40,2:44,
2:51,2:55,2:62,2:63,2:70,2:85,2:125
TRAPCN 1:40 1:37,1:40
TRAPV - 1:37,1:37
TRUE - 2:17,2:17,2:17,2:29,2:29,2:29
TRYRLD 1:9,1:116 1:118,2:36,2:50
TT - 2:77
TTPOKE 2:78 2:89
TTSLEE 2:84 2:84,2:85,2:87,2:89
TTSTAC - 2:85
TTSYSS 2:77 2:84,2:84
TTY 2:87
TTYIBF 2:78 2:82,2:88,2:93,2:93,2:93
TTYINI 2:85 2:84,2:85,2:85,2:88,2:89
TTYLOK - 2:78,2:84,2:84,2:84
TTYOBF 2:78 2:82,2:85,2:85,2:85,2:87,2:88,2:93,2:93,2:93
TTYOUT 2:89 2:87,2:89
TYPOL 2:84 2:81
TTYSP 2:77 2:84,2:84
TTYSTU 2:87 2:87,2:88,2:89
TYP - 1:6,2:4,2:7,2:7
TYPE - 2:7,2:7
TYPE4K 1:28 1:40,2:37,2:45,2:45,2:51,2:51,2:53
TYPH 1:97 1:104
TZ - 1:92,1:92
UCTRL 1:19 1:32,1:39,1:39,1:39
UILLOP 1:19 1:39
UJIFFY 1:19 1:36
UMAP 1:19 1:39
UQUIT 1:19 1:31,1:32,1:52
UQUITD 1:19 1:32
UQUITP 1:19
UQUITS 1:19
USEBUS 1:70 1:74,1:74,1:80,1:81,1:93,1:102
USEIO 1:72 1:72,2:58,2:58,2:58,2:131
USEIOL 1:72 1:102,2:131
UTIME 1:22 1:45,1:46
UWST 1:19 1:51
V2 2:34 2:34,2:124
V2ST 2:34 2:34
VALUE - 2:4,2:4
VAR - 1:13,1:14
VARS 1:14 1:13,1:14,1:28,1:29,2:9,2:70,2:78,2:78,2:80,2:124
VARSPA 1:5 1:13,2:34
VARTYP - 1:40
VDH - 1:102
VECTOR - 2:7
VLST 1:5 1:5,1:6
WAIT 2:22 2:22
WAKEUP 2:22 2:22
WATCH1 2:124 2:128
WATCH2 2:124 2:128
WATCHS 2:124 2:128
WATM1 2:124 2:128
WATM2 2:124 2:128
WDGTIM 2:70 2:70,2:70

| | | |
|---------|------------------------|--|
| WDIS | 1:22 | 1:45, 1:46, 1:46, 1:62, 1:63, 1:63, 1:63, 1:112 |
| WE | - | 2:7, 2:7, 2:7, 2:7 |
| WERE | - | 2:7 |
| WHERE | - | 2:7, 2:7 |
| WHEREV | - | 2:7 |
| WMLOCK | - | 1:28, 1:56, 1:56, 1:56, 1:56 |
| WOPS | 1:50 | 1:8, 1:8, 1:8, 1:8, 1:13, 1:33, 1:34, 1:34 |
| WORDS | 1:17 | 1:6, 1:17, 1:20, 1:23, 1:27, 1:27, 1:30, 1:30, 1:30, 1:31, 1:32, 1:36, 1:36, 1:37, 1:37, 1:37, 1:38, 1:38, 1:40, 1:40, 1:43, 1:43, 1:47, 1:47, 1:47, 1:48, 1:56, 1:63, 1:66, 1:67, 1:74, 1:78, 1:79, 1:79, 1:79, 1:79, 1:79, 1:79, 1:80, 1:81, 1:82, 1:84, 1:85, 1:85, 1:88, 1:88, 1:90, 1:94, 1:94, 1:94, 1:94, 1:96, 1:105, 1:107, 1:107, 1:109, 1:112, 1:112, 1:116, 2:38, 2:39, 2:40, 2:43, 2:63, 2:63, 2:63, 2:70, 2:96, 2:106, 2:106, 2:109, 2:115, 2:126 |
| WPCINT | 1:26 | 1:26, 1:63 |
| WS | 1:51 | 1:13, 2:70 |
| WSLEEP | 1:45 | 1:49, 1:54, 1:60, 1:65, 1:66, 1:74, 1:76, 1:78, 1:79, 1:80, 1:118, 2:48, 2:51, 2:54, 2:56, 2:57, 2:60, 2:65 |
| WSLSP | 1:22 | 1:45, 1:49, 1:63 |
| WSPINT | 1:26 | 1:63 |
| WST | 1:51 | 1:13, 1:32, 1:35, 1:35, 1:36, 1:38, 1:39, 1:50, 1:50, 1:50, 1:50, 1:50, 1:53, 1:74, 1:75, 1:84, 1:119, 2:52, 2:55, 2:127 |
| WSTAGE | 1:22 | 1:45, 1:46, 1:46, 1:47, 1:62, 1:63, 1:63, 1:63 |
| WSTCOM | 1:51 | 1:47 |
| WSTINI | 1:50 | 1:86, 1:94 |
| WSUB | 1:51 | 1:51, 1:51 |
| X | - | 2:17, 2:17, 2:17, 2:17, 2:17, 2:17, 2:29, 2:29, 2:29, 2:29, 2:29, 2:29 |
| XSIOIN | 1:100 | 1:99, 1:100, 1:105 |
| YES | - | 2:6 |
| Z | - | 2:129 |
| \$ | 2:17, 2:29 | 1:5, 2:6, 2:17, 2:17, 2:17, 2:17, 2:21, 2:21, 2:27, 2:29, 2:29, 2:29, 2:29, 2:30, 2:33, 2:33 |
| \$%PLST | - | 2:27 |
| \$ADDPA | 1:5 | 1:5, 1:5, 1:5, 1:5 |
| \$APPLY | - | 1:6, 1:6, 1:6, 1:6, 1:6, 1:12, 2:6, 2:6, 2:7 |
| \$ARGTS | 2:17, 2:29 | 2:17, 2:29 |
| \$CLKFL | 2:27, 2:27 | |
| \$CLOCK | 2:27 | 2:30, 2:30 |
| \$DOPAG | - | 2:6, 2:7 |
| \$DOPAT | 1:11 | 1:11, 1:12, 2:81 |
| \$DOPPA | - | 2:4, 2:4, 2:7, 2:7, 2:16, 2:28 |
| \$ENTER | - | 1:5, 2:4, 2:4, 2:4, 2:4, 2:4, 2:6, 2:6, 2:7, 2:7, 2:8, 2:8, 2:27 |
| \$EXIT | 2:19, 2:19, 2:31, 2:31 | 2:19, 2:27 |
| \$FAST | 2:30 | |
| \$FCBAR | 2:15, 2:15 | 2:15, 2:15, 2:15, 2:16, 2:16 |
| \$FINCH | 1:12 | |
| \$FINKE | 1:12 | 1:119 |
| \$INST | 1:11 | 1:119, 2:138 |
| \$FIXCH | 1:12 | 1:12 |
| \$IFDF1 | - | 2:7 |
| \$INIT | 2:4 | 2:9 |
| \$INITR | 1:5 | |
| \$IR1 | 2:7 | 2:6, 2:6 |

\$IROUT 2:6
\$ITABL 2:7 2:136
\$LCCHK 2:16 2:15
\$LIST - 1:5, 1:5, 1:5, 1:5, 2:4, 2:4, 2:4, 2:4, 2:6, 2:6, 2:7
\$LSCHK 2:28 2:27
\$MAPCH 2:4 2:4
\$NO 2:17, 2:29
\$OPT 2:17, 2:29 2:15, 2:27
\$PGKEY 1:5, 1:5 1:5, 1:5
\$PHEAD 2:16 2:16, 2:16
\$PID 2:27 2:27, 2:27
\$PID1 2:6 2:6
\$PINIT 2:6 2:7, 2:8
\$RATE 2:15 2:16
\$RINIT 2:6 2:7, 2:8
\$SHEAD 2:30 2:28, 2:28
\$SLOW 2:30
\$STACK 2:15, 2:15, 2:15 2:15, 2:15, 2:15, 2:16
\$TIMER 1:5
\$XINIT 2:4 2:4, 2:4
% - 1:4, 2:15, 2:23
%%PHYS - 2:7, 2:7
%3SPAC 2:112 2:96, 2:112, 2:113
%CCHN 2:27 2:30
%CNT 1:6, 1:6, 1:6, 1:6 1:6, 1:6, 1:6, 1:6
%CODE - 1:6, 1:6, 1:6
%CR 1:18
%DOT 2:16, 2:28 2:16, 2:28
%DR 1:18 1:34
%FLG 1:5, 2:7, 2:17, 2:17, 2:17, 2:29, 2:29, 2:29 1:5, 2:7, 2:7, 2:7, 2:17, 2:29
%ICHAI 2:7 2:7, 2:7
%IKEY - 2:4, 2:4
%IKEY3 2:4
%IKEY4 2:5
%IKEY5 2:5
%ILST 2:4, 2:6 2:4, 2:4, 2:4, 2:4, 2:4, 2:4, 2:6
%IMAP - 2:4, 2:4
%IMAP3 2:4
%IMAP4 2:4
%IMAP5 2:4
%INI 1:5, 1:5 1:5, 1:5
%IPAGE - 2:7, 2:7
%KEY - 2:4, 2:4, 2:4
%LIM 1:5 1:5, 1:5
%M 2:4 2:4
%MAP - 2:16, 2:16, 2:28, 2:28
%MAPO - 1:12, 1:48, 1:49, 1:56, 1:56, 1:91, 1:91, 1:92, 1:92, 2:16,
2:28, 2:51, 2:64, 2:64, 2:65, 2:81, 2:126, 2:131, 2:131
%MAP1 - 1:34, 1:34, 1:39, 1:47, 1:49, 1:58, 1:65, 1:66, 1:67, 1:88,
1:88, 2:4, 2:44, 2:44, 2:49, 2:50, 2:51, 2:54, 2:63, 2:81, 2:126
%MAP2 - 1:47, 1:47, 1:87, 2:4, 2:49, 2:50, 2:50, 2:126, 2:128, 2:128,
2:128, 2:129
%MAP3 - 1:48, 1:49, 1:51, 1:56, 1:91, 2:4, 2:39, 2:50, 2:81, 2:126
%NCODE 1:6 1:11, 1:11
%NDOT 2:7, 2:7 2:7, 2:7, 2:7
%NOVAR 1:6 1:11, 1:51, 1:72
%NSPAR 1:6 1:6, 1:11
%NTYP 2:7 2:7, 2:7

```
%NTYPE -      2:7,2:7
%NVARS  1:6      1:6,1:11
%OPTV   -        1:6
%PHY    1:5      1:5,1:5
%PLST   2:4,2:6 2:6,2:27
%RADIX  2:15,2:15 1:4,2:15
%RLST   2:4,2:7 2:6,2:6,2:6,2:7,2:7,2:7,2:8,2:8
%SR    1:18     1:34
%SRADI  1:4,2:15 1:4,2:15
%TAB    1:5,1:5 1:5,1:5
%TOTAL -        1:4
%VARS   -        1:6
.%INIT  2:15     2:18
..FOO   1:28     1:28,1:28
.INSER -        1:1,1:9,1:15,1:99,2:1,2:10,2:24,2:34,2:71,2:123
.LALL  -        1:23
.XALL  -        1:23
```

Pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52
IMPLOD.PLR;1 PAGE 1

PAGE 1

```
.title pluribus IMP Loader
;Assemble special paper tape loader code
0001      PTR = 1
;Assemble the old format Packet Core Loader
0000      NewPkc = 0
;Modify Stage, etc. for Platform
0001      PSE = 1
;First get the various files of macros
      .insym MACROS.SYM
;Now get STAGE configuration file
      .INSERT "STAGE.CFG",STAGE.CFG
      .INSRT STAGE.CFG
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 2
STAGE.CFG;1 PAGE 1

;STAGE.CFG
;John Robinson/Eric Roberts 22-Apr-78
;-----

; This file defines a number of macros which allow
; the application program to specify the configuration to
; be assumed by the stage system. The macros and their
; general use are documented in the file <KERNEL>CFG.DOC.

```
;Macros to define common memory and I/O bus configuration
:DEFBUS

;The DEFBUS macro declares the memory and I/O bus
;addresses for the system. The general form is
;      DEFBUS <m1,...>,<i1,...>
;where the constuctions <m1,...> and <i1,...> are the
;list of memory and I/O bus addresses respectively.

.MACRO DEFBUS MLIS,ILIS
.MACRO BSADIT
    .IRP A,<ILIS>
        H'A
        BESCLK= H'A+RTCADD ;Highest numbered clock
    .ENDM
.ENDER

.MACRO BSADMT
    .IRP A,<MLIS>
        H8000
    .ENDM
.ENDER

.MACRO BSMPTB
    .IRP A,<ILIS>
        1
    .ENDM
.ENDER

.MACRO BSMPTM
    .IRP A,<MLIS>
        A
    .ENDM
.ENDER
.ENDER
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 4
STAGE.CFG;1 PAGE 3

```
;Physical page definition macros

;Pages which are to be managed by Stage
;must be declared during the assembly by the
;use of one of the page declaration macros
;defined below. There are four macros which
;correspond to different page classes supported
;by Stage:

; (1) Code pages (spares duplicate this list)
; (2) Required variables pages
; (3) Desired variables pages
; (4) Optional variables pages

;The definition of the different page types may be
;found in the documentation for Stage.
;The names of physical pages should be no more
;than three characters in length.

;The total number of slots to be reserved for
;each type of page must be declared by the
;DEFMEM macro at Stage generation time. The
;DEFMEM macro is keyword driven and accepts
;the following argument forms:

; TOTAL=n           ;Total number of page slots .
; CODE=n            ;Number of code page slots
; VARS=n            ;Number of variable page slots
; OPTV=n            ;Number of optional variable slots

;Define the DEFMEM macro

.MACRO DEFMEM ARG1,ARG2,ARG3,ARG4,ARG5
%SRADIX=%RADIX          ;Save the radix
.RADIX D10               ;Before moving to decimal
.IRP ARG,<ARG1,ARG2,ARG3,ARG4,ARG5>
.IF NB <ARG>             ;For each non-blank arg
%'ARG                  ;Assign the keyword parameter
.ENDC
.ENDM
.RADIX %SRADIX           ;Set the radix back
NMSEG=%TOTAL/ H8         ;Bytes in memory discovery
.ENDM
```

;Define the page definition macros

```
.MACRO CODEPAGE NAME,ARG1,ARG2,ARG3,ARG4
$ADDPAGE CLST,<NAME>
%LIM == -1
%PHY == -1
%INI == -1
%TAB == -1
.IRP ARG,<ARG1,ARG2,ARG3,ARG4>
.IIF NB <ARG>, '$ARG
.ENDM
.IIF NZ 1+%LIM, NAME'LIM == %LIM
.IIF NZ 1+%PHY, NAME'LOD == %PHY
.IIF NZ 1+%INI, NAME'INI == %INI
.IIF NZ 1+%TAB, NAME'TAB == %TAB
.ENDM
```

;Macros for the CODEPAGE arguments (LIMIT and PHYSICAL in PAGE)

```
.MACRO $INITROUTINE ARG
%INI == ARG
.ENDM
```

```
.MACRO $TMRROUTINE ARG
%TAB == ARG
.ENDM
```

```
.MACRO VARSPAGE NAME
$ADDPAGE VLST,<NAME>
.ENDM
```

```
.MACRO DESVPAGE NAME
$ADDPAGE DLST,<NAME>
.ENDM
```

```
.MACRO OPTVPAGE NAME
$ADDPAGE OLST,<NAME>
.ENDM
```

;And a macro to add the pages to a list

```
.MACRO $ADDPAGE LST,NAME
%FLG == .ADRMD =0+PG$'NAME ;Check if defined previously
.IF NZ %FLG - H880 ;Short constant => 'defined
$ENTER LST,NAME ;Enter name on list
NAME'KEY == $PGKEY ;Assemble unique labels
$PGKEY == $PGKEY + 1
PG$'NAME == 1 ;Define the page
.ENC
.ENDM
```

;Define the page lists

```
$LIST CLST ;List of code pages
$LIST VLST ;List of required vars pages
$LIST DLST ;List of desired vars pages
$LIST OLST ;List of optional vars pages
```



```
;Macro to set up the memory tables

.MACRO LMAPTB
  %CNT==0
  $APPLY DEFM1,CLST
  MNCODE=-LMAP-2
  .BLKW %CODE-%CNT
  %NCODEP=-LMAP
  .BLKW %CODE
  %NSPARP=-LMAP
  %CNT==0
  $APPLY DEFM1,VLST
  MNVARS=-LMAP-2
  .BLKW %VARS-%CNT
  %NVARSP=-LMAP
  %CNT==0
  $APPLY DEFM1,DLST
  INDFAR=<<,-LMAP-%NVARSP>+<MNVARS-%NSPARP>/WORDS>+1 .
  $APPLY DEFM1,OLST
  MNOVAR=-LMAP-2
  .BLKW %OPTV-%CNT
  %NOVARP=-LMAP
  .ENDM

.MACRO DEFM1 P
  %CNT==%CNT+1
  .IIF NDF P'LOD, P'LOD=-1
  P'TYP=-LMAP
  MAP'P: .BLKW 1
  .ENDM

.MACRO MMLIMS
  $APPLY MLIMX,CLST           ;Add limits for code pages
  .REPT %CODE-<<,-LIMTAB>/2>
    -1
  .ENDR
  .ENDM

.MACRO MLIMX PG
  <M1/><PG'LIM-2>+2
  .ENDM
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 7
STAGE.CFG;1 PAGE 6

;Now for the specific addresses in IO space

```
.MACRO DEFIO LIS1
  .MACRO IOBTAB
    .IRP A,<LIS1>
      H'A
    .ENDM
  .ENDM
.ENDM
```

;Macro to help set up Rely page (init, timeout)

```
.MACRO DEFRELINI,TIM
  .IF NBINI
    .IF DFINI
      RELINI=INI
    .IFF
      RELINI=0
    .ENDC
  .ENDC
  ;DFINI
  .IF NBTIM
    .IF DFTIM
      RELTAB=TIM
    .IFF
      RELTAB=0
    .ENDC
  .ENDC
  ;NBTIM
.ENDIF
```

;macro to let user specify max strip time(+ 100 microsec)

```
.MACRO MSTRIP ARG
  MAXSTR=D'ARG
.ENDIF
```

;Macro to test assembly switch settings

```
.MACRO TESTSW LIS
  .IRP SW,<LIS>
    .IF NDFSW
      PRINT |SW UNDEFINED, ASSUMED 0
      |
      SW=0
    .ENDC
  .ENDIF
.ENDIF
```

;A macro to let you change default Interrupt handlers

```
.MACRO DEFINT L1,L2,L3,L4
    .IF B L1
        DEFINT WOPS,L2,L3,L4
    .IFF                                ;B L1
        .IF B L2
            DEFINT L1,WOPS,L3,L4
        .IFF                                ;B L2
            .IF B L3
                DEFINT L1,L2,WOPS,L4
            .IFF                                ;B L3
                .IF B L4
                    DEFINT L1,L2,L3,WOPS
                .IFF                                ;B L4
                    .MACRO INTABL
                        L4
                        L3
                        L2
                        L1
                    .ENDM
                .ENDC                                ;B L4
            .ENDC                                ;B L3
        .ENDC                                ;B L2
    .ENDC                                ;B L1
.ENDM
```

;Macro to load interrupt handlers

```
.MACRO DEFLIN LIST
    .MACRO LODINT
        .IRP A,<LIST>
        .IF NB A
            .=%SERVC+< H8*.IRPCN>
            A
        .ENDC
    .ENDM
    .ENDM
.ENDM
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 9
STAGE.CFG;1 PAGE 8

```
;A macro for specifying a reloader program source file

.MACRO DEFRLD ARG1,ARG2
.IF NB <ARG1>
.MACRO RLDINS
    .INSERT "ARG1",<ARG2>
.ENDM
.IFF           ;No reloader
.MACRO RLDINS
    ROUTINE POLRLD
    ENDROUTINE
    ROUTINE TRYRLD,ARG R1,INLINE R2
        TRAP 77,<;LOST REQUIRED CODE page 9>
    ENDROUTINE
.ENDM
.ENDC
.ENDM
```

;Now for the code-generators

;First, a macro to define the macros to define the macros

```
.MACRO DEFMAC OUTER,INNER
  .MACRO OUTER A
    .MACRO INNER
      A
    .ENDM
  .ENDM
.ENDM
```

DEFMAC DEFBEG,LOCBEG ;Code to assemble at head of LCode page

DEFMAC DEFILLOP,PILLOP ;User ILLOPR handler to call

DEFMAC DEFATN,PATTN ;User ATTN handler to call

DEFMAC DEFLOOP,P\$LOOP ;Stage Exit to System

DEFMAC DEFPCNT,P\$PCNT ;Min procs to run system

```
.MACRO DEFENTS E1,E2,E3,E4 ;Entries to non-kernel stages
  .MACRO ENTLIST
    E1,E2,E3,E4
  .ENDM
.ENDM
```

;Set up some default empty macros

```
DEFATN
DEFILLOP
DEFLOOP
DEFPCNT
```

```
.MACRO ENTLIST
.ENDM
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52
STAGE.CFG;1 PAGE 10

PAGE 11

```
;Define a macro to patch Stage system

.MACRO $FINSTAGE
$DOPATCH LMAP,LVARS
LMAPTB
$DOPATCH LIMTAB,RELCODE
MMLIMS
$DOPATCH BSMAPS
BSMPTB
BSMPTM
.if df minproc
$DOPATCH MINPROC
P$PCNT
.endc
$DOPATCH NCODEP
%NCODEP
$DOPATCH NSPARP
%NSPARP
$DOPATCH NVARSP
%NVARSP
$DOPATCH NOVARP
%NOVARP
$DOPATCH NCODEM
MNCODE
$DOPATCH NSPARM
MNCODE+%NCODEP
$DOPATCH NVARSM
MNVARS
$DOPATCH NOPTVM
MNOVAR
$DOPATCH NDVARS
INDVAR
$DOPATCH IOBASE
IOBTAB
$DOPATCH CWPCIN
ENTLIST
$DOPATCH JLOOP
P$LOOP
$DOPATCH MO#PGINIT
RELINI
RELTAB
$DOPATCH ILLOPAT,L.CODE
PILLOP
$DOPATCH ATTNPAT
PATTN
$DOPATCH MMAXSTR
MAXSTR
$DOPATCH L$25.6
P$25.6
$DOPATCH L$1.G
P$1.G
$DOPATCH BSADRS
BSADIT
BSADMT
.ENDM
```

```
.MACRO $DOPATCH DOT,PG
.IF NB <PG>
  PAGE PG
  PAGE DUMMY
.ENDC
.=DOT
.ENDM
```

```
pluribus IMP Loader      PLURIBUS V2.9B  25-Jun-87 11:20:52      PAGE 12
STAGE.CFG;1      PAGE 11

        ;macro for patching non-kernel checksums

        .macro $fixchk name
            .=%map0          ;physical map change
            name'1od
            .=cksum
            ckpass
        .endm

        .macro $finchk
            page Dummy
            $dopatch localc
            ckpass
            $apply $fixchk,c1st
        .endm

        ;Macro to finish up kernel definitions

        .macro $finker

            page LCode

            lkeren: .b1kw 1           ;end of Local Kernel

            page RelVars

            lcomar=.-comar          ;how much Stage vars to QUIT-check

            page Relcode

            rkeren: .b1kw 0           ;end of Rely Kernel

            ;These must be loaded onto by rest of system

            cwpcin: .b1kw comsts      ;common stage entries
            jloop: .b1kw 1             ;system entry

            page Dummy

            .=rkerck                ;init cksum passwords
            ckpass

            .=lkerck
            ckpass

            page Dummy

        .endm

        ;Macro to use at end of a .BIN that wants to keep loading

        .macro contload
            .end  HFE00
        .endm
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 13
STAGE.CFG;1 PAGE 12

;Now, do the default definitions

DEFBUS <0, H4000>,<E000,FO00>

DEFMEM TOTAL=G4, CODE=8, VARS=8, OPTV=8

CODEPAGE REL, LIMIT RELVST, PHYSICAL RELLOD

VARSPAGE VAR

DEFIO <E100,E200,F100,F200>

DEFINT LEVEL1,LEVEL2,WOPS,SJIF

DEFLIN <.....> ;Halt on quits

DEFREL 0,0

MSTRIP 200 ;Allow 20 millisecond strips

DEFRLD ;No reloading

DEFBEG <
JSB R4,WS ;Clean restart and init
JSB R4,WST ;Just enter system
JSB R4,DSTAND ;Start DDT+STAGE only
>

;Give default values to the clock lists

0001 P\$25.6==NIL
0001 P\$1.6==NIL

;end of stage.cfg

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 14
IMPLOD.PLR;1 PAGE 1.1

```
;Define the initial pages to load

0000    rel1od=0

:Page origins

0050    locvst= H50          ;local vars after hardware
02F8    loccst= H300- H8    ;local code, 8 for cksums
0038    lstkln= D56         ;length of local stack (words).
0020    slstk1= D32         ;length of system (STAGE,DDT) portion
0288    locsst=loc cst-<lstkln*2>;local stack area
4000    locend=m0           ;always permit 8K locals
5D50    relvst= H5D50      ;stage vars

:Define the logical pages

Defpage LVars, org locvst, limit locsst
Defpage Lc1Stk, org locsst, limit loccst
Defpage LCode, org loccst, limit m0

Defpage RelCode, map code, limit relvst, physical rel
Defpage RelVars, org relvst, limit m1

Defpage Vars, limit m2, physical var

Defpage Dummy ;for non-code assembly

;Now set up the parameters for assmbling Stage

DEFMEM TOTAL= D64, CODE= H8, VARS= H8, OPTV= D16
;64 pages in stage MD, 8 each code (and spare code),
; required vars, 16 desirable + optional vars in stage MM

;four 16-device segments on 2 buses
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 15
IMPLOD.PLR;1 PAGE 2

```
DEFLIN <level1,.....> ;enable S-A DDT.  
;halt if quit.jiffy  
  
;File to insert for reload code  
  
DEFRLD <IMP.PKCORE>.PKCORE  
  
DEFATN DDTATTN  
  
;Suppress all F traps for Platform  
  
.if df PSE  
  
E000      FTrapv = HEOOO  
  
.endc    ;df PSE  
  
;Now insert STAGE Kernel code  
  
.INSERT "STAGEK.PLR",STAGEK  
.INSRT STAGEK.PLR
```

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 16
 STAGEK.PLR;1 PAGE 1

.comnt |
 Pluribus Stage System

Currently, the stages are:

```
0 LK Local memory Kernel checksum
1 MD common Memory Discovery
2 RK Reliability page Kernel checksum
3 BD common Bus Discovery
4 CD processor Coupler Discovery
5 RC Reliability page Code checksum
6 LC Local memory Code checksum
7 MM common Memory Map management
8 ID I/o devices Discovery
9 AR Application Reliability and initialization
10 The operational system (sic)
```

;some important parameters

```
000A totsts= D10      ;total number of stages, see contab
0004 comsts=4        ;how many stages not in kernel
0500 stgrat= D1280    ;speed (in 100 micsec) to run stage at
0005 stgtic=stgrat/ D256   ;stage clock ticks every fast rtc tick
003C stgcyc=<2*totsts>*stgtic ;how often each stage gets to run
0F00 memrat= D8*nmseg*stgcyc ;how long to wait to fix a comprtr
0078 prrate=2*stgcyc ;time interval to give procs to start
0010 nproc= D16        ;how many procs max *must be power of 2*
0258 inirat= D10*stgcyc ;how many 25.6 ms ticks before reinit
0400 totstb=1_totsts :bit position of last stage
0020 b1tstb=1_5       :bit of stage (RC) that may run BLT subroutine
0ACE stgpas= HACE     :password to identify rely kernel
FEED ckpass= HFEED    :checksum to say recompute checksum .
0040 b1tmax= H40       :max bytes per block transfer strip
0078 b1trat=2*stgcyc ;rate for BLT timeout
000F anypro= HF        :special number for processor set
0020 LKP1len= H20      :Local Kernel Patch area length.
0040 RKP1len= H40      :Rely Kernel Patch area length
0008 cilnum= D8        :Number of Traps to buffer in common
```

;some hardware definitions

page Dummy

0002 words=2 ;how many bytes (i e adresses) per memory word
8000 sign= H8000 ;sign bit

F800 iomask= HF800 ;significant bits to choose i/o bus
0800 businc= H800 ;offset to get from one bus to the next
FEO0 mapmsk= HFE00 ;significant bits to choose 4K common memory

;Common I/O Bus Layout

0000 .=0

;first the PID:

0000 pidsto: .blkw 1 ;address to store PID setting
;reads back highest
0002 pidrcl: .blkw 1 ;read-and-clear highest PID set
0004 .blkw 1 ;write to clear whole PID

;RTC here:

0006 rtcadd: .blkw 1 ;how far into a bus for RTC time
0008 rtcpds: .blkw 1 ;PID settings: slow,,fast
000A rtcsws: .blkw 1 ;IMP NUMBER SWITCH SETTING,
000C RTCTEM: .BLKW 2 ;AND TEMPERATURE SENSORS

;Now for couplers:

0010 couplr: .blkb H70 ;coupler space on a bus
DE79 bbcpas= HDE79 ;password to change coupler state
FFFF bbcres=-1 ;reset the processor bus
0002 bbctr=2 ;forward enable
0004 bbcbak=4 ;backward enable
0080 bbctrn: .blkw 4 ;BBC window, just after last coupler
0088 .blkw 3 ;answers, but not used
008E bbcmpr: .blkw 1 ;BBC map register
0002 bbccod=2 ;key bit in BBC map
FFF8 bbcmask= HFFF8 ;meaningful address bits for BBC map
0006 bbccwmk=<-1?bbcmask>&-words ;bits within window
2100 bbccans= H2100 ;the bits the coupler reports

;Pluribus device standards

FF00 devtyp= HFF00 ;what kind of device this is
0010 devinc= H10 ;how much to get from one device to next

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 18
STAGEK.PLR;1 PAGE 3

```
;processor control register (R15) bits

0001 pchalt=1      :halt a processor in control register (%ctrl)
0002 pcrun=2       :start a processor
0003 pcstep=pchalt}pcrun   :step a processor

;pseudo-control register bits
;(communicates to DDT via left byte of "R15")

0001 locqut=1     :processor "stopped" on a quit
0002 locilo=2     :ditto for -1 or funny illopr
0004 locfad=4     :ditto for FADE illopr

;paper-tape reader parameters

FC30 ptprdr= HFC30 ;typical address
.iif ndf PTR, PTR = 0 ;none if missing def

0000 .=0
0000 %sr: .blkw 1 ;status register
0002 .blkw 2
0006 %cr: .blkw 1 ;control register
0008 %dr: .blkw 1 ;data register

;*** various assigned pid levels

0000 .=0
0000 emtpid: .blkw 1 ;this PID is empty, go to next
; PID table goes here

0002 .blkw D125

00FC nopid: .blkw 1 ;pid when all real PIDs are empty
00FE .blkw 1 ;reserved highest PID
```

; ***< IMP STAGE SYSTEM - LOCAL CODE >***

0006 04B2 lodint :first clobber interrupt handlers
001E 001E
002E 002E

0002 PAGE LVARS
 bitsnk=2 ;local memory bit sink

0050 ;jiffy locals
 clokrt: .blkw 1 ;number of RTC retries (not zeroed)
 ;quit locals

0052 uquit: .blkw 1 ;unexpected quit handler
0054 quityv: .blkw 1 ;which quit vector (2A or 3A)
0056 quitrt: .blkw 1 ;number of successful quit retries
0058 rtryad: .blkw 1 ;last retry referencing here
005A rtrypc: .blkw 1 ;last retry from this pc value
005C qqhct: .blkw 1 ;count of QUITs in QUIT handler
005E qqhad: .blkw 1 ;latest of above referenced here
0060 qqhpc: .blkw 1 ; with this PC

0062 ;quittm though end of qx is a single logical block
 quittm: .blkw 1 ;time of reference
0064 quitad: .blkw 1 ;address referenced
0066 quitst: .blkw 1 ;reg 8
0068 quitpc: .blkw 1 ;reg 0
006A qx: .blkw 7 ;regs 1-7
 ;end of block

 ;illop locals

0078 snapbg: ;beginning of snapshot
 uillop: .blkw 1 ;last F-illopr (simulated R0)
 ix: .blkw 7 ;regs 1-7 (at last illopr or jiffy)
 iret: .blkw 2 ;regs 8, 0 (ditto)
008C uwst: .blkw 1 ;last call to wst (or r4 at startup) (sim. R10)
008E ujiffy: .blkw 1 ;location of last "program in a loop" (sim. R11)
0090 uquitd: .blkw 1 ;got unex quit trying to look here (sim. R12)
0092 uquits: .blkw 1 ;status at unex quit (sim. R13)
0094 uquitp: .blkw 1 ;address of place that did reference (sim. R14)
0096 uctrl: .blkw 1 ;my own cntl reg (R15) simulated
0098 umap: .blkw 4 ;saved maps at illopr/quit
00A0 stime: .blkw 1 ;local copy of sytime (to avoid map changes)
00A2 stim2: .blkw 1 ;high order time, 27.96 min/tick, 51.50 tick/day
00A4 pidp: .blkw 1 ;last pid dispatch

00A6 myproc: .blkw 1 ;processor number from coupler
00A8 procbt: .blkw 1 ;processor bit

0032 snapln=-snapbg ;how much to save in snapshot

00AA prochno: .blkw 1 ;processor number
00AC pid: .blkw 1 ;PID for software to poke in

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 20
STAGEK.PLR;1 PAGE 5

OOAE mapcom: .blkw 1 ;communication page
;local map table; this is a copy of cmap
;maprel must be beginning of lmap and cmap

lmap: lmaptb
;local table of illops

0007 nsuer=7 ;how many registers in a SUE
0008 ibuf1=nsuer+1 ;remember trap no. & regs
0004 nibuf=4 ;number of local ILLOP buffers

0100 locipt: .blkw 1 ;pointer into following
0102 locill: .blkb ibuf1+nibuf*words ;local Trap/regs table

.comnt |
Snapshot area. Gets a copy of uiloop thru procbt as of the last lock
timeout, non-logical illopr, unexpected quit, or snapshot trap (Fxxx),
with the following priorities: If the area is "empty" (snapshot R15
is a 2, to say "running"), any snapshot, quit, lock timeout, or
non-logical illopr will overwrite it. Snapshots and lock timeouts
set R15 to 0 ("half-halted"). In this state, any non-logical
illopr or unexpected quit will still overwrite the snapshot.
but the next snapshot or lock timeout trap won't. Any of
the high-priority events will set the simulated halt (!) bit, and some
other bits in the left half, in R15. DDT will interpret these bits,
and the processor will hang in STAGE, if the processor is among the
set in debug mode (DEBUGM). In the IMP, the snapshot will eventually
be packaged up and sent off to TENEX by the tlog process, which sets
the R15 back to 2 to say the snapshot is empty. DDT will cause a .
processor to rejoin the system by setting its R15 to 2 (G or
P) when debugging. Likewise, DDT X and Z set the odd bit in R15
to halt processors during debugging.
|

 snap:
 bltmyr: ;BLT will reference here for registers of running procs
0142 .b1kw 1 ;DDT "R0": illopr that triggered this snapshot
0144 .b1kw 7 ;R1-R7: saved regs 1-7 at latest event
0152 .b1kw 1 ;R8: status at illopr
0154 .b1kw 1 ;R9: PC at illopr or lock timeout
0156 .b1kw 1 ;R10: last restart call (WST)
0158 .b1kw 1 ;R11: PC at last "program in a loop" (FO02)
015A .b1kw 1 ;R12: address referenced in last unex. quit (FO01)
015C .b1kw 1 ;R13: status for unex. quit
015E .b1kw 1 ;R14: PC at last unex. quit
 bltmyc: ;BLT will reference here to "start" or "stop" me
0160 .b1kw 1 ;R15: pseudo-control register (see above)
 bltmym: ;BLT will reference here for maps of running procs
0162 .b1kw 4 ;maps at time of snapshot
016A .b1kw 2 ;32-bit system time at snapshot
016E .b1kw 1 ;last PID before snapshot
0170 .b1kw 2 ;copy of my coupler number and processor bit

0174 pidget: .b1kw 5 ;list of pids, last = pid3
 000A pidtot=-pidget ;length of pidget

017E temp1: .b1kw 1
0180 temp2: .b1kw 1
0182 temp3: .b1kw 1
0184 temp4: .b1kw 1

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 22
STAGEK.PLR;1 PAGE 7

```
;stage local variables
0186    loczer: .blkw 0 ;begin zeroing here for local init
0186    quitfl: .blkw 1 ;non-zero means quit handler is running
0188    wstage: .blkw 1 ;what stage running
018A    consol: .blkw 1 ;address of console if exists, else 0
018C    wdis: .blkw 1 ;bits on disable stages, turn off from right
018E    ws1sp: .blkw totsts ;stack pointer save for wsleep.

01A2    ttime: .blkw 1 ;time stage is next set to be run
01A4    1clock: .blkw 1 ;clock we are using to time stage system
01A6    jtime: .blkw 1 ;the rtc reading last time we got a jiffy
01A8    svtime: .blkw 1 ;local clock reading at last stage

01AA    utime: .blkw 1 ;when next to run stage LK

0026    loczel=-loczer ;how much to zero for local init

01AC    mysegs: .blkb nmseg ;my picture of common memory (MD)
01B4    smdtim: .blkw 1 ;next time stage MD may run
01B6    smdf1g: .blkw 1 ;flag to let stage MD do comprtr fixes
01B8    newcom: .blkw 1 ;new value for map com. in stage MD.

01BA    srkker: .blkw 1 ;stage RK kernel found if not odd
01BC    srkrel: .blkw 1 ;comrel at start of loop

01BE    scdbus: .blkw 1
01CO    procd: .blkw 1 ;word for discovering processors IPC job 0

01C2    smmok: .blkw 1 ;page found of correct type
01C4    smmspa: .blkw 1 ;page found of spare type
01C6    smmfre: .blkw 1 ;page found with least important type
01C8    smmfty: .blkw 1 ;type of free page we found
```

```
;Define the local stacks

.macro stack name,len    ; define stack name to be len words long.
.xall
name'beg: .blkw len      ; stack.
name'end: .blkw 0         ; upper limit.
name'stack: .blkw 0       ; initial stack pointer.
.xall
.endm

;Define the shared interrupt stack:
stack i,<4>

01CA
01D2
01D2

; Define stage private stacks next:
stack 1k,<7>   ;Stage Local Kernel checksum.

01D2
01EO
01EO
01EO      stack md,<6>   ;Stage Memory Discovery.

01EC
01EC      stack rk,<7>   ;Stage Rely Kernel checksum

01FA
01FA      stack bd,<7>   ;Stage Bus Discovery.

0208
0208      stack cd,< H8>  ;Stage Coupler Discovery.

0218
0218      stack rc,<6>   ;Stage Rely Checksum.

0224
0224      stack lc,<7>   ;Stage Local Checksum.

0232
0232      stack mm,< D17> ;Stage Memory Management.

0254
0254      stack id,<6>   ;Stage I/O Discovery.

0260
0260      stack ar,< Df5> ;Stage Application Reliability.

027E

; Then processor common stack:
page Lc1Stk

0288      stack 1,1stkln     ; define 1stack.

02F8
02F8

02C8      s1stack=lbeg+<s1stk1*words>      ;use bottom in system (STAGE,DDT)
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 24
STAGEK.PLR;1 PAGE 9

;Local STAGE Constants Area
page LCode
;local page checksum block
02F8 0000 0000 localc: .blkw 1 ;checksum on all of local
02FA 4000 0000 hotlim: 1ocend ;limit
;Local kernel checksum block
02FC 0000 0000 1kerck: .blkw 1 ;checksum on just local kernel
02FE OFOC 0000 kerlim: 1keren ;limit
0300 4048 OAO8 0000 1ocbeg
0304 4048 OA1C 0000
0308 4048 09DE 0000
030C 0000 0000 debugm: 0 ;bits for procs to be in debug mode .
030E 0000 0000 offdis: 0 ;bits to hang stages (don't use bit 1)
0310 0000 0000 jpol1: .blkw 1 ;patched by ddt.
;Faked PID to end PID read loop, configured by Stage BD
0312 00FC 0000 pid3: nopid
;parameter for maximum strip time
0314 00C8 0000 mmaxst: maxstr
;Clock list entries
0316 0001 1\$25.6: .word 1 ;filled in later
0318 0001 1\$1.6: .word 1
;user interrupt handlers
031A 07D8 0000 attnpat: rfail ;gets ATTN interrupts
031C 07D8 0000 illopatt: rfail ;gets ILLOPs
;user ILLOP table overflow handler
031E 07B0 0000 illopov: pillow ;called if CILLOV fills

.comnt |
tables to control I/O discovery and inter-processor communication
bsadrs tells what local bus addresses to use.
|

0320 E000 bsadrs: bsadit ;table of addresses of I/O buses
0322 F000 0004 bsadil=-bsadrs ;length of basdrs
0324 8000 bsadrm: bsadmt ;memory addresses
0326 8000 0004 bsadm1=-bsadrm ;length of bsadrm

.comnt |
Word for removing buses. Correspond to entries in bsadrs,
where bit 1 = basdrs+0, bit 2 = bsadrs+2, bit 4 = bsadrs+4 ...
maintained by stage BD
|

0328 0000 buskil: 0

.comnt |
Memkil table. A bit on says don't use the memory. Bits are packed
16 per word, for 64 memories in 4 words
Word 0 has bits for memories 0 (bit 1) to 1E00 (bit 8000)
Word 2 is memories 2000-3E00, etc.
|

0004 memkil: .rept nmseg/2
0
032A 0000 .endr
032C 0000
032E 0000
0330 0000

0332 Defpage LKPatch.org .,limit <.+LKP1en>
1kerp: .blk 1 LKP1en ;patch space within kernel

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 26
STAGEK.PLR;1 PAGE 11

```
;stage constants, in tables by stage
;initial dispatch for stages MD and up

0352 0AAA wpcint: smd00,srk00 ;local kernel stages
0354 0C90
0356 416A sbd00,scd00,src00 ;common kernel stages
0358 42C6
035A 4FF0
000A nlocst=-wpcint :how many stages in kernel

;common code stages initialized from common file
;system dispatch imitialized from target system file

;initial stack pointers for stages MD and up.

035C 01EC wspint: mdstack,rkstack ; local kernel stages
035E 01FA
0360 0208 bdstack,cdstack,rcstack ;common kernel stages
0362 0218
0364 0224
0366 0232 lcstack,mmstack,idstack,arstack
0368 0254
036A 0260
036C 027E
```

```
;table of consensuses to join

036E 0000    contab: 0
0370 4096      segcon  ;stage MD
0372 40A8      stgcon  ;stage RK
0003      locsts=<.~-contab>/words ;stages in local
;here to end are in rel vars
0374 5D50      concom: buscon ;stage BD
0376 5D5C      coucon  ;stage CD
0378 5E28      rckcon  ;stage RC, used by BLT, DDT
037A 5E2E      loccon   ;stage LC
037C 5E36      memcon   ;stage MM
037E 5E90      iocon    ;stage ID
0380 5EA4      inicon    ;stage AR
0000      .iif nz totsts-<<.~-contab>/words>, .error totsts defined wrong
0382 0000      0       ;stage (system)
0010      1com1=.-concom ;locks unlocked in stage init

;table of fixit words by stage

0384 0000    fixtab: 0      ;stage LK: none
0386 409C      segfix   ;stage MD
0388 40AE      stgfix   ;stage RK
038A 5D58      busfix   ;stage BD
038C 5D62      coufix   ;stage CD
038E 0000      0       ;none for stage RC
0390 5E34      locfix   ;stage LC
0392 5E3C      memfix   ;stage MM
0394 5E96      iofix    ;stage ID, only use is sclear
0396 5EA2      inifix   ;stage AR
                  ;stage (system): none

;Prototype memory test pattern for Stage MD

0398 0000    smdcon: 0,-1, HAAAAA, H5555
039A FFFF
039C AAAA
039E 5555
0008      1mdcon=.-smdcon
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 28
 STAGEK.PLR;1 PAGE 13

```
; ***< STAGE System -- Variables on Every Page >*** .

PAGE Vars

0000      ..foo=0%.      ;remember start of user vars if any

4080      .=m0+bbcwin    :start right after couplers

4080      smdbuc: .blkw 1 ;bucket to just store in
        locdef wmlock.<;memory test lock - page 28>
4084      smdblk: .blkb 1mdcon   :place to do a memory test .
408C      slfptr: .blkw 1 :pointer to this page
        locdef slflk.<;locked copy (+2) of slfptr - page 28>.
4090      comprt: .blkw 1 ;map of current comm page
4092      comtst: .blkw 1 ;timer word for comprt fixing
4094      sytime: .blkw 1 ;this page's copy of the master system time
4096      segcon: .blkw 1
        locdef ,<;memory discovery consensus - page 28> .
409A          .blkw 1
409C      segfix: .blkw 1 ;processors who would change memseg table
409E      memseg: .blkb nmseg   ;bit table of existing memories
40A6      memtot: .blkw 1 ;number of pages of memory

40A8      stgcon: .blkw 1 ;consensus for stage RK
        locdef ,<;Common Kernel Discovery Consensus - page 28>
40AC          .blkw 1
40AE      stgfix: .blkw 1 ;procs who'd fix rely kernel
40B0      comrel: .blkw 1 ;common kernel page address. odd->reload

40B4      .=.+ HA& HFFFO+4      :for a nice boundary for pagebc
        ;see cpage macro above

40B4      intime: .blkw 1 ;init timer held by timeout
40B6      cksum: .blkw 1 ;page checksum
40B8      tlimit: .blkw 1 ;top limit
40BA      pginit: .blkw 1 ;init routine this page if not 0
40BC      topntr: .blkw 1 ;pointer to config/timeout table
40BE      type4k: .blkw 1 ;page type this page

40C0      pagebc: .blkw 0 ;page beginning in common

0000      .1if nz ..foo
        .= ..foo      ;restore user's page start. default to pagebc
```

60C0 .=m1#. ;rest of vars through map 1 please

;Table of Traps.
;Contains 16 10-word buffers, each of which has
;Trap num, proc mask, count, and R1-R7

;Buffer definitions

0000 page Dummy
.0
0000 table cilbuf
0000 cilloc: .blkw 1 ;Trap num
0002 cilpro: .blkw 1 ;Proc mask
0004 cilcnt: .blkw 1 ;Count
0006 cilreg: .blkw 7 ;Registers
endtable cilbuf

0000 page Vars

60C0 cilovf: .blkw 1 ;number of Traps missed on overflow
60C2 cilops: .blkb cilnum*1cilbuf ;room for CTLNUM Traps
6162 cilend: .blkw 0

.if df PSE

;PSE illop tables for NCC here instead

0008 pilnum=cilnum ;same number of buffers
6162 pilovp: .blkw 2 ;two overflow indices
6166 pilops: .blkb pilnum*1cilbuf ;ILLOPs go here
6206 pilend: .blkw 0

.endc ;df PSE

6206 dinit: .blkw 1 ;nz - display Traps, od - display both
6208 dhalt: .blkw 1 ;password 1ADE} => stop all procs if debug mode

620A sidflg: .blkw 1 ;flag: set 0 if I/O config changes.

;Paper tape reader interrupt variables here

0001 .if nz PTR
0040 ptrbl= H40 ;chars in ptr interrupt buffer
620C ptrflg: .blkw 1 ;1:read in,2:(verify),4:running
620E ptrip: .blkb 1 ;input buffer pointer
620F ptrop: .blkb 1 ;output buffer pointer
6210 ptrst: .blkw 1 ;state if active
6212 ptrcs: .blkw 1 ;checksum so far
6214 ptrsr: .blkw 1 ;secondary return temp
6216 ptrfb: .blkw 1 ;first byte temp
6218 ptrbc: .blkb 1 ;byte count remaining
6219 ptrdn: .blkb 1 ;how much to adjust ".."
621A ptrb: .blkb ptrbl ;interrupt buffer
625A .blkb 0

.endc ;nz PTR

;Local STAGE Code begins here

page LCode

.comnt |
Non-existant memory ("Quit") Interrupt handlers.
initialize H2E and H3E to go to q50 and q70 respectively
retry a data access if it hasn't failed recently
patterned quits have password as next instruction
password is "nop foo" where foo is where to go if quit.
"nop" itself isn't a password
if password = "nop .-2", a branch is next, followed by dispatch
uquit is pointer to a routine for unexpected quits
|

03A0 0335 q50: rtm qx
03A2 4818 0028 lda r1,%abrt0
03A6 9004 br q1

03A8 0335 q70: rtm qx
03AA 4818 0038 lda r1,%abrt1
03AE 6061 q1: lda r6,(r1)+ ;get address he was after
03B0 3018 0054 sta r1.quitv
03B4 6051 lda r5,(r1)+ ;and his status
03B6 3518 0186 eorm r1,quitf1
03BA 9A60 ifnot zero ;not a QUIT in QUIT handler
03BC 7028 0064 lda r2,quitad ;old address
03C0 7038 0068 lda r3,quitpc ;old pc
03C4 3068 0064 sta r6,quitad
03C8 3058 0066 sta r5,quitst
03CC 7051 lda r5,(r1) ;new pc
03CE 3058 0068 sta r5,quitpc ;can now stand quits in handler
;check for instruction fetch quit
if r6 <> =words(r5) & r6 <> =2*words(r5)

03D2 4E6D 0002
03D6 914E
03D8 4E6D 0004
03DC 914B
03DE 6045 lda r4,(r5)+ ;fetch his instruction
03EO 48F0 lda r7,=0 ;in case clock got the QUIT
03E2 7668 01A4 if r6 <> 1clock ;QUIT not from local clock
03E6 9105
03E8 70F8 01A4 lda r7,@1clock ;if not recent, it's not a retry
03EC 7178 0062 sub r7,quitmm
endif
;now see if it's a recent QUIT and retry
if r7 <= D15 & r2 = r6 & r3 = quitpc

03F0 4EFF
03F2 9C22
03F4 4E26
03F6 8120
03F8 7638 0068
03FC 811D
03FE 48F1 lda r7,=1
0400 3078 0062 sta r7,quitmm ;indicate a hard quit
0404 3178 0056 subm r7,quitrt ;retry must have failed
0408 4FC8 if r4.bit. = H8 ;extended instruction
040A 9^2
040C ! add r5,=words ;skip extended : ess

endif

```
040E 7045          lda r4,(r5)      ;check for quit pattern
0410 4874          lda r7,r4 :get the nop offset
0412 A2F8          sll r7, H8       ;prepare for signed offset
;must be NOP with non-zero offset
0414 9A06          if zero } r4 > =0 } r4 > = H80FF

0416 4EC0
0418 9C04
041A 4E48 80FF
041E 8C07
0420 4078 0494      call qsubr      ;fix quit stuff
0424 0735          mtr qx
0426 E001          Trap 1,<;Unexpected Quit--mem ref fail (H) - page 31>
0428 4088 0052      jmp @uquit
endif
042C A4F7          sra r7,7
042E 4A57          add r5,r7
0430 3058 0068      sta r5,quitpc
0434 901B          else        ;QUIT we should retry
0436 7058 0062      lda r5,quittm ;last QUIT retry succeed?
043A 4ED1          if r5 <> =1    ;last QUIT wasn't real
043C 9105
043E 3028 0058      sta r2,rtryad ;save address
0442 3038 005A      sta r3,rtrypc ;save pc
endif
0446 3278 0062      addm r7,quittm ;remember current time
044A 4874          lda r7,r4      ;adjust xr for retry
044C 4BF7          and r7,=7
044E A2F1          sll r7,1
0450 48A1          lda r2,=1      ;amount to adjust
0452 3228 0056      addm r2,quitrt ;count retry attempts, failures undo this
0456 A1C4
0458 9B02          ifnot minus
045A 48A2          lda r2,=words
endif
045C 8403          if carry      ;auto-decrement
045E 322F 0068      addm r2,quitpc(r7) ;:qx ;mostly
endif
0462 4FC1          if r4 .bit. =1    ;auto-increment
0464 9A03
0466 312F 0068      subm r2,quitpc(r7) ;:qx
endif
046A 3518 0186      eorm r1,quitfl ;clear quit flag and resume code
046E 0735          mtr qx
0470 0433          ret quitst
endif      ;QUIT on instruction fetch
0472 4078 0494      call qsubr      ;common quit stuff
0476 0735          mtr qx      ;refetch regs
; Trap 53,<;Quit on instr fetch (DESTGN PROB) (H) - page 31>
```

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 32
STAGEK.PLR;1 PAGE 17

```
0478 900C      else          ;QUIT in QUIT handler
047A 3068 005E  sta r6,qqhad ;save QUIT address
047E 7031      lda r3,(r1)   ;and PC
0480 3038 0060  sta r3,qqhpc
0484 48B1      lda r3,=1    ;and count counter
0486 3238 005C  addm r3,qqhct ;since a Trap could loop here
048A 4890      lda r1,=0    ;fix quit things; don't touch quitf1
048C 4078 0494  call qsubr
0490 4088 0052  endif
                jmp @uquit
```

;subroutine to set up quit stuff

```
routine qsubr,nosave, arg r1, uses r2,uses r6
0494 3518 0186  eorm r1,quitf1      ; fix quit flag.
0498 4828 0101  lda r2,=<locqut_H8>}pchalt
049C 3028 0096  sta r2,uctrl       ; set pseudo-halt and condition
04A0 48E6      lda r6,=3*words    ; record quit's characteristics.
repeat
04A2 502E 0064  lda r2,quitad(-r6) ;copy quit parameters.
04A6 302E 0090  sta r2,uquitd(r6)  ;to unexpected quit place.
04AA 88FC      until loop
endrepeat
04AC 4807      endroutine qsubr
                ;system unexpected quit handler
04AE 4048 0A1C  sysuq: jsb r4,wst
```

```
;level 1 interrupt. maybe remote power fail

04B2 033D    level1: rtm ix
04B4 7018 0000    lda r1,%1v11+%devno      ;remote power fail?
04B8 4E91        if r1 = =1      ;yes
04BA 8108
04BC E016        Trap 26.<;Rmt pwr fail E/F Bus (NOT PROCS) (H) - page 23>
04BE 7018 0328    lda r1,buskil      ;ignore if some amputated
04C2 8A03        if zero       ;none are
04C4 4048 09AO    jsb r4,haltus      ;pause, then restart
04C8 9008        endif
04CA 4868 01D2    else          ;not a power fail
04CE 40F8 031A    lda sp,=istack
04D2 8A03        call @attnpat      ;call user routine
04D4 4048 09DA    if fail       ;user didn't handle it
04D8 073D        jsb r4,wops      ;unexpected interrupt.
04DA 0401        endif
04D8 073D        mtr ix
04DA 0401        ret %1v11+%pstat

;Include DDT routine if patched

ROUTINE DDTATTN, NOSAVE, ARG R1, USES R2
04DC 7028 0004    LDA R2,%LVL1+%CURPC
04EO 4E18 FF80    IF R1 <> =%AREG1 } R2 <> =0      ;Is this RESET/ATTN?
04E4 8103
04E6 4EA0
04E8 9103
04EA 4FF0        FAIL RETURN      ;No, do standard processing
04EC 4007        ENDIF
04EE 4048 09DE    JSB R4,DSTAND      ;Go off to DDT.
04F2 4807        ENDRoutine DDTATTN
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 34
STAGEK.PLR;1 PAGE 19

```

        ;paper tape reader interrupt code
0001    .if z PTR
              level12=wops      ;no ptr interrupts
        .iff ;z PTR
04F4 FC30    ptrad: ptpadr      ;ptr address

04F6 033D    level12: rtm ix      ;level two interrupt routine
04F8 7018 0008    lda r1,%lv12+%devno
04FC 7618 04F4    if r1 > ptrad ;some other device}
0500 9103
0502 4048 09DA    jsb r4,wops   ;serve one ptr only
        endif
0506 7021    lda r2,(r1)      ;;%sr
0508 8922    if odd      ;go away if ptr busy still (how?)
050A 7039 0008    lda r3,%dr(r1)
050E 4848 FFFF    lda r4,=-1
0512 7048 608C    lda r4,m1/s1fptr
0516 8001    ifnot quit
        endif
0518 7078 0000    lda r7,mapvar
051C 3078 FC02    sta r7,%map1
0520 7868 620E    ldat r6,ptrip :put the char in the buffer
0524 183E 621A    stab r3,ptrb(-r6)
0528 8803    if loop      ;buffer wrap
052A 4868 0040    lda r6,=ptrb1
        endif
052E 3868 620E    stab r6,ptrip
0532 49E1    sub r6,=1
0534 8A03    if zero      ;buffer wrap
0536 4868 0040    lda r6,=ptrb1
        endif
053A 7E68 620F    cmpb r6,ptrrop
053E 8105    if equal      ;buffer filled up}
0540 4878 2000    lda r7,=%L2 ;disable our interrupts
0544 3478 000A    iorm r7,%lv12+%pstat      ;stop if buffer is full
        endif
0548 3048 FC02    sta r4,%map1 ;restore the map
        endif
054C 073D    mtr ix
054E 0405    ret %lv12+%pstat
        endc ;z PTR

```

```
.comnt |
Level 4 interrupt handler (Jiffy, local power fail and restore)
|
0550 033D      sjif:   rtm ix
0552 7018 0018    lda r1.%1v14+%devno
0556 A592        rra r1,2      ;dispatch on three reason bits
0558 8404        if carry     ;power fail = 2 bit
055A E022        Trap 42,<;Proc Pwr Fail (H) - page 35>
055C 4048 09AO    jsb r4,haltus ;halt buddy too, if any
055E             endif
0560 8904        if odd      ;power restore = 4 bit
0562 E021        Trap 41,<;Proc RSTRTD after Pwr fail (H)- page 35>
0564 4048 OA1C    jsb r4,wst   ;if bus was reset
0566             endif
0568 9B04        ifnot minus  ;jiffy should set 1 bit
056A 073D        mtr ix
056C E023        Trap 43,<;illeg level 4 intrpt (H) - page 35>
056E 9032        else       ;got a jiffy
0570 4868 01D2    lda sp,=istack ;get us an interrupt stack
0574 4078 05D4    call rclock  ;clock stuck?
0578 7628 01A6    if r2 = jtime ;yes}
057C 810B
057E 7018 01A4    lda r1,lclock ; don't find this clock
0582 4078 0DCC    call fndclk ; try to find a new clock
0586 8A03        if fail      ; couldn't find any clock}
0588 E017        Trap 27,<;Can't find an RTC(CALL MAINT if RSTRT fails)(H) - page 35>
058A 9002        page 35
058C E004        else
058E 4048 OA1C    Trap 4,<;RTC stppd--switched to new RTC (H) - page 35>
0590             endif
0592             jsb r4,wst ; reset stages.
0594             endif
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 36
 STAGEK.PLR;1 PAGE 21

```

0592 3028 01A6      sta r2,jtime ;remember this reading
0596 7128 0314      sub r2.mmaxstr ;give system a strip-time extra
059A 7128 01A2      sub r2,ltime ;should stage have already run?
059E 9B19      ifnot minus ;not yet time to worry
05A0 7018 001C      lda r1,%l14+%curpc
05A4 4838 9AFE      lda r3,=bz .-4
05A8 7631      if r3 = (r1) ;locking instruction?
05AA 8104
05AC 4078 0654      call snaplok ;yes, snap and unlock it
05B0 9010      else ;maybe next one is
05B2 4A94      add r1,=2*words
05B4 7631      if r3 = (r1)

05B6 8104
05B8 4078 0654      call snaplok ;was a lock
05BC 900A      else
05BE 4928 0500      sub r2,=stgrat
05C2 9B07      ifnot minus ;wait a bit more
05C4 4994      sub r1,=2*words ;restore PC
05C6 3018 008E      sta r1,ujiffy
05CA 073D      mtr ix
;      Trap 2,<;Proc fail in LOOP mod/sftw timeout (H/S) - page 36>
05CC 4048 0A1C      jsb r4,wst
;      endif
;      endif
;      endif
;      endif
05D0 073D      mtr ix
;      endif
05D2 040D      ret %l14+%pstat

;subroutine to read the RTC reliably
; called from jiffy and stage AR

routine rclock,nosave,result r2.uses r4
repeat
  lda r2,@1clock ;read the clock
  lda r4,@1clock ;then read it again
  sub r4,r2
  05DE 4F48 FFF0      while r4.bit. = HFFFO ; can differ by up to 1500 usec.
  05E2 9A05
  05E4 48C1      lda r4,=1 ;count retries
  05E6 3248 0050      addm r4,clokrt
  05EA 90F5      endrepeat
  05EC 4807      endroutine rclock

```

;*** illoopr code - catch logical Traps ***

```
05EE 033D    illp0: rtm ix          ;processor 0 (even)
05F0 4818 0020  lda r1,%illop0   ;illoopr level interrupt address
05F4 9004      br illc

05F6 033D    illp1: rtm ix          ;processor 1 (odd)
05F8 4818 0030  lda r1,%illop1   ;odd proc interrupt
05FC 6021      illc: lda r2,(r1)+  ;get the illegal instruction
05FE 6031      lda r3,(r1)+  ;copy processor status
0600 3038 0088  sta r3,iret
0604 7031      lda r3,(r1)      ; and PC to return vector
0606 3038 008A  sta r3,iret+words
060A 4868 01D2  lda sp,=istack   ;get the interrupt stack
060E 40F8 031C  call @illopat   ;user illopr processing (if any)
0612 8A1C      if fail        ;user routine didn't handle
0614 4D28 E000  eor r2,=Trapv
0618 4F28 F000  ifnot r2 .bit. = HF000   ;got an E-trap
061C 8A04
061E 4078 06F4  call trapcnt   ;record the logical trap only
0622 9014      else           ;Not E-illoopr. Take a snapshot
0624 7049 FFFC  lda r4,-4(r1)  ;refetch the illop
0628 4890      lda r1,=0      ;eventual snapped R15
062A 4E28 1ADE  if r2 = =dhpass  ;got a FADE
062E 8104
0630 4818 0401  lda r1,<locfad_ H8>}pchalt   ;halting codes
0634 9009      else
0636 4E28 1FFF  ;now see if a -1 (FFFF) or other (non-snap) illoopr
063A 9104      if r2 = =crash?trapv } r2 .bit. = HE000
063C 4F28 E000
0640 9A03
0642 4818 0201  lda r1,<locilo_ H8>}pchalt   ;halt codes
0646 4078 067C  endif
064A 4892      endif
064C 3218 008A  call snapill   ;record this snapshot
0650 073D      endif
0652 0444      lda r1,=words ;skip over the trap
                  addm r1,iret+words
                  mtr ix
                  ret iret
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52
STAGEK.PLR;1 PAGE 23

PAGE 38

71

;Routines for creating snapshots for Traps or Hung Locks
;SNAPLOK unlocks the lock, and creates the snapshot and
;trap table entry. SNAPILL does snapshot and trap table.
;TRAPCNT just counts the trap in the (local or common) table.

```
0654 1076    routine snaplok, arg r1, uses r1-r4
0656 4994    sub r1,-2*words ;get index to point at instruction
0658 6041    lda r4,(r1)+ ;get index reg no.
065A 4BC7    and r4,=7
065C 9A04    ifnot zero ;got an indexed instruction
065E A2C1    s11 r4,1
0660 704C 0078  lda r4,ix-words(r4) ;get its old contents
endif
0664 7241    add r4,(r1) ;must always lock with 2-wd inst
0666 487C C000  lda r7,=-m0(r4) ;must be mapped address
066A 8B04    if minus ;woops. don't touch
066C E014    Trap 24,<;Hung on invalid s/w lock - page 38>
066E 4048 OA1C  jsb r4,wst
endif
0672 48A1    lda r2,=1
0674 3024    sta r2,(r4) ;unlock the lock now
0676 4824    lda r2,r4 ;setup for snapshot
0678 4890    lda r1,=0
```

067A 9002 entry snap11, arg r1-r2, arg r4, uses r1-r4
067C 1076

067E 3418 0096 iorm r1,uctrl ;set up pseudo-control register
0682 3048 0078 sta r4,uiilop ;save trap or lock address
0686 48B8 lda r3,= H8 ; number of maps * 2.
0688 4848 A08E lda r4,=s1f1k ; locked self pointer on each page.
repeat
068C 4878 0032 lda r7,= D50 ; try to get lock up to 50 times.
0690 4858 FFFF lda r5,=-1 ; in case we get a quit
0694 7054 lda r5,(r4) ; get locked self p.
0696 8009 ifnot Quit
0698 8A05 repeat while zero ; we don't have lock.
069A 49F1 sub r7,=1 ; count first try, too.
069C 9B03 until minus
069E 7054 lda r5,(r4) ; We know it won't quit now.
06A0 90FC endrepeat
06A2 4ED0 if r5 <> =0 ; we got lock.
06A4 9102
06A6 3054 sta r5,(r4) ; release lock.
endif
endif
06A8 4948 2000 sub r4,=m1-m0 ; next window
06AC 105B 0098 sta r5,umap(-r3) ; store that map value
06B0 88EE until loop
endrepeat
06B2 7078 0160 lda r7,b1tmvc ;what's in snapshot now?
06B6 990D ifnot odd ;:pchalt ;might overwrite
06B8 7038 0096 lda r3,uctrl ;what snap to copy?
06BC 9903 if odd } r7 <> =0 ;has more priority
06BE 4EFO
06C0 9108
06C2 4838 0032 lda r3,=snap1n ;size of snapshot
repeat
06C6 504B 0078 lda r4,snapbg(-r3)
06CA 304B 0142 sta r4,snap(r3) ;copy important stuff out
06CE 88FC until loop
endrepeat
endif
endif
06D0 48C0 lda r4,=0 ; clear simulated ctl reg.
06D2 3048 0096 sta r4,uctrl
06D6 4811 lda r1,r1 ;got a bad illop?
06D8 9A0D ifnot zero
06DA 4E28 1ADE if r2 = #dhpss ;did we get a FADE?
06DE 8108
06E0 7038 00D0 lda r3,mapvar ;yes, first reset vars map
06E4 3038 FC02 sta r3,%map1
06E8 3028 6208 sta r2,dhalt ;then halt everyone
06EC 8001 ifnot quit
endif
endif
06EE 4048 OA1C jsb r4,wst ;yup, back to stage now
endif

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 40
STAGEK.PLR;1 PAGE 25

```
;enter here to just count the Trap, save regs

06F2 9002    entry trapcnt, arg r2, uses r1, uses r3-r4
06F4 1076

06F6 7018 60BE  lda r1,m1/type4k      ;what page type?
06FA 8009  if nquit & r1 = =vartyp   ;good vars page
06FC 4E18 0020
0700 8106
0702 4818 007A  lda r1,=ix          ;where regs are
0706 4078 0734  call illcnt        ;record Trap number in common
070A 9010  else
070C 7018 0100  lda r1,locipt      ;next reg
0710 4B18 0030  and r1,=ibuf1+words<nibuf-1> ;force reasonable value
0714 2029 0102  sta r2,locill(r1)+ ;into buffer
0718 48B0  lda r3,=0
               repeat
071A 607B 007A  lda r7,ix(r3)+      ;how much to copy
071E 2079 0102  sta r7,locill(r1)+  ;until r3 = =nsuer*words
0722 4EBE
0724 81FB  endrepeat
0726 3018 0100  sta r1,locipt
               endif

072A 6006  endroutine snaplok     ::snapill trapcnt
```

```
:ILLCNT

;count logical traps in common
;will manage two tables optionally
;also action on overflow may be modified
;R2 has trap number, R1 points to registers to copy

;tables to drive the routine

072C 60C2    .if df PSE      illbuf: cilops,pilops      :illopr buffer addresses
072E 6166
0730 6162    illend: cilend,pilend   :ends of same
0732 6206

.iff    ;PSE
      illbuf: cilops,-1      :illopr buffer addresses
      illend: cilend,0       :ends of same
.endc

0734 1076    routine illcnt,arg r1-r2,uses r1,uses r3-r4

0736 48C4    lda r4,=illend-illbuf    ;how many ILLOP tables
repeat
0738 503C 072C  lda r3,illbuf(-r4)  ;next illop buffer
073C 9927    ifnot odd             ;real buffer
repeat
073E 7073    lda r7,(r3)
0740 8002    if quit } zero       ;a free entry
0742 8A04
0744 4078 0790  call illcop      ;set up its regs
0748 900C    break ::success    ;found usable entry
endif    ::success
074A 4E72    until r7 = r2       ;got same entry
074C 910A
074E 4A38 0014  add r3,=1cilbuf
0752 763C 0730  if r3 = illend(r4)  ;exceeded buffer
0756 8104
0758 40F8 031E  call @illopov    ;call overflow-handler
075C 9002    break
endif
075E 90F0    endrepeat
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 42
STAGEK.PLR;1 PAGE 27

```
;Here when we found an entry to use (SUCCESS)
;or table is full (FAIL)

0760 9A0F      if success           :got a usable entry
0762 48F1      lda r7,=1
0764 727B 0004  add r7,cilcnt(r3)
0768 8001      ifnot quit
endif
076A 207B 0004  sta r7.cilcnt(r3)+ 
076E 7078 00A8  lda r7.procbt
0772 7473      ior r7.(r3)    ::cilpro
0774 8001      ifnot quit
endif
0776 3073      sta r7.(r3)    ::cilpro
0778 3408 6206  iorm r0,dinit   :signal a change
077C 9007      else               :just overflow then
077E 48F1      lda r7,=1
0780 7278 60C0  add r7.cilovf   ;bump overflow cnt
0784 8001      ifnot quit
endif
0786 3078 60C0  sta r7.cilovf
endif
078A 4EC0      endif
078C 81D6      until r4 = =0      ;done list
endrepeat

078E 6006      endroutine il1cnt
```

```
;ILLCOP
;Routine to set up the trap buffer for a new trap
;in the common trap area (CILOPS, etc.)
;called with R3/ pointer to trap buffer,
;R1/ pointer to saved registers.R2/trap num
```

```
0790 1076    routine i11cop,arg r1-r3
0792 1056    PUSH R5
0794 2023    sta r2,(r3)+ ;:c11loc
0796 48F0    lda r7,=0      ;clear the count
0798 2073    sta r7,(r3)+ ;:cilpro   ;and proc mask
079A 2073    sta r7,(r3)+ ;:cilcnt
079C 48DE    lda r5,=<1cilbuf-cilreg> ;how much to copy
repeat
079E 6071    lda r7,(r1)+ ;next reg to save
07A0 2073    sta r7,(r3)+ ;to trap area
07A2 49D2    sub r5,=words ;copied a word
07A4 8AFD    until zero
endrepeat
07A6 6056    POP R5
07A8 499E    sub r1,#nsuer*words ; and R1
07AA 4938 0014 sub r3,=1cilbuf ;back to beginning

07AE 6006    endroutine i11cop
```

```
;PILLOV
;Pick the next ILLOP buffer to fill if all full
;Uses them in round robin order
;R1 points to registers to copy, R2 is trap number,
;R3 is address of buffer to use, R4 says which set of ILLOP buffs
;Always succeed return to say "found a buffer".

.if df PSE ;This is a Platform feature

07B0 1076 routine pillow,arg r1-r4,result r3

07B2 703C 6162 lda r3,pilovp(r4) ;trap to overwrite
07B6 8002 if quit } minus ;check for troubles
07B8 8B02
07BA 48B0 lda r3,=0 ;default if any
endif
07BC 487B 0014 lda r7,=1cilbuf(r3) ;next buffer to use
07CO 4E78 00AO if r7 >= =pilnum*1cilbuf ;at end of buffers
07C4 9202
07C6 48F0 lda r7,=0 ;clear offset
endif
07C8 307C 6162 sta r7,pilovp(r4) ;remember for next one
07CC 723C 072C add r3,i1lbuf(r4) ;address in trap table
07D0 4078 0790 call i1lcop ;set up for our trap

07D4 6006 endroutine pillow

.iff ;df PSE
    pillow = rsucceed
.endc ;df PSE

;definitions for null routines

07D6 4807 routine rsucceed,nosave
endroutine

07D8 4FF0 routine rfail,nosave
    fail return
07DA 4007 endroutine
```

;***** local memory stage code *****

.comnt |
Stage entry mechanism

The stage cycle either exits to loop and reenters at sj6, or will
fail some test and stay in the stage system.
Stage 0 (LK) unilaterally enables interrupts.
Map 1 is set to mapvar, map 2 to mapcom.
Maps 0 and 3 are set to mapcom (local stages) or to comrel
|

07DC 1076 routine wsleep,local r1-r5
07DE 1016
07EO 1026
07E2 1036
07E4 1046
07E6 1056

07E8 7018 0188 lda r1,wstage ;which stage running.
07EC 8A07 if zero ;remember next time to run stage 0 (LK)
07EE 70F8 01A4 lda r7,@1clock ;time now
07F2 4A78 0500 add r7,=stgrat ;next time to run stage 0 (LK)
07F6 3078 01AA sta r7,utime
endif
07FA 3069 018E sta sp.wslsp(r1) ;save stack pointer.
07FE 7078 018C lda r7.wdis
0802 4F78 0400 if r7 .nbit. =totstb ;could run system now
0806 8A19
0808 7078 0160 lda r7,bltmyc ;pseudo-halted
080C 890D if odd & debugm .bit. procbit ;and debugging
080E 7078 030C
0812 7778 00A8
0816 9A08
0818 4868 02C8 lda sp,=s1stack ;operating system stack area
081C 40F8 0310 call @jpoll ;run DDT
sj2: call polbit ;(DDT returns with a JMP)
0820 4078 0990 else
0824 9009
0826 7078 00B0 setmap m0,maprel ;where main system pointer is
082A 3078 FC00
082E 4868 02F8 lda sp,=lstack ;main system stack
0832 40F8 501E call @jloop ;run it
sj6: ;(system returns with a JMP)
endif
0836 9003 else
0838 30F8 018A sta r7,@consol ;hung: do STAGE 1ites
endif
;(MORE)

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 46
STAGEK.PLR;1 PAGE 31

```
083C 70F8 01A4    lda r7,@1clock      ;set up when stage is next to run
0840 4A78 0500    add r7,=stgrat
0844 3078 01A2    sta r7,1time
0848 7018 0188    lda r1,wstage      ;next stage runnable?
084C 7078 018C    lda r7,wdis
0850 6779 0EB8    if r7 .bit. bittab+2(r1)+ ;no, try other stuff
0854 9A10
0856 4F78 0020    if r7 .nbit. =bltstb   ;running enough for BLT
085A 8A05
085C 4078 0990    call polbit
0860 7078 018C    lda r7,wdis      ; needed below.
          endif
0864 7018 0188    lda r1,wstage
0868 70B8 01A4    lda r3,@1clock      ; time now
086C 7138 01AA    sub r3,utime      ; time to go to stage 0
0870 9B02    ifnot minus
0872 4890    lda r1.=0          ;start again at stage 0
          endif
          endif
```

```
0874 4E18 0014    if r1 >= =totsts*words
0878 9202
087A 4890        lda r1,=0                      ;wrap around to stage 0.
                endif
087C 3018 0188    sta r1,wstage
0880 70D8 01A4    lda r5,@1clock
0884 7068 00AO    lda r6,stime
0888 7048 00AE    lda r4,mapcom
088C 992C        ifnot odd
088E 3048 FCO4    sta r4,%map2
0892 7048 8090    lda r4,m2#comptr
0896 8002        if quit
0898 901A

                ;got a quit from com page - smash all com pointers

089A 0888        inh .L4                      ;takes a while
089C 4818 7FFF    lda r1,= H7FFF
08A0 4878 7E00    lda r7,<nmseg* D8-1>* H200   ;largest odd number
08A4 48E6        lda r6,=nmseg-words
                repeat
08A6 4858 8000    lda r5.=sign
                repeat
08AA 775E 032A    ifnot r5 .bit. memkil(r6) ; can we use?
08AE 8A06
08BO 3078 FCO2    sta r7,%map1
08B4 3018 6090    sta r1,m1#comptr
08B8 8001        ifnot quit
                endif
                endif
08BA 4978 0200    sub r7,= H200      ; next page.
08BE A6D1        sr1 r5,1      ; bit for this page;
08C0 8AF5        until zero
                endrepeat
08C2 49E2        sub r6.=words     ; next memseg
08C4 8BF1        until minus
                endrepeat
08C6 E011        Trap 21,<:Lost our comm pg (H/S) - page 47>
08C8 4048 OA14    jsb r4,wstcom
                now restart stage
                endif

08CC 3048 00AE    sta r4,mapcom
08D0 3048 FCO4    sta r4,%map2
08D4 7068 8094    lda r6,m2#sytime
08D8 8006        ifnot quit
08DA 7028 9EAA    lda r2,m2#sytim2
08DE 8003        ifnot quit
08EO 3028 00A2    sta r2,stim2
                endif
                endif
                endif
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 48
 STAGEK.PLR;1 PAGE 33

```
;Here with R6/ best estimate of System time (SYTIME or STIME)
;R7/ WDIS

08E4 7668 00AO      if r6 <> stime           ;time ticking along
08E8 9104
08EA 3058 01A8      sta r5,svtime        ;Make jiffy time track
08EE 9014      else                      ;system stop or no SYTIME?
08F0 4835      lda r3,r5
08F2 7158 01A8      sub r5,svtime
08F6 A4D9      sra r5, H9
08F8 9AOF      ifnot zero            ;51.2 ms gone by
08FA 4F78 0400      ifnot r7 .bit. =totstb ; running all stages yet?
08FE 8A05
0900 7078 030C      ifnot debugm          ;and no debugging?
0904 8A02
0906 E00B      Trap 13,<;Sys missed a tick (H) - page 48>
                endif
                endif
0908 3038 01A8      sta r3,svtime        ;tick in an orderly fashion
090C 4AE1      add r6,=1
090E 4844      lda r4,r4
0910 9903      ifnot odd             ;MAPCOM's there
0912 3068 8094      sta r6,m2#sytime
                endif
                endif
                endif
0916 3068 00AO      sta r6,stime         ;our copy of system's time

;Now set up maps for consensus, dispatch

091A 4E96      if r1 >= =locsts*words    ; common stage
091C 9207
091E 7048 80BO      lda r4,m2#comrel   ;get com rely ptr
0922 4B48 7E00      and r4,=nmseg* H8* H200- H200
0926 3048 00BO      sta r4,maprel     ;keep maprel current.
                endif
092A 3048 FCO0      sta r4.%map0      ;rely or com page
092E 3048 FCO6      sta r4.%map3      ;locking on com or rely page
```

```
;Stay in a consensus, done automatically by STAGE
;A consensus is a three-word block:
; <smoothed consensus>
; <next consensus (lock)>
; <time to update consensus>
;CONTAB has a table by Stage number of consensi to join

0932 9923      ifnot odd    ;maprel mapcom ;page for consensus
0934 7049 036E  lda r4,contab(r1)
0938 9A20      ifnot zero   ;consensus to join.
093A 48D0      lda r5.=0    ;value if quit
093C 705C 6002  lda r5,m3-m0+2(r4)
0940 8005      if nquit & zero  ; got it, but locked.
0942 8A04
0944 705C 6002  lock r5,m3-m0+2(r4)
0948 9AFE

094A 7458 00A8  endif
094E 4838 FFFF  ior r5,procbt
0952 703C 0004  lda r3,=-1      ;in case QUIT
0956 8003      lda r3,4(r4)
0958 7138 00A0  ifnot quit
095C 4F38 FFOO  sub r3,stime
0960 9A0A      endif
0962 3054      if r3 .bit. = HFFOO    ;time to update consensus
0964 7038 00A0  sta r5,(r4)
0968 4A38 003C  lda r3,stime
096C 303C 0004  add r3,=stgcyc
0970 7058 00A8  sta r3,4(r4)
0974 305C 0002  lda r5,procbt
0978 7078 00D0  endif
097C 3078 FC02  sta r7,%map1
0980 7069 018E  lda sp,wslsp(r1)  ;restore stack pointer for this stage.

0984 6056      endroutine wsleep
0986 6046
0988 6036
098A 6026
098C 6016
098E 6006

;POLBLT - poll BLT after fixing maps

0990 7048 00B0  ;routine polbit
0994 3048 FC00  polblt: lda r4,maprel
0998 3048 FC06  sta r4,%map0
099C 4008 45FE  sta r4,%map3
: setmap <m0,m3>.maprel      ; setup to call blt
:         jmp blt                 ; poll block transfer.
:         call blt
: endroutine polbit
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 50
 STAGEK.PLR;1 PAGE 35

```

:Various entries to stage

; jsb r4, to WS, WST, WSTCOM, DSTAND, HALTUS, WOPS

;Power fails come here - stop buddy and wait awhile,
;then get us and buddy into stage again (if we're still powered)

09AO 0210      haltus: rst %f1           ;assume no buddy
09A2 7018 FF3E    lda r1,%cpu1+%ctrl   ;see a proc?
09A6 800A      ifnot quit
09A8 7018 FF20    lda r1,%cpu1+%reg0   ;running?
09AC 8002      if quit
09AE 9006
09B0 4891      lda r1,=pchalt        ;stop our buddy
09B2 3018 FF3E    sta r1,%cpu1+%ctrl   ;see a proc?
09B6 8002      ifnot quit
09B8 0290      sst %f1           ;we successfully halted buddy
                                . . .
                                . . .
                                . . .
09BA 4818 07D0      lda r1,= D2000       ;now wait awhile
                                repeat
09BE 5021      lda r2,(-r1)
09C0 88FF      until loop
                                endrepeat
09C2 850B      if f1           ;got a buddy to start
09C4 4818 OA1C    lda r1,=wst         ;where to restart
09C8 3018 FF20    sta r1,%cpu1+%reg0   ;running?
09CC 8006      ifnot quit
09CE 3008 FF28    sta r0,%cpu1+%reg4   ;buddy starts "here"
09D2 4892      lda r1,=pcrun
09D4 3018 FF3E    sta r1,%cpu1+%ctrl   ;ctrl
                                . . .
                                . . .
09D8 9022      br wst
                                . . .

;Got an interrupt we weren't expecting

09DA E006      wops: Trap 6.<;Unexpctd intrpt (Poss RESET/ATTN) (H) - page 50>
09DC 9020      br wst           ;restart stages
                                . . .

:enter here to start in STAGE+DDT (if DEBUGM set up)

09DE 48E1      dstand: lda r6,=pchalt     ;just stop us in STAGE+DDT
09EO 3068 0160    sta r6,b1tmyc      ;set my halt control
09E4 901C      br wst           ;enter stage
                                . . .

:processors are normally started here

09E6 3038 00AA      wstini: sta r3,procno    ;r3 preset by startup
09EA 4048 OA1C      jsb r4,wst

```

```
; WSUB - routine to smash all init timers.

routine wsub,nosave,uses r1-r3

09EE 4818 0050    lda r1,%novarp          ; for all pages. . .
repeat
09F2 5029 00B0    lda r2,1map(-r1)      ; next page in table.
09F6 9906          ifnot odd           ; any page there.
09F8 3028 FC06    sta r2,%map3        ; bum off of the clear.
09FC 7038 AOB4    lda r3,m3#intime   ; smash init timer
0AO0 8001          ifnot quit         ; ignore quits.
endif
0AO2 88F8          until loop        ; endrepeat

0AO4 4807          endroutine wsub

;start here from loader

0AO6 0000          setup: hlt          ;continue to start system
0AO8 4818 03A0    ws:    lda r1,q50       ;need quit handler
0AO9 3018 002E    sta r1,%abrt0+%servc ;in even guy at least
0A10 4078 09EE    jsb r7,wsub        ;set inits, please
0A14 4818 7FFF    wstcom: lda r1,H7FFF   ;largest positive odd int
0A18 3018 00AE    sta r1,mapcom     ;fall into wst

;here to reset wst code

0A1C 3048 008C    wst:   sta r4,uwst     ;save reason for restarting stage
0A20 0888          inh .L4          ;jiffy would need a clock
0A22 4890          lda r1,=0        ;how much to clear . .
0A24 4828 0026    lda r2,loczel1   ;repeat
0A28 101A 0186    sta r1,loczer(-r2)
0A2C 88FE          until loop
endrepeat
0A2E 601A 0142    lda r1,snap(r2)+   ;bum - sets R2 = 2
0A32 8A03          if zero           ;we may have just been loaded
0A34 3028 0160    sta r2,b1tmyc   ;pcrun
endif
0A38 4868 01E0    lda r6,=1kstack   ;needed by SBAD and LK.
0A3C 4078 0D3E    call sbad
;fall into stage LK
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 52
 STAGEK.PLR;1 PAGE 37

.comnt |
 Stage LK - Checksum Local Kernel

Set up interrupts, enable. Checksum local stage kernel. If okay,
 look for local I/O (console, etc.). Discover some RTC.
 Someday, make local I/O discovery table-driven.

```

| ;s1k00:           ;Checksum Local Kernel.
|   repeat          ;forever.
|
OA40 48AO      lda r2.=0           ;make location 0 halt us
OA42 3022      sta r2,(r2)
|
OA44 4818 003E lda r1,%abrt1+%servc ;setup interrupt vectors
|   repeat
OA48 603A OA9A lda r3,inter(r2)+ ;vector value.
OA4C 3031      sta r3,(r1)
OA4E 4998      sub r1,%abrt1-%ilop1 ;step to next vector.
OA50 8BFC      until minus
|   endrepeat
|
OA52 4818 04AE lda r1,=sysuq       ;general unexpected quit handler.
OA56 3018 0052 sta r1,uquit        ;where to go on unex. quit
|
OA5A 4892      lda r1,=bitsnk       ;where to store lites if no console
OA5C 7038 FF80 lda r3,%aregi       ;console is here, if at all.
OA60 8003      ifnot quit          ;if console exists
OA62 4818 FF80 lda r1,%aregi       ;console addr. for lights code.
|   endif
OA66 3018 018A sta r1,consol       ;tell lites where to go.
|
OA6A 4078 0E10 call cksub,1kerck,1ocend ;cksum local kernel.
OA6E 02FC
OA70 4000
OA72 9103      ifnot equal         ;cksum bad
|   Trap 5,<;Local Kernel Cksum Broken-FATAL to Proc (H/S) - page 52>
OA74 E005
age 52
OA76 0000      hlt                  ;wait for help
|   endif
|
OA78 7018 00AA lda r1,procno       ;my proc number
OA7C 48A1      lda r2.=1
OA7E A321      r11 r2,r1          ;my proc bit
OA80 3028 00A8 sta r2,procbt

```

```
0A84 4890      lda r1,=0           ;any clock will do.
0A86 4078 0DCC  call fndclk
0A8A 8A04      if fail
0A8C E015      Trap 25,<:Stge LK: can't find a clk - page 53>
0A8E 4048 OA1C  jsb r4,wst       ;start stages up again.
0A90             endif

0A92 0809      enb .L1+.L4       ;enable jiffies and attention.
0A94 4078 0D50  call sokay        ;pass this stage

0A98 90D4      endrepeat

0A9A 03A8      ;initial interrupt vectors.
0A9C 05F6      inter: q70
0A9E 03AO      i11p1
0AA0 05EE      q50
0AA2 0550      i11p0
0AA4 09DA      intab1
0AA6 04F6
0AA8 04B2
```

.comnt |
Stage MD - Discover all useable memory pages
First useable page will become communication page
Maintains a bit-table of these pages, segtab, in common.
Checks self-pointers and comm-pointers for each page, fixes
them if wrong. Builds copy of memseg, mysegs, in local.
Maintains memtot, the number of pages in all memseg words.

```
|  
  
smd00:  
repeat  
OAAA 48FO      lda r7.=0  
OAAC 3078 01B6  sta r7,smdflg           ;permit comprtr fixes this pass  
OAB0 7078 OOAO  lda r7,stime  
OAB4 4A78 OFOO  add r7,=memrat  
OAB8 3078 01B4  sta r7,smdtim           ;when nxt to permit them  
  
repeat  
OABC 48F1      lda r7.=1  
OABE 3078 01B8  sta r7,newcom           ;no com page yet  
OAC2 48AO      lda r2.=0               ;first mem segment  
  
repeat  
OAC4 48BO      lda r3.=0               ;first of segment  
OAC6 4890      lda r1.=0               ;memory picture  
  
repeat  
OAC8 707B OEB6  lda r7,bittab(r3)     ;bit for nxt memory  
OACC 777A 032A  tst r7.memkil(r2)  
OADO 8A08      if zero  
OAD2 4078 OB58  call smdtst           ;use this memory?  
OAD6 9A03      if succeed  
OAD8 741B OEB6  ior r1,bittab(r3)     ;do memory test  
                  build picture of memory  
OADC 4078 07DC  endif  
                  call wsleep  
                  endif  
OAE0 4AB2      add r3.=2  
OAE2 4E38 0020  until r3 = = H20  
OAE6 81F1      endrepeat  
  
OAE8 301A 01AC  sta r1,mysegs(r2)     ;save mem picture  
OAEc 4AA2      add r2.=2  
OAEe 4EA8      until r2 = =nmseg  
OAF0 81EA      endrepeat
```

:now see if we match the system

OAF2 48A8 1da r2,=nmiseg
OAF4 7078 00AE 1da r7,mapcom
OAF8 9921 ifnot odd
OAF8 48D0 1da r5,=0
OAF8 48B0 1da r3,=0

repeat
 sub r2,=2
 until minus
 1da r1,mysegs(r2)
 1da r7,memseg(r2)
 if quit } r7 <> r1

OAFE 49A2 1da r2,=2
OBO0 9B16 until minus
OBO2 701A 01AC 1da r1,mysegs(r2)
OBO6 707A 409E 1da r7,memseg(r2)
OBOA 8003 if quit } r7 <> r1
OBOC 4E71
OBOE 9108
OB10 4078 0D2E call sfxbad
OB14 9103 ifnot equal
OB16 4AD1 add r5,=1
OB18 900A break
endif
 sta r1.memseg(r2)
endif

OB1E 4E90 repeat until r1 = =0
OB20 9105
OB22 4AB1 add r3,=1
OB24 4B19 FFFF and r1,=-1(r1)
OB28 90FB endrepeat

OB2A 90EA endrepeat

OB2C 4ED0 if r5 = =0
OB2E 8105
OB30 3038 40A6 sta r3,memtot
OB34 4078 0D22 call sclrok
endif

OB38 9008 else
 Trap 50,<No usable comm mem (MEM PROB. CALL MAINT) (H) - page 55>
OB3A E028
age 55
OB3C 4878 7FFF 1da r7,= H7FFF
OB40 3078 00AE sta r7,mapcom
OB44 4078 0D3E call sbad
endif

OB48 3008 01B6 sta r0,smdflg
OB4C 7078 01B4 1da r7,smdtim
OB50 7178 00AO sub r7,stime
OB54 8BB4 until minus
endrepeat

OB56 90AA endrepeat

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 56
 STAGEK.PLR;1 PAGE 41

;SMDTST is the memory test routine for Stage MD.
 ;It gets quits from nonexistent memory and test memory,
 ;couplers, locking. It also does the self pointer and
 ;comptr tests. R2 contains the segment index and R3 the memory index.

```

OB58 1076      routine smdtst,arg r2-r3,local r1-r3.uses r5
OB5A 1016
OB5C 1026
OB5E 1036

OB60 4813      lda r1,r3                      :calculate map from indices
OB62 A298       sll r1, D8                    :low-order map bits
OB64 A7A4       rrl r2,4                     :high-order map bits
OB66 4A12       add r1,r2                   :map setting this page
OB68 48F8       lda r7,=%map3+words-%map0
repeat
OB6A 101F FC00  sta r1,%map0(-r7)
OB6E 88FE       until loop
endrepeat
OB70 3008 4080  sta r0,smdbuc                :memory there?
OB74 8002       if quit
OB76 9007
OB78 6036       fail return
OB7A 6026
OB7C 6016
OB7E 6076
OB80 4FF0
OB82 4007
endif
OB84 7078 A082  lda r7,wmlock
OB88 8005       if nquit & zero
OB8A 8A04
OB8C 7078 A082  lock wmlock
OB90 9AFE
endif
OB92 4078 OBAA  call smdts2                 ;do main tests
OB96 8A04       if fail
OB98 3008 A082  unlock wmlock
OB9C 90EE       fail return
endif
OB9E 3008 A082  unlock wmlock
OBA2 6036       endroutine smdtst
OBA4 6026
OBA6 6016
OBA8 6006

```

;SMDTS2 - Main memory test subroutine

OBAA 1076 routine smdts2, arg r1, uses r2-r3, uses r5
OBAC 48D8 1da r5,=1mdcon
repeat
OBAE 507D 0398 1da r7,smdcon(-r5)
OBB2 307D 4084 sta r7,smdb1k(r5) ;init mem test
OBB6 88FC until loop
endrepeat
OBB8 48D8 1da r5,=1mdcon
repeat ;test memory
OBBA 507D 0398 1da r7,smdcon(-r5)
OBBE 483D 4000 1da r3,=m0(r5)
repeat
OBC2 702B 0084 1da r2,0//smdb1k(r3)
OBC6 8003 if quit } r7 <> r2 ;failing mem or coupler
OBC8 4E72
OBCA 9105
OBCC 8000 notrap ;illopr here to diagnose memory failures
OBCE 6076 fail return
OBDO 4FF0
OBD2 4007 endif
OBD4 4A38 2000 add r3,=m1-m0
OBD8 4F3B C000 tst r3,=-m0(r3)
OBDC 8BF3 until minus ;try all maps
endrepeat
OBDE 707D 4084 1da r7,smdb1k(r5)
OBE2 8002 if quit } nonzero ;quit or locking failed?
OBE4 9A03
OBEG 8000 notrap ;illopr here to diagnose memory failures
OBE8 90F3 fail return
OBEA 88E8 endif
until loop
endrepeat

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52
STAGEK.PLR;1 PAGE 43

PAGE 58

```
:selfpointer test

OBEC 7028 00A8    lda r2,procbt
OBFO 7058 408C    lda r5,s1fptr
OBF4 8012          ifnot quit
OBF6 4B58 FEOO    and r5,=mapmsk
OBFA 4E15          if r1 <> r5
OBFC 910E
OBFE 3058 FCO2    sta r5,%map1      ;to page pointed to
OC02 7078 6084    lda r7,m1/smdblk
OC06 8009          if nquit & zero   ;garbage or nonzero ptr?
OC08 8A08
OCOA 3028 4084    sta r2,smdblk     ;write into locked area
OCOE 7628 6084    if r2 = m1/smdblk ;match on other page?
OC12 8103
OC14 8000          notrap           ;i1opr- memory really appears twice
OC16 90DC          fail return
                     endif
                     endif
                     endif
                     endif
                     endif
                     sta r1,s1fptr      ;fix s1fptr
                     lda r7,=0          ;case if quit
                     cklock r7,s1f1k
                     if nquit & zero   ;now busy-wait
                     lock r7,s1f1k
                     endif
                     and r7,= HFF       ;now get low-order
                     ior r7,= H100(r1) ;get map plus 100} (its a lock)
                     unlock r7,s1f1k  ;and put it back
OC18 3018 408C
OC1C 48FO
OC1E 7078 A08E
OC22 8005
OC24 8A04
OC26 7078 A08E
OC2A 9AFE
OC2C 4B78 00FF
OC30 4C79 0100
OC34 3078 A08E
```

:now check comprtr

```
OC38 7078 01B8 lda r7,newcom ;this com page?
OC3C 890B if odd ;haven't got one
OC3E 3018 01B8 sta r1,newcom ;remember candidate
OC42 7618 00AE if r1 <> mapcom ;match old mapcom?
OC46 9105
OC48 3018 00AE sta r1,mapcom ;save new one
OC4C 4078 0D3E call sbad ;hang in this stage
endif
OC50 4871 lda r7,r1 ;new mapcom
endif
OC52 7018 4090 lda r1,comptr
OC56 8017 if nquit & r1 < r7 ;this page points lower
OC58 4E17
OC5A 8215
OC5C 7018 01B6 lda r1,smdflg ;fixes this pass?
OC60 8A10 if zero ;I want to fix it?
OC62 3528 4092 eorm r2,comtst
OC66 8002 if quit
OC68 9006
OC6A 3028 4092 sta r2,comtst ;fix parity
OC6E 8000 notrap
OC70 90AF fail return
OC72 9006 else
OC74 7728 4092 tst r2,comtst
OC78 9A03 ifnot zero ;can't yet?
OC7A 8000 notrap
OC7C 90A9 fail return
endif
endif
OC7E 9003 else
OC80 8000 notrap
OC82 90A6 fail return
endif
endif
OC84 3078 4090 sta r7,comptr ;rewrite comprtr
OC88 48F0 lda r7,=0 ;don't use map3} (parity)
OC8A 3078 4092 sta r7,comtst ;clear comprtr fix cntrs
OC8E 6006 endroutine smdts2
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 60
STAGEK.PLR;1 PAGE 45

.comnt | Stage RK - Find and Checksum Common Reliability Kernel

Find the common memory kernel of STAGE, which ought to have a good checksum and reside on the comrel page. If the one there isn't a kernel, or has bad checksum, look for another copy through memory. When we succeed, we can execute the rest of the STAGE kernel code from common. Tries to find a whole good rely page, but settles for just a kernel if necessary. Low order bits of comrel are state of rely page: 0=okay.

1=please reload

```
OCE8 4A18 0200      add r1,= H200      ;select next page.
OCEC 4B18 7E00      and r1,=nmseg* H8* H200- H200
OCFO 7618 01BC      until r1 = srkrel
OCF4 81DB
                     endrepeat

OCF6 7018 01BA      lda r1,srkker      ;until kernel not found.
OCFA 9B0E            until minus

OCFC 7078 40BO      lda r7,comrel
ODO0 4E17            if r1 = r7          ;we agree with system.
ODO2 8104
ODO4 4078 0D22      call sclrok        ;Enable next stage.
ODO8 9006
ODOA 4B78 7E00      and r7,=nmseg* D8* H200- H200
ODE0 7678 01BC      break if r7 = srkrel ;restart calc. if smm did something
OD12 9102
                     endif
OD14 90BE            endrepeat

OD16 4078 0D2E      call sfxbad       ;vote to fix comrel.
OD1A 8103            if equal         ;we can fix.
; Trap 11,<;adj comrel (If cont. RELD IMP) (H/S) '- page 61>
                     sta r1,comrel    ;r1 is proposed comrel.
OD1C 3018 40BO      endif
                     endrepeat
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 62
STAGEK.PLR;1 PAGE 47

.comnt |
Stage Advancing, Hanging, and Fixing Subroutines

sclrok - Do sclear + sokay

sfxbad - Do sbad + sfixed

sbad - Hang us in present stage; disable all later stages.

sokay - Proceed; reinit next stage if it's just starting.

sclear - Clear our bit out of current fixit word.

sfixed - Try to fix, return equal if we can fix.

|
OD22 1076 routine sclrok,uses r4
OD24 4078 OD90 call sclear
OD28 4078 OD50 call sokay
OD2C 6006 endroutine sclrok

OD2E 1076 routine sfxbad,local r4
OD30 1046
OD32 4078 OD3E call sbad
OD36 4078 ODAC call sfixed
OD3A 6046 endroutine sfxbad
OD3C 6006

OD3E 1076 routine sbad,uses r4
OD40 7048 0188 lda r4,wstage ; present stage
OD44 48FO lda r7,=0
OD46 717C 0EB8 sub r7,bittab+2(r4) ; bits for all succeeding stages
OD4A 3478 018C iorm r7,wdis ; disable them
OD4E 6006 endroutine sbad

```
OD50 1076      routine sokay,local r3.uses r4
OD52 1036
OD54 7048 0188  lda r4,wstage          ; present stage
OD58 703C 0EB8  lda r3,bittab+2(r4)    ; bit for next stage
OD5C 7738 018C  if r3 .bit. wdis       ; next stage not ruing
OD60 9A16
OD62 7538 018C  eor r3,wdis          ; let it run
OD66 7438 030E  ior r3,offdis        ; keep debugging bits on
OD6A 3038 018C  sta r3,wdis
OD6E 4F38 0400  if r3 .bit. =totstb     ; later stage to start up.
OD72 9A0D
OD74 703C 035C  lda r3,wspint(r4)    ; initial stack pointer.
OD78 707C 0352  lda r7,wpcint(r4)    ; initial program counter.
OD7C 4ECA
OD7E 9203
OD80 707C 500C  lda r7,cwpcin-nlocst(r4)
OD84 1073      endif
OD86 49BA      sta r7,(-r3)           ; push program counter.
OD88 303C 0190  sub r3,%reg5-%reg0   ; adjust sp for wsleep return.
OD8C 6036      sta r3,ws1sp+words(r4) ; save new sp for wsleep return.
OD8E 6006      endif
endroutine sokay

OD90 1076      routine sclear,uses r4
OD92 7048 0188  lda r4,wstage          ; present stage
OD96 70FC 036E  lda r7,@contab(r4)    ; its consensus
OD9A 7578 00A8  eor r7,procbt        ; take me out.
OD9E 33FC 0384  andm r7,@fixtab(r4)  ; dead procs and me' out.
ODA2 8002      if quit
ODA4 9003
ODA6 30FC 0384  sta r7,@fixtab(r4)    ; fix parity quit
ODAA 6006      endif
endroutine sclear

ODAC 1076      routine sfixit,uses r4
ODAE 7048 0188  lda r4,wstage          ; present stage
ODB2 7078 00A8  lda r7,procbt        ; just me if quit
ODB6 74FC 0384  ior r7,@fixtab(r4)    ; set that wants to fix
ODBA 8001      ifnot quit
ODBC 73FC 036E  and r7,@contab(r4)    ; only live voters
ODCO 76FC 036E  if r7 <> @contab(r4)  ; disagreement
ODC4 9103
ODC6 30FC 0384  sta r7,@fixtab(r4)    ; live voters into fixit
ODCA 6006      endif
endroutine sfixit
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 64
STAGEK.PLR;1 PAGE 49

.comnt |
fnclk - routine to find a local clock.

Accepts, in r1, the address of a clock which it should
not use; returns, in r3, the address of the found clock.
|

ODCC 1076 routine fnclk,arg r1,uses r3-r5,result r3

```
ODCE 7078 0328    lda r7,buskil      ; buses not to use.
ODD2 48D4        lda r5,=bsadil     ; length of bus table.
                  repeat          ; look at next bus
ODD4 577D OEB6    ifnot r7 .bit. bittab(-r5) ; bus enabled
ODD8 8A18
ODDA 704D 0320    lda r4,bsadrs(r5)   ; bus base address
ODDE 483C 0006    lda r3,=rtcadd(r4)   ; where clock lives
ODE2 4E31        if r3 <> r1       ; not same clock as last time.
ODE4 9112
ODE6 7044        lda r4,(r4)       ; must be a PID here.
ODE8 8010        ifnot quit } r4 .bit. = HFFOO
ODEA 4F48 FFOO
ODEE 8A0D
ODFO 7043        lda r4,(r3)       ; RTC here, too?
ODF2 800B        ifnot quit      ; Yes
ODF4 7638 01A4    if r3 <> 1clock
ODF8 9107
ODFA 3038 01A4    sta r3,1clock    ; new local clock
ODFE 4A48 0500    add r4,=stgrat   ; correct local time.
OE02 3048 01A2    sta r4,1time
                  endif
OE06 6006        return
                  endif
                  endif
                  endif
OE08 88E6        until loop      ; no more buses
                  endrepeat
OE0A 6076        fail return    ; couldn't find clock.

endroutine fnclk
```

.comnt |
cksub - Checksum a piece of memory.

Call with r1 set to common memory page address, if any.
Takes two inline arguments: the first is brought to r5 and
is a pointer to the checksum header; the second goes to
r4 and is the maximum upper limit value. The checksum
header is a two-word block at the beginning of the area to be
checksummed holding the checksum in the first word and the
upper limit of the area in the second.

The routine returns equal/not equal; equal implies the
checksum is correct, while not equal can indicate any of a number
of failures. If "equal" is returned, r2 is the address of the
first word past the checksummed region.

|
routine cksub. arg r1, inline r5, inline r4, result r2, uses r3-r5

```
OE10 6057
OE12 6047
OE14 1076
OE16 4078 07DC    call wsleep      ;prevent loops in stages calling us.
OE1A 3018 FCO2    sta r1,%map1   ;get the page again
OE1E 0201          rst %e        ;always return unequal
OE20 7035          lda r3,(r5)   ;current checksum value.
OE22 8032          ifnot quit    ;no problem getting cksum. .
OE24 707D 0002    lda r7,2(r5)   ;cksum upper limit.
OE28 802F          ifnot quit    ;this one ok, too.
OE2A 992D          ifnot odd     ;must be even; odd is garbage.
OE2C 4947          sub r4,r7    ;limit must be less than max' limit.
OE2E 9B2B          ifnot minus   ;and not less than lower limit.
OE30 4975          sub r7,r5    ;initial cumulative cksum.
OE32 9B29          ifnot minus   ;first address to cksum.
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 66
STAGEK.PLR;1 PAGE 51

```
;now, do real checksumming.  
repeat ; until some return happens  
    call wsleep ; first, catch up on sleep  
    sta r1,%map1 ; then, restore map1 for work.  
    lda r7,r2 ; next word address.  
    sub r7,2(r5) ; cksum limit.  
    if minus ; more work to do  
        if r7 <= == H40  
            lda r7,== H40 ; never do more than 40  
        endif  
        repeat ; checksum a piece  
            sub r4,(r2)+ ; checksum a word  
            ifnot quit  
                add r7,=words ; count a word as done.  
                next ifnot zero  
                break  
            endif  
            Trap 15,<;Quit during cksuming (H) - page 66>  
            rst %e ; Return unequal (cksum bad)  
            return ; and done  
        endrepeat  
        else ; no more work to do  
            if r3 = (r5) ; no-one has touched checksum.  
                if r4 = (r5) ; checksum right  
                    return ;  
                endif  
                if r3 <> =ckpass ; password  
                    return ;wrong; tell caller.  
                endif  
                sta r4,(r5) ; store new checksum  
            else ; checksum changed  
                lda r3,(r5) ; start calculation over.  
                lda r4,r3  
            endif  
            lda r2,r5 ; Restart checksumming  
        endif  
    endrepeat ; not done, do another piece.  
endif  
endif  
return ; cksum header broken.  
endif  
Trap 14,<;Quit in cksum param (H) - page 66>  
endroutine cksub
```

.comnt |
MEMTST - check if memory in memseg
Sets %map1 to desired map if page exists.
Accepts memory address in r1, map2 must point to communication page.
|

OE8A 1076 routine memtst,arg r1,local r2,result m1
OE8C 1026
OE8E 4821 lda r2,r1
OE90 A6A8 sr1 r2, D8
OE92 4B28 001E and r2,= H1E ;16 bits per segment
OE96 4871 lda r7,r1
OE98 A3F4 r11 r7,4
OE9A 4BF6 and r7,=nmseg-words ;segment index
OE9C 707F 809E lda r7,m2//memseg(r7)
OEAO 777A 0EB6 ifnot r7 .bit. bittab(r2)
OEAA 8A05
OEAB 6026 fail return
OEAC 6076
OEAD 4FF0
OEAC 4007
endif
OEAE 3018 FCO2 sta r1,%map1
OEB2 6026 endroutine memtst
OEB4 6006

:Table to translate an index to a bit

0010 bittab: .rept D16
1_.rpCnt
.endr
OEB6 0001
OEB8 0002
OEBA 0004
OEBC 0008
OEBE 0010
OECA 0020
OEC2 0040
OEC4 0080
OEC6 0100
OEC8 0200
OECA 0400
OECC 0800
OECE 1000
OED0 2000
OED2 4000
OED4 8000

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 68
 STAGEK.PLR;1 PAGE 53

```
; ***< STAGE SYSTEM -- RELIABILITY CODE >***
```

| | | |
|---|----------|--|
| FC00 0000 | PAGE | Relcode |
| 40C0 | .=pagebc | |
| ***** reliability page tables ***** | | |
| 40C0 | rkerck: | .blkw 1 ;rely kernel checksum |
| 40C2 7016 | rkelim: | m1#rkeren ;Rely kernel limit: checksum |
| 40C4 0ACE | rkepas: | stgpas ;mark to say rely kernel is here |
| 40C6 0000 | prokil: | 0 ;processors not to start |
| 40C8 0000 | prohng: | 0 ;procs to hang at stage ID |
| ;amputation control words | | |
| ;offset defs: | | |
| 0000 | amemio=0 | ;memory or i/o bus (1 bit/bus) |
| 0002 | aproc=2 | ;an individual processor |
| 0004 | abus=4 | ;a processor bus (2 bits/bus) |
| ;tables for manual,system, and random amputation: | | |
| ;must be contiguous | | |
| 40CA 0000 | ampman: | 0 |
| 40CC 0000 | prohlt: | 0 ;procs to halt and not start |
| 40CE 0000 | proamp: | 0 ;proc buses to amputate (proc halts too) |
| 40DO 0000 | ampsys: | 0,0,0 |
| 40D2 0000 | . | |
| 40D4 0000 | . | |
| 40D6 0000 | amtry: | 0,0,0 |
| 40D8 0000 | . | |
| 40DA 0000 | . | |
| ;Tables for initial checksum limits | | |
| 40DC 4000 | ihotlm: | 1ocend ;local memory limit |
| 40DE 7D50 | limtab: | mm1ims ;assemble common page limits |
| 40E0 FFFF | . | |
| 40E2 FFFF | . | |
| 40E4 FFFF | . | |
| 40E6 FFFF | . | |
| 40E8 FFFF | . | |
| 40EA FFFF | . | |
| 40EC FFFF | . | |
| ;Block transfer tables for checksum limits | | |
| ; must be even and positive | | |
| 40EE 02FE | 1cktab: | kerlim,hotlim,0 ;local limits |
| 40FO 02FA | . | |
| 40F2 0000 | . | |
| 40F4 60C2 | rcktab: | m1#rkelim ;rely limits |
| 40F6 60B8 | ccktab: | m1#t1limit,0 ;every page limits |

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 69
 STAGEK.PLR;1 PAGE 54

```
;following table is parallel to bsadrs. Used by stage
;BD to maintain amputation of processors

40FA 0001      bsmaps: bsmptb ;table of what to put in maps (io)
40FC 0001
40FE 0000      bsmapm: bsmptm ;ditto for memories
4100 4000

;Tables to drive Stage MM

;Table of page category offsets

4102 0000      smmbas: 0      ;code pages
4104          ncodep: .blkw 1    ;spare pages (set later)
4106          nsparp: .blkw 1    ;required, desired vars
4108          nvarsp: .blkw 1    ;optional vars
        0008  lmmbas=-smmbas ;list of offsets to check in SMMCHECK
410A          novarp: .blkw 1   ;offset of last page to allocate + 1

;Parallel table (to SMMBAS) of minimum required

smmins:
410C          ncodem: .blkw 1   ;code pages (set later)
410E          nsparm: .blkw 1   ;spare pages
4110          nrequp: .blkw 1   ;Total minimum page number to run system
4112          nvarsm: .blkw 1   ;required vars
              noptvm: .blkw 1   ;optional vars

;Other parameters

4114          ndvars: .blkw 1 ;How many variables pages to allocate before spares
              ;base addresses of io segments. each covers 100} bytes,
              ; for 16 (dec.) devices at 10} bytes per

4116 E100      iobase: iobtab ;assemble io segment base address table
4118 E200
411A F100
411C F200
        0008  nsegs=-iobase ;how much stage 6 finds
              ;tables to cause amputation of individual devices
              ;one bit causes system to ignore the device
              ;each word controls 16 devices, starting with the
              ;corresponding address in iobase (bit 0 = 1}) up to
              ;device at address+FO} (bit 15 = 8000}

        0004  iokill: .rept nsegs/2
              0
411E 0000      .endr
4120 0000
4122 0000
4124 0000

Defpage RKPatch map code.ora ..limit<.FRKPIen>.physical reflod
```


uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 70
STAGEK.PLR;1 PAGE 55

```
; consensus communication variables
; look thru map 0

PAGE Relvars

comar: ;beginning of communication area

;vars for stage BD
5D50 buscon: .blkw 1 ;bus discovery consensus
      locdef ,<;Common Bus Discovery Consensus - page 70>
      .blkw 1
5D54 stgtim: .blkw 1 ;how current are stage vars
5D56 busfix: .blkw 1 ;procs who'd fix usebus
5D58 usebus: .blkw 1 ;common buses that exist

;vars for stage CD
5D5C coucon: .blkw 1
      locdef ,<;processor and bus coupler discovery consensus - page 70>
      .blkw 1
5D60 coufix: .blkw 1 ;fixit for processor discovery
5D62 procx: .blkw 1 ;processors that exist to BD
5D64 prior: .blkw 1
5D66 locdef bbclok,<;lock on bus coupler states - page 70>
      coutab: .blkb nproc      ;table of processor coupler addresses
      coutbl=-coutab          ;its length, *'must be power of 2**'
0010 coubus: .blkb coutbl
5D7A ioctab: .blkw H8       ;max 8 other couplers
      ioctbl=-ioctab
0010 amproc: .blkw H8
5D9A ampcom: .blkw H8
```

```
.comnt |
Block transfer control variables
|
      locdef bltlok,<;Block transfer lock - page 71>
5DBC   bltst: .blkw 1 ;transfer state
      0001    bltact=1      ;active
      0002    bltful=2      ;buffer full
      0004    bltetos=4     ;blt timeout error
      0008    bltcck= H8    ;fix dest checksum for me or common
      0010    bltedq= H10   ;dest quit
      0020    bltedk= H20   ;non-existant dest
      0040    bltedf= H40   ;dest format error
      0080    bltsdf= H80   ;check spare (if any) for differences

      0300    bltctl=<pcrun>pchalt>_ H8      ;bits to ctrl reg
      0400    bltrun= H400   ;start proc in stage
      0800    bltspa= H800   ;use spare page if any
      1000    bltesq=bltedq_ H8      ;source quit
      2000    bltesk=bltedk_ H8      ;no source
      4000    bltesf=bltedf_ H8      ;source bad format
      7074    blters=blteto}bltedq}bltedk}bltedf}bltesq}bltesk}bltesf
              ;all possible errors in BLTST

5DBE   bltto: .blkw 1 ;blt timeout
5DC0   bltadd: .blkw 1 ;current transfer address
      8000    bltlog=sign  ;a logical page of common (if odd)
      0001    bltnlc=1    ;not local memory if set (ie common)
5DC2   bltsiz: .blkw 1 ;transfer size in bytes
5DC4   bltdon: .blkw 1 ;how much in this strip
5DC6   bltbm: .blkw 1 ;blt buffer map
5DC8   bltbfa: .blkw 1 ; and address
5DCA   bltpro: .blkw 1 ;proc who did BLT
5DCC   bltbmk: .blkw 1 ;mask of procs who failed BLT
5DCE   bltdid: .blkw 1 ;which proc we looked into
5DD0   bltpok: .blkw 1 ;PID to poke back
5DD2   bblrst: .blkw 1 ;coupler address of last BBC started
5DD4   bbcbad: .blkw 1 ;coupler address of last BBC failure

5DD6   bltsty: .blkw 1 ;source type
      2000    bltddt= H2000 ;bit if DDT partaking
      4000    bltfak= H4000 ;bit if fake process partaking
      8000    bltfex= H8000 ;bit if external reload (bltrld)
5DD8   bltspm: .blkw 1 ; and proc mask

5DDA   bltdty: .blkw 1 ;dest type,
5DDC   bltdpm: .blkw 1 ; and current mask
5DDE   bltdpi: .blkw 1 ;initial dest mask
5DE0   bltbuf: .blkw D36   ;place for core transfer

5E28   rckcon: .blkw 1 ;Rely checksum consensus (RC)
5E2C   locdef .<;Consensus for Rely page Checksum - page 71>
              .blkw 1
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 72
STAGEK.PLR;1 PAGE 57

```
5E2E      loccon: .blkw 1 ;local checksum consensus (LC)
          locdef ,<;Consensus for Local Checksum - page 72>
          .blkw 1
5E32      locfix: .blkw 1 ;fixit for local checksum reload
5E34
5E36      memcon: .blkw 1 ;stage MM consensus
          locdef ,<;memory configuration consensus - page 72>
          .blkw 1
5E3A      memfix: .blkw 1 ;procs who would fix cmap
5E3C      bufflg: .blkw 1 ;flag to not allocate mem O buffers or vars
5E3E      cmap: .blkb %novarp ;common table of maps (configuration)
5E40
          ;vars for stage ID - i/o discovery
5E90      iocon: .blkw 1
          locdef ,<;consensus for i/o discovery - page 72>
          .blkw 1
5E94      iofix: .blkw 1 ;fixit word for io discovery
5E96      useio: .blkb nsegs ;16 devices per word, parallels iobase
          0008    useiol=-useio ;length of device table
5E98
          0100    seginc= D16*devinc ;16 devices per segment
          FFO0    segmsk=<seginc-1>?-1 ;bits that determine segment
5EA0      prtime: .blkw 1 ;time some stage AR will next start procs
5EA2      inifix: .blkw 1 ;fixit word for page initialization (stage AR)
5EA4      inicon: .blkw 1 ;consensus for initialization stage
          locdef ,<;initialization consensus lock - page 72>
          .blkw 1
5EA8
          ;vars really for stage (system)
5EAA      sytim2: .blkw 1 ; high-order bits of system time.
```

;**** Reliability page code ****

FC00 0000 Page Relcode

.comnt |
Stage BD - Discover Common Buses

First, make sure stage variables in common are current, and
reinitialize if not.

A memory bus exists if its lowest-numbered memory is in the system
(for the purposes of amputation only). An I/O bus exists if its
PID exists. To prevent the discovery of a bus, set its bit
in the manual amputation table (AMPMAN), at offset 0 (AMEMIO).
This word is copied to BUSKIL by this stage, for use by the
rest of the system. Output of this stage is BUSUSE,
which is used by BBC-related activities and I/O discovery.
To remove a memory bus completely, use MEMKIL (stage 1).

|

```
4166 BD68      ilockt: bbclok ;BBC lock
4168 BDBA      b1tlok ; and lock on block transfer
0004      1lockt=-ilockt

        sbd00:
        repeat
            repeat
                call sbdqch
                ;forever
                ;until variables quit or timeout
                ;check all stage variables for quit.
                ;some variable quit.
                until fail
                ;variables watchdog timer.
                lda r7,stgtim
                ;is watchdog still on?
                sub r7,stime
                ;watchdog goes off.
                until r7.bit. = HF000
                ;keep stage vars time
                call sbdtim
                ;build buskil word
                lda r7,ampman+amemio
                ior r7,ampsyst+amemio
                ior r7,amptry+amemio
                sub r7,buskil
                ;update buskil word in local.
                subm r7,1kerck
                ;keeping checksum good.
                addm r7,buskil
                ;used in loop below .
                lda r4,buskil
                lda r5,=0
                ;build a tentative usebus.
                lda r3,=0
                ;index into pidget.
                lda r1,=bsadil+bsadm1
                ;number of busses to explore.
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 74
STAGEK.PLR;1 PAGE 59

```

repeat                                ;for each possible bus.
  ifnot r4 .bit. bittab(-r1) ;bus not killed.

41A4 5749 OEB6
41A8 8A1F
41AA 7029 40FA      1da r2,bsmaps(r1)      ;how to map to bus.
41AE 8916      if odd                   ;IO bus
41BO 7029 0320      1da r2,bsadrs(r1)    ;where in address space bus lives.
41B4 707A 0000      1da r7,pidsto(r2)   ;look for pid
41B8 8010      ifnot quit } r7 .bit. = HFFOO ;PID exists.

41BA 4F78 FFOO
41BE 8A0D
41C0 3028 00AC      sta r2,pid       ;software pid (okay 'cause no sleep)
41C4 4AA2      add r2,=pidrc1     ;setup PIDGET as we find PIDs.
41C6 202B 0174      sta r2,pidget(r3)+ .
41CA 707A 0004      1da r7,rtcadd-pidrc1(r2) ;if RTC tell usebus.
41CE 8003      ifnot quit      ;RTC there.
41DO 7459 OEC6      ior r5,bittab+< H8*words>(r1) ;set RTC bit in USEBUS
                    endif
41D4 7459 OEB6      ior r5,bittab(r1)  ;set bus bit in USEBUS.
                    endif
41D8 9007      else                   ;memory bus.
41DA A3A4      r11 r2,4
41DC 707A 809E      1da r7,m2#/memseg(r2)
41EO 8903      if odd                   ;low memory on bus exists.
41E2 7459 OEB6      ior r5,bittab(r1)  ;include bus in picture.
                    endif
41E6 4E90      endif
41E8 81DE      until r1 = ==          ; no more busses to do.

endrepeat

41EA 4EB0      if r3 = ==            ;no pids found.
41EC 8103
:      Trap 10,<;No PIDs CALL *** BBN MAINT. *** (H) - page 74>
      jsb r4,wst      ;restart system.
      endif
41F2 4878 0312      1da r7,=pid3      ;setup rest of pidget.
      repeat
        sta r7,pidget(r3)+ .
        until r3 >= =pidtot

41F6 207B 0174
41FA 4EBA
41FC 92FD
endrepeat

41FE 7658 5D5A      if r5 = usebus      ;we agree with system's picture.
4202 8104
4204 4078 0D22      call sclrok      ;allow later stages.
4208 9006
420A 4078 0D2E      else
420E 8103      call sfxbad      ;vote to fix usebus.
4210 3058 5D5A      if equal      ;we may fix usebus.
4211 4078 0D2E      sta r5,usebus   ;fix it with our picture.
4212 8104      endif
4213 4078 0D2E      endif

4214 4078 07DC      call wsleep      ;rest a while
4218 9049
endrepeat

```

```
421A 4078 0D2E    call sfxbad           ;vote to fix
421E 811A    if equal            ;okay to fix.
4220 4078 427A    call sbdcclr        ;clear common area
4224 8A08    if fail             ;we got some quit or other here.
                           Trap 44,<;Stge vars mem fail *** CALL BBN MAINT. *** (H) -4226 EO24      if fail
t some quit or other here.
page 75
4228 7018 00B0    lda r1,maprel       ;disable this common page.
422C 4078 42AC    call memdis
4230 4048 0A1C    jsb r4,wst         ;and restart all stages
endif

4234 7018 00A8    lda r1,procbt       ;good init is my bit
4238 48A4    lda r2,=llockt        ;aux lock table
repeat
423A 109A 4166    sta r1,@ilockt(-r2)   ;unlock a lock
423E 88FE    until loop
endrepeat

4240 4828 0010    lda r2,=lcom1        ;how many consen locks
repeat
4244 503A 0374    lda r3,concom(-r2)
4248 301B 0002    sta r1,bitsnk(r3)     ;into bitsnk if no consensus
424C 88FC    until loop
endrepeat
; Trap 22,<;Stge Common Reint (IMP RSTRTED) - page 75>
424E 4078 429C    call sbdtim          ;now fix time
endif

4252 908C    endrepeat
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 76
STAGEK.PLR;1 PAGE 61

```
.comnt |
sbdqch - check stage common variables for quit; fail return if quit.

|  
  
4254 1076      routine sbdqch,uses r1-r2
4256 48A0          lda r2,=0
                    repeat
4258 601A 5D50      lda r1,comar(r2)+  
425C 8002          if quit
425E 9005
                    Trap 20,<;Stge vars area quit--12 TRAP means fix (H) - page 76>
4260 E010
e 76
4262 6076          fail return
4264 4FF0
4266 4007
                    endif
4268 4F28 003F      ifnot r2 .bit. = H3F
426C 8A03
426E 4078 07DC      call wsleep
                    endif
4272 4E28 02AE      until r2 = =1comar
4276 81F1
                    endrepeat
4278 6006          endroutine sbdqch
```

.comnt |
sbdcclr - clear stage common variables; be careful with quits.
|

427A 1076 routine sbdcclr,uses r1-r2
427C 4890 lda r1,=0
427E 4828 02AE lda r2,=1comar
repeat
4282 101A 5D50 sta r1,comar(-r2) ; clear a word.
4286 8008 until quit
4288 701A 5D50 lda r1,comar(r2)
428C 8005 until quit } nonzero
428E 8A04
4290 8802 if loop
4292 6006 return
endif
4294 90F7 endrepeat
4296 6076 fail return
4298 4FF0
429A 4007 endroutine sbdcclr

.comnt |
SBDTIM - maintain the stage variables timer STGTIM.
|

429C 1076 routine sbdtim
429E 7078 00A0 lda r7,stime
42A2 4A78 0258 add r7,=inirat ;keep watchdog fed and awake.
42A6 3078 5D56 sta r7,stgtim
42AA 6006 endroutine sbdtim

.comnt |
memdis - locally remove page in r1 from this processor.
|

42AC 1076 routine memdis,arg r1,uses r2-r3
42AE A698 srl r1, H8
42BO 4821 lda r2,r1
42B2 4B28 001E and r2,= H1E
42B6 A694 srl r1,4
42B8 703A 0EB6 lda r3,bittab(r2)
42BC 3239 032A addm r3,memkill(r1)
42CO 3138 02FC subm r3,1kerck
.42C4 6006 endroutine memdis

:Stage CD - Coupler Discovery

;First, set up the amputation control words PROIOR, AMPROC,
;and AMPCOM. Then proceed to discover the couplers in the
;system, building an answer in COUTAB, COUBUS, and IOCTAB.
;Finally, check processors discovered against PROCEX.

scd00:

repeat :forever

42C6 48B0 lda r3.=0
42C8 48C0 lda r4.=0
42CA 48D0 lda r5.=0
42CC 4878 0012 lda r7.=3*3+words :scan 3 3-word tables
repeat :until thru tables or trouble
42D0 545F 40CA ior r5,ampman(-r7) ;abus proamp :proc busses to amputate
42D4 544F 40CA ior r4,ampman(-r7) ;aproc prohl1 :procs to halt and not run
42D8 543F 40CA ior r3,ampman(-r7) ;amemio :io busses to amputate
42DC 88FA until loop
endrepeat
42DE 4C45 ior r4,r5 :these procs should halt
42EO 3048 5D66 sta r4,proior :and not be in system
42E4 7748 00A8 if r4 .bit. procbt :I should halt
42E8 9A02
42EA 0000 hlt
endif
42EC 4893 lda r1.=3 :procs by pair
42EE 48A1 lda r2.=1 :io busses singly
;;lda r7.=0
repeat :set up amputate tables
42FO 4848 DE79 lda r4.=bbcpas :bare password
42F4 4F15 ifnot r1 .bit. r5 :not amputated
42F6 8A02
42F8 4CC2 ior r4.=bbcf0r
endif
42FA 304F 5D9A sta r4,amproc(r7) :set proc amp table
42FE 4848 DE79 lda r4.=bbcpas :just password again
4302 4F23 ifnot r2 .bit. r3 :allow io bus access
4304 8A02
4306 4CC2 ior r4.=bbcf0r
endif
4308 204F 5DAA sta r4,ampcom(r7)+ :set io amp table
430C A2A1 sll r2.1 :next io coupler
430E A392 r11 r1.2 :next proc pair
4310 89FO until odd :done 8 of each bus
endrepeat
4312 4078 07DC call wsleep :time to rest
4316 7078 4096 lda r7,segcon :these procs are running
431A 3078 01C0 sta r7.proc0 :so assume they exist
431E 4890 lda r1.=0 :current COUTAB index
4320 4828 0010 lda r2.=couplr :first coupler to find
4324 48B0 lda r3.=0 :IOCTAB index

```
repeat
4326 4E28 0080      if r2 < =bbcwin           ;looking for legit coupler
432A 8208
432C 4078 43B0      call scdtst
4330 4078 07DC      call wsleep
4334 7058 01BE      lda r5,scdbus
4338 9003      else
433A 4858 FFFF      lda r5,=-1
endif
433E 9A36      ifnot zero
4340 4FD3      if r5 .bit. =<1_<bsadi1/words>>-1 & r1 <= =coutb1-2
4342 9A15
4344 4E9E
4346 9C13
4348 4842      lda r4,r2           ;found a proc
434A A2C8      s11 r4, H8
434C 4C4A 0001      ior r4,=1(r2)       ;build name pair
4350 6649 5D6A      if r4 <> coutab(r1)+ } r5 <> coubus-words(r1)
4354 8104
4356 7659 5D78
435A 9108
435C 4078 0D2E      call sfxbad          ;some table is wrong
4360 8127      break ifnot equal
4362 3049 5D68      sta r4,coutab-words(r1)   ;await consensus
4366 3059 5D78      sta r5,coubus-words(r1)  ;we'll fix
endif
436A 9020      else
436C 4EBE      if r3 > =ioctb1-words    ;IOCTAB is full
436E 8C14
4370 7018 01C0      lda r1,procד
4374 7618 5D64      if r1 <> proceх
4378 9106
437A 4078 0DAC      call sfixit          ;procs we found
437E 8118      break ifnot equal
4380 3018 5D64      sta r1,proceх       ;mismatch system
endif
4384 4078 0D22      call sclrok          ;can't fix
4388 7018 00AA      lda r1,procno
438C 7879 5D6A      ldab r7,coutab(r1)   ;do the fix
4390 3078 00A6      sta r7,myproc
4394 900D      break
endif
4396 A2D8      s11 r5, H8
4398 4C52      ior r5,r2
439A 665B 5D8A      if r5 <> ioctab(r3)+  ;busses to left half
439E 9106      ;remember the coupler
43A0 4078 0D2E      call sfxbad          ;good io coupler?
43A4 8105      break ifnot equal
43A6 305B 5D88      sta r5,ioctab-words(r3) ;can't fix it
endif
endif
43AA 4AA2      add r2,=words
43AC 90BD      endrepeat
43AE 908C      endrepeat
```

```
;Stage CD Subroutines

;SCDTST
;Check the current coupler on each bus. R2 is the coupler offset,
;R1 is the current index into COUTAB, R3 is IOCTAB index.
;Builds in SCDBUS the picture of which busses this coupler is
;connected to, and in PROCD the picture of which processors exist.

43B0 1076    routine scdtst, arg <r1-r3>, uses <r4-r5>, uses m2

43B2 1036    save r3                                ;not needed for a while
43B4 48B0    lda r3,=0
43B6 3038 01BE  sta r3,scdbus                  ;initially no busses
43BA 48B4    lda r3,=bsadil                  ;for each I/O bus
repeat
43BC 4EB0    until r3 = =0
43BE 911A
43C0 504B OEB6  lda r4,bittab(-r3)            ;bit for this bus
43C4 7748 5D5A  next ifnot r4 .bit. usebus      ;bus exists
43C8 9AFA
43CA 4078 07DC  call wsleep                  ;now rest
43CE 705B 0320  lda r5,bsadrs(r3)            ;bus offset
43D2 4A52    add r5,r2                      ;get to right coupler
43D4 7075    lda r7,(r5)                    ;coupler here?
43D6 8006    ifnot quit                   ;yes, note it.
43D8 3448 01BE  lorm r4,scdbus                ;check BBC, procs
43DC 4078 4458  call scdbbc                 ;got our coupler
43E0 9AEE    next if fail                  ;looking for me}
endif
43E2 7078 00A6  lda r7,myproc                ;my name
43E6 4B78 007E  and r7,=bbcwin-words        ;without odd bit
43EA 4E72    if r7 = r2                     ;looking for me}
43EC 8102    Trap 37,<;Broken Coupler *** CALL BBN MAINT. *** (H) - page 80>

43EE EO1F
e 80
endif
43F0 90E6    endrepeat
43F2 6036    restore r3
```

;Now check memory bus coupler ends

```
43F4 48D4    lda r5,-bsadmi      ;now for all memory busses
43F6 7078 BD68    lock bbclok      ;get BBC
43FA 9AFE
repeat
until r5 == 0
;for each memory bus
43FC 4ED0    if usebus .bit. bittab+bsadrm-bsadrs(-r5)      ;next M bus there
43FE 9128
4400 7078 5D5A    setmap m2,bsmapm(r5)      ;its map setting
4404 577D OEBA
4408 9A22
440A 707D 40FE    1da r4,=bbcfor+bbcpas      ;default: forward enable
440E 3078 FC04    1da r7,=0
4412 4848 DE7B    sta r7,m2(r2)      ;first look for coupler
4416 48FO    ifnot quit
4418 307A 8000    1da r7,m2(r2)      ;see one?
441C 8017    ifnot quit
441E 707A 8000    if r7 == bbcans      ;yes
4422 8014
4424 4E78 2100
4428 810D
442A 707D OEBA
442E 3478 01BE
4432 7049 5D9A    1da r7,bittab+bsadrm-bsadrs(r5) ;coupler's bit
4436 7078 01BE    iorm r7,scdbus      ;this coupler on this bus
443A 4FF3    1da r4,amproc(r1)      ;now for proper password
443C 8A03    ifnot scdbus .bit. =<1_<bsadi1/words>>-1      ;no io coupler
443E 704B 5DAA
1da r4,ampcom(r3)      ;common bus amputate state
endif
endif
4442 304A 8000    sta r4,m2(r2)      ;set proper BBC state
4446 8002    next ifnot quit      ;okay for this bus
4448 90DA
endif
endif
Trap 23,<;QUIT on BCM coupler *** CALL BBN MAINT. *** (H) - page 81>
444A E013
- page 81
endif
444C 90D8    endrepeat
444E 3008 BD68    unlock bbclok      ;done BBC for a while
4452 6006    endroutine scdtst
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE -82
 STAGEK.PLR;1 PAGE 67

```

:SCDBBC
;Routine to maintain BBC state for a given coupler. If any
;processors need to be discovered on this bus, look for
;them using BBC. Results of processor discovery to PROCD.
;R1 has current COUTAB index, R5 has coupler address to use.

;table of BBC map settings to get to processor control registers

4454 FF18 scdtab: <%cpu0+%ctr1>&bbcmask      ;even proc
4456 FF38             <%cpu1+%ctr1>&bbcmask      ;odd proc

4458 1076 routine scdbbc, arg r1-r2, arg r5, local r3, uses r4
445A 1036

445C 7049 5D9A 1da r4,amproc(r1)           ;proper proc amputation state
4460 7038 01CO 1da r3,procd               ;procs already found
4464 7438 40C6 ior r3,prokil             ;plus ones not to look for
4468 A631 sr1 r3,r1                      ;interesting ones to bottom
446A 4BB3 and r3,=3
446C 7078 BD68 lock bbclok                ;do some BBC
4470 9AFE
4472 4EB3 if r3 <> =3                   ;others to 'look for'
4474 9122
4476 4CC4 ior r4,=bbcbak                 ;backwards enable
4478 4078 44C8 call scdset               ;try to turn it on
447C 8A05 if fail                         ;found us (BBCLOCK unlocked)
447E 6036 fail return
4480 6076
4482 4FF0
4484 4007

        endif
4486 4B58 F800 and r5,=iomask            ;which bus
448A 48C4 1da r4,=2*words                 ;for each proc on bus
repeat
        ifnot r3 .bit. bittab(-r4)          ;this proc not found yet
448C 573C 0EB6 1da r7,scdtab(r4)          ;map to this proc
4490 8A10 sta r7,bbcmap(r5)              ;set up BBC window
4492 707C 4454 ifnot quit
4496 307D 008E 1da r7,bbcwin+<%ctr1&bbcwmark>(r5) ;proc there?
449A 800A ifnot quit
449C 707D 0086 1da r7,bbcwin+<%ctr1&bbcwmark>(r5) ;proc there?
44AO 8006 ifnot quit
44A2 707C 0EB6 1da r7,bittab(r4)          ;yes
44A6 A271 sll r7,r1                      ;got the proc
44A8 3478 01CO ior r7,procd              ;set its bit
        endif
44AC 9002 else
        Trap 7.<;BBC map fail (Poss bad coupler) (H) - page 82>
44AE E007
        endif
        endif
until loop
endrepeat
44B2 7049 5D9A 1da r4,amproc(r1)          ;proper amputation
44B6 4A52 add r5,r2                      ;and coupler address
endif

```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 83
STAGEK.PLR;1 PAGE 68

```
44B8 4078 44C8 call scdset           ;restore proper amputate
44BC 8A02      if fail             ;was our coupler (BBCLOK unlocked)
44BE 90E0      fail return
44C0 3008 BD68 unlock bbclok       ;done BBC for now
44C4 6036      endroutine scdbbc
44C6 6006
```

```
;SCDSET
;Subroutine to set the coupler state.
;Called with password to use in R4, coupler address in R5.
;If necessary, the bits to inhibit processor discovery for this
;bus are in R3. R1 has current index into COUTAB, which is PROCNO.
;of even proc on this bus. Processor discovery answer is in PROCD.
;If a QUIT happens, unlock BBCLOCK, check to see if its our
;coupler; if not, reset our coupler number and restart, else
;do processor discovery on our bus and fail return.
;R4 and R5 intact in succeed return.

44C8 1076    routine scdset.arg r1.arg r3-r5.uses r4-r5

44CA 3045    sta r4,(r5)                      ;do the coupler setup
44CC 8002    ifnot quit
44CE 6006    return
44D0 3008 BD68    unlock bbclok
44D4 4841    lda r4,r1
44D6 7078 0054    lda r7,quitv
44DA 4E78 002A    if r7 <>=%abrt0+words
44DE 9102
44EO 4CC1    ior r4,=1
44E2 7648 OOAA    if r4 <> procno
44E6 9106
44E8 3048 OOAA    sta r4,procno
44EC E01A    Trap 32,<:Stge CD:Bad Proc Id (H/S) - page 84>
44EE 4048 OA1C    jsb r4,wst
44F2 48F1
44F4 A274    sll r7,r4
44F6 3478 01C0    iorm r7,proc0d
44FA 4DC1    eor r4,=1
44FC 48F1    lda r7,=1
44FE A274    sll r7,r4
4500 4BC1    and r4,=1
4502 A634    sr1 r3,r4
4504 9907    ifnot odd
4506 A2C5    sll r4,5      ;;%cpu1-%cpu0
4508 774C FF1E    tst r4,%cpu0+%ctrl(r4)
450C 8003    ifnot quit
450E 3478 01C0    iorm r7,proc0d
4512 6076    endif
4514 4FF0    fail return
4516 4007

endroutine scdset
```

.comnt |
Block Transfer Subroutine

Monitors state of the block transfer control variables. If any transferring needs to be done, do it. Will do physical or logical memory transfers. Polls application-supplied external reload routines as well. Used by later stages when any memory is found to be lacking, or for processor-reloading and restarting, and external reloading and restarting. Application programs (e. g. DDT, IMP dump/reload) may also use it as well. Called from stage once we get to stage RC. Use rckcon for a consensus to drive us.

.comnt |
Block transfer to one's buddy. Only called if buddy isn't in the RCKCON consensus. Assume buddy is halted, and step him to do the deposits or examines.

```
4518 3478 0180 b1tbud: iorw r7,temp2 ;remember buddy's bit
451C 3008 0182 sta r0,temp3 ;can't check buddy for spares
4520 4078 4A14 jsb r7,b1tprs ;do proc stuff
4524 8703 tf3 bltc3 ;I'll read his regs)
4526 4008 46B0
452A 483A FF00 1da r3,%cpu0(r2)      ;point to buddy's regs
452E 4891 1da r1,%pchalt
4530 301B 001E sta r1,%ctrl1(r3)    ;halt the buddy proc
4534 4818 F800 1da r1,%a+%11+%12+%13+%14 ;inhibited and active
4538 301B 0010 sta r1,%stat(r3)     ;into his status
453C 802C qutpat b1tbcq ;he didn't stop properly
453E 487E FFFC 1da r7,=-2+words(r6) ;start PC just before xfer
4542 3073 sta r7,(r3)    ;;%reg0 ; so dumping uses inst fetch
4544 A6A5 sr1 r2,5      ;get back odd bit
4546 4C28 0810 ior r2,=key 0 ;instruction to properly key buddy
454A 4893 1da r1,%pcstep ;constant to step proc
454C 302B 0012 sta r2,%inst(r3)   ;key inst to inst reg
4550 301B 001E sta r1,%ctrl1(r3) ;now execute the key inst
4554 963D bf2 b1tbus ;what we should do is start him
4556 4878 8000 1da r7,=nop    ;inst if reading memory
455A 8503 bnf1 .+6      ;that's what we want
455C 4878 2012 1da r7,=sta r1,(r2)+ ;writing memory
4560 306B 0004 sta r6,%reg2(r3) ;location to examine/deposit
4564 7024 b1tbu1: 1da r2,(r4)    ;next word from b1t buffer .
4566 302B 0002 sta r2,%reg1(r3) ;into his r1
456A 307B 0012 sta r7,%inst(r3) ;setup his inst
456E 301B 001E sta r1,%ctrl1(r3) ;step him
4572 702C 0000 1da r2,0(r4) ;long, long nop
4576 7663 cmp r6,(r3)  ;;%reg0 ;step okay?
4578 8123 bne b1tbmq ;hope, give up with error
457A 4AE2 add r6,=2    ;expected next pc
457C 702B 0012 1da r2,%inst(r3) ;contents he fetched
4580 2024 sta r2,(r4)+ ;into buffer
4582 49D2 sub r5,=words ;number of bytes left
4584 8AFO bnz b1tbu1 ;more yet
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 86
 STAGEK.PLR;1 PAGE 71

```

4586 9503    b1tbuc: tnf1 b1tmsk   ;filling buffer: toggle bit
4588 4008 46FE
458C 7038 0180 b1tbgo: lda r3,temp2 ;restore bit(s) of proc(s) we did
4590 4008 47AC      tr b1tbit  ;emptying: remove proc bits

4594 4818 0020 b1tbcq: lda r1,=b1tedk ;proc access error
4598 3158 5DC4 b1tbue: subm r5,b1tdon
459C 9502          bf1 .+4   ;storing core
459E A298          s11 r1, H8   ;shift into position
45A0 3418 5DBC     iorm r1,b1tst  ;set error bits
45A4 7038 0180     lda r3,temp2  ;procs we tried
45A8 3438 5DCC     iorm r3,b1tbmk  ;remember bad guys
45AC 4D38 FFFF     eor r3,=-1   ;were getting core
45B0 850A          bnf1 b1tbug  ;remove procs from dest
45B2 3338 5DDE     andm r3,b1tdpi
45B6 8AEB          bnz b1tgo   ;others are all right
45B8 4891          b1tbut: lda r1,=b1tact ;give up altogether
45BA 4008 4708      tr b1teor

45BE 4818 0010 b1tbmq: lda r1,=b1tedq ;dest quit bit
45C2 90EB          br b1tbue

45C4 3338 5DD8 b1tbug: andm r3,b1tspm ;forget this source proc
45C8 9AF8          bz b1tbut  ;no more to try
45CA 4008 470C      tr b1tbpr  ;flip bits

45CE 9503    b1tbus: tnf1 b1ttag  ;filling, just toggle
45DO 4008 4706
45D4 7818 5DBC     l1dab r1,b1tst  ;state
45D8 4F97          tst r1,=bitrun}bitctl_- H8      ;;pcrun}pchalt ;start proc?
45DA 9AD6          bz b1tbuc  ;none, no start
45DC 4B93          and r1,=bitctl_- H8
45DE 8A0D          bnz b1tbuz
45E0 4818 09E6     lda r1,=wstini  ;nominal start
45E4 3013          sta r1,(r3)   ;;%reg0 ;new PC
45E6 7018 5DCA     lda r1,b1tpro
45EA 301B 0006     sta r1,%reg3(r3) ;his procno
45EE 4818 F800     lda r1,=%a}%11)%12)%13)%14 ;start status
45F2 301B 0010     sta r1,%stat(r3) ;into reg 8
45F6 4892          lda r1,=pcrun  ;start buddy
45F8 301B 001E      ; Trap 31,<Buddy Procs strtd (Call MAINT if continuous) - page 86>
45FC 90C5          b1tbuz: sta r1,%ctrl(r3)
45FD 90C5          br b1tbuc

```

.comnt |
Main entry to Block Transfer Subroutine

Sets up registers, and dispatches depending on type of memory being transferred. If either type is external (sign bit true) then poll the external reload routine.

```
|  
45FE 3078 017E b1t:    sta r7,temp1      ;put away the return  
4602 7018 BDBA      lda r1,b1tlok      ;interlock b1t  
4606 9AFE          bz .-4  
4608 4838 FFFF      lda r3,=-1        ;haven't picked a proc yet  
460C 3038 0180      sta r3,temp2      ;  
4610 0270          rst %f1+%f2+%f3 ;init states  
4612 7018 5DBC      lda r1,b1tst      ;what's doing  
4616 9903          tno b1tx         ;;b1tact      ;nothin'  
4618 4008 4720  
461C 4838 5DD6      lda r3,=b1tsty     ;point to source params  
4620 4F92          tst r1,=b1tful     ;full b1t buffer?  
4622 9A04          bz b1t02  
4624 0290          sst %f1          ;signal to empty buffer  
4626 4838 5DDA      lda r3,=b1tdty     ;destination parameters  
462A 7048 5DC6 b1t02: lda r4,b1tbmf     ;map for b1t buffer  
462E 3048 FCO4      sta r4,%map2     ;is through map 2  
4632 7048 5DC8      lda r4,b1tbfa     ;its address there  
4636 7058 5DC2      lda r5,b1tsiz     ;how much to do  
463A 8A02          bnz .+4          ;non-zero  
463C 02AO          sst %f2          ;at end- maybe start up  
463E 4E58 0040      cmp r5,=b1tmax     ;more than maximum  
4642 9203          bg .+6          ;nope  
4644 4858 0040      lda r5,=b1tmax     ;just do that much  
4648 3058 5DC4      sta r5,b1tdon     ;length we might transfer  
464C 6023          lda r2,(r3)+     ;;b1tsty b1tdty :type  
464E 4F28 E000      tst r2,=b1tfak}b1tdt}b1tfex   ;non-b1t use?  
4652 8A06          bnz b1tfgo       ;yes, fake it  
4654 7068 5DC0      lda r6,b1tadd     ;address  
4658 996C          bo b1tcom       ;;b1tnlc ;common memory transfer  
465A 4008 484E      jmp b1tpch       ;must be a processor then
```

;Fake half of a non-BLT transfer :

```
465E 4818 0040 b1tfgo: lda r1,=b1tmax     ;how much transferred  
4662 3218 5DC8      addm r1,b1tbfa     ;advance buffer pointer  
4666 854C          bnf1 b1tmsk      ;set up procmsk and flip ful bit  
4668 4008 47B6      tr b1tfdn      ;advance pointers, etc.
```

;check b1t timeout

```
466C 7038 5DBE b1ttch: lda r3,b1tto      ;how much time gone  
4670 7138 00AO      sub r3,stime     ;less present  
4674 4F38 F000      tst r3,= HF000    ;reasonable?  
4678 9A54          bz b1tx         ;yes  
467A EO1B          Trap 33,<;Blk Trans Timeout (proc trb1) - page 87>  
467C 4895          lda r1,=b1teto}b1tact  
467E 9045          br b1teor       ;stop b1t
```

.comnt |
Block transfer to or from common memory, or my own local
Enter at bltcom
|

```
4680 2026      bltceq: sta r2,(r6)+    ;put into memory
4682 809E          qutpat b1tbmq    ;no memory
4684 4008 47A4      tr bltce1    ;but proceed

4688 A2A9      bltc1: s11 r2, H9    ;physical page
468A 3028 0182      sta r2,temp3    ;remember type for spare check
468E 9611      bltc4: bf2 b1tc3    ;no checking, transfer finished
4690 4C68 6000      ior r6.=m1    ;make map 1 address
4694 3028 FCO2      sta r2,%map1    ;get target
4698 9993          bo b1tbmq    ;not a legit map}
469A 4078 4A10      jsb r7,b1tprm
469E 4878 40F6      lda r7,=ccktab    ;cksum table for common
46A2 4838 OACE      lda r3,=stgpas
46A6 7638 60C4      cmp r3,m1/rkepas    ;see rely?
46AA 8003          qutpat b1tc3    ;no - bad parity or no memory
46AC 8102          bne b1tc3    ;no see rely
46AE 49F2          sub r7,=ccktab-rcktab    ;do see - rely checksums too
46B0 955D      b1tc3: bf1 b1tcc    ;empty the buffer
46B2 9626          bf2 b1tmsk    ;done transfer
46B4 6036      b1tcf: lda r3,(r6)+    ;get a word
46B6 8084          qutpat b1tbmq    ;oops
46B8 2034          sta r3,(r4)+    ;into buffer
46BA 49D2          sub r5,=words
46BC 8AF0          bnz b1tcf    ;more to fill
46BE 4F18 0080      if r1.bit. =b1tsdf    ;checking spares
46C2 9A1E
46C4 7028 0182      lda r2,temp3    ;saved type
46C8 9B1B          ifnot minus    ;not big physical page
46CA 7628 4104      if r2 < ncodep    ;was code
46CE 8218
46DO 7228 4104      add r2,ncodep
46D4 703A 00B0      lda r3,1map(r2)    ;get map
46D8 9913          ifnot odd    ;page exists
46DA 3038 FCO2      sta r3,%map1
46DE 7058 5DC4      lda r5,b1tdon    ;how much to check
46E2 4965          sub r6,r5    ;back up pointers
46E4 4945          sub r4,r5
repeat
46E6 6036          lda r3,(r6)+    ;next spare word
46E8 8005          until quit
46EA 6634          until r3 <> (r4)+    ;check in buffer
46EC 8103
46EE 49D2          sub r5,=words
46FO 8AFB          until zero
endrepeat
46F2 4ED0          if r5 <> =0 ;got quit or mismatch
46F4 9105
46F6 4D18 0080      eor r1,b1tsdf    ;stop checking then
46FA 3018 5DBC      sta r1,b1tst    ;and report difference
endif
endif
```

endif
endif

if (abs(1.0 - $\sqrt{1.0 - \frac{1}{n} \sum_{i=1}^n x_i^2}$) < 1e-10) {
 cout << "The vector is orthogonal." << endl;
}
else {
 cout << "The vector is not orthogonal." << endl;
}
}

C

C

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 89
STAGEK.PLR;1 PAGE 74

```
46FE 7068 5DDE b1tmsk: lda r6,b1tdpi ;init dest proc mask
4702 3068 5DDC sta r6,b1tdpm
4706 4892 b1ttog: lda r1,=b1tful ;change full state
4708 3518 5DBC b1teor: eorm r1,b1tst
470C 7038 00A6 b1tbpr: lda r3,myproc
4710 3038 5DCA sta r3,b1tpro ;I did it
4714 7038 00AO lda r3,stime ;time now
4718 4A38 0078 add r3,-b1trat ;how long before timeout
471C 3038 5DBE sta r3,b1tto ;update timeout
4720 3008 BDBA b1tx: sta r0,b1tlok ;unlock b1t
4724 4868 02C8 lda sp,=s1stack
4728 4078 4A46 call polrid ;check if reloading
472C 4088 017E jmp @temp1 ;done the subroutine

;BLT to/from common memory

4730 4B68 9FFE b1tcom: and r6,=<b1tlog>packm>&<-1?b1tnlc> ;relevant bits
4734 8BAA bnm b1tc1 ;b1tlog :physical page
4736 4F18 0800 b1tc2: tst r1,=b1tspa ;want spare?
473A 9AOB bz b1tc6 ;no
473C 7628 4104 cmp r2,ncodep ;A real code page?
4740 8208 bng b1tc6 ;nope
4742 7228 4104 add r2,ncodep
4746 703A 00BO lda r3,1map(r2) ;spare exist?
474A 8903 bno b1tc6 ;yes
474C 7128 4104 sub r2,ncodep ;transform to main type
4750 3028 0182 b1tc6: sta r2,temp3
4754 702A 00BO lda r2,1map(r2) ;logical to physical
4758 4B68 1FFE b1tc7: and r6,=packm ;will use map 1
475C 9099 br b1tc4

475E A698 b1tmys: srl r1, H8
4760 4B93 and r1,=b1tct1_- H8 ;;pchalt>pcrun
4762 9A23 bz b1tmyb ;no mess with R15
4764 3018 0160 sta r1,b1tmyr+%ctrl
4768 9020 br b1tmyb
```

:check here for checksums to adjust

476A 96FA b1tcc: bf2 b1tmys ;transfer complete
476C 4F98 tst r1.=b1tcck ;really checksumming?
476E 9AOE bz b1tce ;no, not really
4770 6037 b1tcc1: lda r3,(r7)+ ;next checksum range
4772 9AOC bz b1tce ;end of table
4774 4E63 cmp r6,r3 ;addr in range?
4776 92FD bg b1tcc1 ;addr < low limit
4778 4826 lda r2,r6
477A 7123 sub r2,(r3) ;high limit
477C 80FA qutpat b1tcc1 ;oops, bad parity
477E 8BF9 bnm b1tcc1 ;addr GE hi limit
4780 5073 lda r7,(-r3) ;cksum parity okay?
4782 8004 qutpat b1tce ;nope
4784 4E78 FEED cmp r7,=ckpass ;don't modify if password
4788 8102 bne b1tce2
478A 48B2 b1tce: lda r3,=bitsnk ;default if no checksum to update
478C 6024 b1tce2: lda r2,(r4)+ ;next in buffer
478E 7126 sub r2,(r6) ;change
4790 8002 qutpat b1tceq ;oops, see if parity
4792 9003
4794 4008 4680 subm r2,(r3) ;diddle checksum
4798 3123 addm r2,(r6)+ ;and fix core
479A 2226 qutpat b1tbmq ;must be a hard QUIT if any
479C 8002
479E 9003
47A0 4008 45BE
47A4 49D2 b1tce1: sub r5,=words
47A6 8AF3 bnz b1tce2 ;more in buffer
47A8 7038 0180 b1tmyb: lda r3,temp2 ;my bit
47AC 4D38 FFFF b1tbit: eor r3,=-1
47B0 3338 5DDC andm r3,b1tdpm ;one proc done
47B4 8AAC bnz b1tbpr ;but others aren't
47B6 7068 5DC4 b1tfdn: lda r6,b1tdon ;how many bytes done
47BA 9AO6 bz b1tend ;just finished: maybe startup
47BC 3168 5DC2 subm r6,bitsiz ;total bytes remaining
47CO 3268 5DC0 addm r6,b1tadd ;update transfer address
47C4 90A1 br b1ttog ;flip the bit

47C6 7038 00AO b1tend: lda r3,stime ;set up next time
47CA 4A38 0078 add r3,=prrate ;for processor start-up
47CE 3038 5EA0 sta r3,pftime ;(give new ones a chance)
47D2 7818 BDD1 l1dab r1,m3/b1tpok+1
47D6 3098 00AC sta r1,@pid
47DA 4893 lda r1.=b1tact}b1tful ;completed transfer
47DC 9096 br b1teor

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 91
 STAGEK.PLR;1 PAGE 76

.comnt |
 Transfer to/from processor space. Simulate maps in memory (bltmym)
 |

```

47DE 3478 0180 b1tme: iorm r7,temp2 ;bits for procs we did
47E2 4078 4A10 b1tme1: jsb r7,b1tprm ;check if regs
47E6 8703          bnf3 .+6 ;not regs
47E8 496A FDBE          sub r6,%cpu0-b1tmyr(r2) ;point at simulated regs
47EC 4878 40EE          lda r7,=lcktab ;default checksum list
47F0 3078 0182          sta r7,temp3 ;never spares
47F4 4866          lda r6,r6 ;what address?
47F6 9B11          bm b1tm1 ;may be I/O
47F8 4E68 4000          cmp r6,=m0 ;maybe map 0 or 1
47FC 8203          tg b1tc3 ;local memory address
47FE 4008 46B0
4802 4826 b1tm3: lda r2,r6 ;simulate maps here
4804 A3A4          r11 r2,4 ;high to low order
4806 4BAE          and r2,= HE ;packm ; to select map
4808 702A 015E          lda r2,b1tmym-<m0_- D12>(r2) ;proper map
480C 4F28 FF01          tst r2,= HFF01
4810 9A93          bz b1tc2 ;logical page
4812 4B28 FFFE b1tm2: and r2,=-2 ;use physical page
4816 90A1          br b1tc7

4818 4E68 C000 b1tm1: cmp r6,=m3+<m3-m2>
481C 92F3          bg b1tm3 ;common memory address
481E 4E68 FC00          cmp r6,%map0 ;changing map reg?
4822 92EE          tg b1tc3 ;nope
4824 4E68 FC06          cmp r6,%map3&-2
4828 9CEB          t1 b1tc3 ;nope
482A 4968 FA9E          sub r6,%map0-b1tmym ;use virtual maps
482E 90E8          tr b1tc3

```

.comnt |
 Blt to a specific proc. Check for me, then set up mask
 and go into proc mask checking code
 |

```

4830 7628 00A6 b1t10: cmp r2,myproc ;me the proc?
4834 91D7          be b1tme1 ;sure 'nuff
4836 48B1          lda r3,=1
4838 4890          lda r1,=0
483A 6E29 5D6A b1t11: cmpb r2,coutab(r1)+ ;this one in the table
483E 911F          be b1t12 ;found the one
4840 A2B1          s11 r3,1 ;next proc's bit
4842 8AFC          bnz b1t11 ;more to try
4844 E01C          Trap 34,<;BLT Target proc not intbl - page 91>
4846 4818 0040 b1tfer: lda r1,=b1tedf ;format error
484A 4008 4598          tr b1tue

```

```
484E 6033      b1tpch: lda r3,(r3)+    ;:b1tdpm b1tspm ;proc mask
4850 4EAF        cmp r2,=anypro   ;processor set?
4852 81EF        tne b1t10    ;nope, one only
4854 3038 0180   sta r3,temp2  ;in case format error
4858 7338 5D64   and r3,procex ;what procs there are
485C 9AF5        bz bltfer   ;none??
485E 9607        bf2 b1tpcs  ;at end, do everyone
4860 487E 0400   lda r7,=%map0(r6) ;local proc address?
4864 9B07        bm b1tpco   ;yes, so leave all bits in mask
4866 4E78 4400   cmp r7,=m0-%map0
486A 8204        bng b1tpco  ;yes
486C 4F00        b1tpcs: lda r7,=0    ;we'll just do one here
486E 3078 0180   sta r7,temp2  ;haven't picked one yet
4872 7078 00A8   b1tpco: lda r7,procbt
4876 4F37        tst r3,r7   ;me in mask?
4878 8AB3        bnz b1tme   ;yes, I must do it
487A 9507        bf1 b1t03   ;emptying buffer
487C 7738 5E28   b1t12:  tst r3,rckcon ;any in consensus
4880 9A03        tnz blttch  ;yes, let 'em do it
4882 4008 466C
4886 9007        br b1t04

4888 7018 5E28   b1t03:  lda r1,rckcon ;consensus
488C 4D18 FFFF   eor r1,=-1   ;procs not in consensus
4890 4B31        and r3,r1   ;I might do one of these
4892 9AF8        tz blttch  ;none to do
4894 7028 00AA   b1t04:  lda r2,procno ;my number
4898 4DA1        eor r2,=1   ;my buddy
489A 4F1         lda r7,=1   ;his bit
489C A272        s11 r7,r2   ;he to be done?
489E 4F37        tst r3,r7   ;yes, I'll do him
48A0 9A03        tnz bltbud
48A2 4008 4518
48A6 7018 5E28   lda r1,rckcon ;consensus
48AA 4871        lda r7,r1   ;even guys' bits
48AC 4B18 5555   and r1,= H5555 ;odds
48BO 4D71        eor r7,r1
48B2 A6F1        srl r7,1
48B4 A291        s11 r1,1
48B6 4C17        ior r1,r7   ;buddies of running procs
48B8 9504        bf1 b1t05   ;emptying buffer
48BA 4F31        tst r3,r1   ;any with live buddy?
48BC 9A06        bz b1t06   ;no, I'll do it by hbc
48BE 90E2        tr blttch  ;yes, let that buddy do it

48C0 4D18 FFFF   b1t05:  eor r1,=-1   ;not buddies of live guys
48C4 4B31        and r3,r1   ; anyone need help?
48C6 9ADE        tz blttch  ;none left
48C8 48A0        b1t06:  lda r2,=0   ;find this coupler
48CA 4F1         lda r7,=1
48CC 4F37        b1t07:  tst r3,r7
48CE 8A06        bnz bltbdc  ;found one to do
48DO 4AA1        add r2,=1
48D2 A2F1        s11 r7,1   ;next coupler and bit
48D4 8AFc        bnz b1t07  ;check next bit
48D6 E01D        Trap 35,<:Non-existant proc in b1t?? (S) - page 92>
48D8 8           br bltfer   ;Bad b1t format
```



.comnt |
Block transfer via Backwards Bus Coupling (BBC)

Enter with r2/coupler number, r4/ buffer address, r5/words to transfer
Assume the bus is amputated, since its pros not in system yet. Later
stages would fix that anyway.

|

```
48DA 3478 0180 b1tbbc: iorm r7,temp2 ;remember proc bit for later
48DE 703A 5D9A    lda r3,amproc(r2)      ;proper amputate word
48E2 3038 0184    sta r3,temp4       ;remember for later
48E6 783A 5D6A    ldab r3,coutab(r2)   ;processor coupler
48EA 4078 4A14    jsb r7,b1tptrs    ;fix proc stuff
48EE 4894    lda r1,=bsadil    ;for all I/O buses
48F0 7028 5D5A b1tbb2: lda r2,usebus  ;buses in system
48F4 5729 0EB6 b1tbb0: tst r2,bittab(-r1) ;bus exist?
48F8 8A04    bnz b1tbb1    ;yes
48FA 88FD    bnlp b1tbb0    ;look more
48FC E01E    Trap 36,<;No I/O bus for BBC (H) - page 93>
48FE 90A4    br b1tferr  ;format error exit

4900 3018 0182 b1tbb1: sta r1,temp3 ;bus we tried
4904 7079 0320    lda r7,bsadrs(r1)  ;address of the bus
4908 4828 FF18    lda r2,=<%cpu0+%ctrl1>&bbcmask ;assume even proc
490C 4A37    add r3,r7      ;get coupler address
490E 8903    bno .+6 ;its an even proc
4910 4828 FF3A    lda r2,=<%cpu1+%ctrl1>&bbcmask>+bbcodd ;odd proc
4914 7018 BD68    lda r1,bbclok  ;lock for BBC
4918 9AFE    bz .-4
491A 7018 0184    lda r1,temp4  ;proper amputates, password
491E 4C94    ior r1,=bbcbak
4920 3013    sta r1,(r3)    ;enable for BBC
4922 302F 008E    sta r2,bbcmmap(r7) ;set map for control reg
4926 7818 5DBC    ldab r1,b1tst
492A 4F97    tst r1,=b1tct1}b1trun_- H8
492C 9A05    bz b1tbb9    ;don't need to halt him
492E 4891    lda r1,=pcchalt ;first halt the proc
4930 301F 0086    sta r1,<%ctrl1&bbcwmk>+bbcwin(r7)
4934 8053    qutpat b1tbbk ;no proc here
4936 9626 b1tbb9: bf2 b1tbb5 ;really want to start him
4938 4BA2    and r2,=bbcodd ;need this bit
493A 4816    lda r1,r6
493C 4B18 FFF8    and r1,=bbcmask ;map bits
4940 4D61    eor r6,r1    ;window index
4942 4A67    add r6,r7    ;bus base for BBC window
4944 4C12    ior r1,r2    ;get the key bit
4946 301F 008E b1tbb3: sta r1,bbcmmap(r7) ;point to memory
494A 850A b1tbb4: bnf1 b1tbb5 ;filling the buffer
494C 6024    lda r2,(r4)+ ;next buffer word
494E 202E 0080    sta r2,bbcwin(r6)+ ;BBC store
4952 8047    qutpat b1tbbq ;no memory there
4954 762E 007E    cmp r2,bbcwin-2(r6) ;store okay?
4958 8044    qutpat b1tbbq
495A 8143    bne b1tbbq ;problems
495C 9005    br b1tbb6 ;proceed
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 94
STAGEK.PLR;1 PAGE 79

```

495E 602E 0080 b1tbb5: lda r2,bbcwin(r6)+ ;next word from BBC
4962 8040         qutpat b1tbb7 ;no memory
4964 2024         sta r2,(r4)+ ;into buffer
4966 49D2         b1tbb6: sub r5,=words ;done a word
4968 9A06         bz b1tbb7 ;transfer complete
496A 4FE6         tst r6,=bbcwmk ;through window?
496C 8AEF         bnz b1tbb4 ;not yet
496E 4A98         add r1,=bbcwmk+words ;next map setting
4970 49E8         sub r6,=bbcwmk+words ;back to window start
4972 90EA         br b1tbb3

4974 7028 0184 b1tbb7: lda r2,temp4 ;undo BBC enable
4978 3023         sta r2,(r3)
497A 3008 BD68     sta r0,bbclok ;done with BBC
497E 4008 4586     tr bltbuc ;fix the bits

4982 85F9         b1tbb8: bnf1 b1tbb7 ;filling: ignore
4984 7818 5DBC     l1dab r1,b1tst ;state
4988 4F97         tst r1,=b1trun>b1tctl_- H8 ;;pchalt pcrun ;starting?
498A 9AF5         bz b1tbb7 ;no - go away
498C 4B93         and r1,=b1tctl_- H8
498E 8A23         bnz b1tbbz
4990 49A8         sub r2,=<%ctr1-%inst>&bbcmask ;get to IR
4992 302F 008E     sta r2,bbcmmap(r7)
4996 4812         lda r1,r2
4998 4B92         and r1,=bbccodd ;just odd bit
499A A691         srl r1,1 ;to low order
499C 4C18 0810     ior r1,=key 0 ;proper key inst
49A0 301F 0082     sta r1,bbcwin+<%inst&bbcwmk>(r7) ;to IR
49A4 8020         qutpat b1tbbt
0000 .if nz <%inst-%stat>&bbcmask
                 sub r2,=<%inst-%stat>&bbcmask
                 sta r2,bbcmmap(r7) ;get to status
.endc
49A6 4818 F800     lda r1,=%a%L1%L2%L3%L4 ;new value
49AA 301F 0080     sta r1,bbcwin+<%stat&bbcwmk>(r7)
49AE 4928 0010     sub r2,=<%stat-%reg0>&bbcmask ;get to PC
49B2 302F 008E     sta r2,bbcmmap(r7)
49B6 4818 09E4     lda r1,wstini-words ;nominal start address
49BA 301F 0080     sta r1,bbcwin+<%reg0&bbcwmk>(r7) ;init PC
49BE 7018 5DCA     lda r1,b1tpro ;his procno
49C2 301F 0086     sta r1,bbcwin+<%reg3&bbcwmk>(r7) ;to r3
49C6 4A28 0018     add r2,=<%ctr1-%reg0>&bbcmask
49CA 302F 008E     sta r2,bbcmmap(r7) ;get to control reg
                 ; Trap 30,<Rstrtd proc via BBC - page 94>
49CE 3038 5DD2     sta r3,bbcrst ;remember who we did
49D2 4892         lda r1,=pcrun ;to start the processor
49D4 301F 0086     b1tbbz: sta r1,bbcwin+<%ctr1&bbcwmk>(r7)
49D8 90CE         br b1tbb7 ;starting: allow proper access

```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 95
STAGEK.PLR;1 PAGE 80

```
49DA 4878 4594 b1tbbk: lda r7,=b1tbcq ;routine to go to
49DE 9008 br b1tbbq ;see if another bus to try

49EO 49C2 b1tbbq: sub r4,=2 ;backing up the pointers
49E2 49E2 b1tbbq: sub r6,=2
49E4 E020 b1tbbt: Trap 40,<;BBC fail (CALL MAINT if 1E TRAP occurs) (H) - page 95>
ge 95
49E6 3038 5DD4 sta r3,bbcbad ;remember who failed
49EA 4878 45BE lda r7,=b1tbmq
49EE 7028 0184 b1tbbq: lda r2,temp4 ;fix BBC
49F2 3023 sta r2,(r3)
49F4 3008 BD68 sta r0,bbclok ;done BBC
49F8 4B38 007F and r3,=bbcwin-1 ;get back proc num
49FC 4BE6 and r6,=bbcwmk ;reconstruct where we are
49FE 4B18 FFF8 and r1,=bbcmesk
4A02 4A61 add r6,r1
4A04 7018 0182 lda r1,temp3
4A08 9A03 tnz b1tbb2 ;other bus to try
4AOA 4008 48FO
4AOE 4007 jmp (r7) ;error: set bits

;subroutine for b1tbbd, b1tbbc
;return r2/ 0 (cpu0) or 20 (cpu1)
;call r7/ return, r2/ procno of dest, r6/ b1t address

4A10 7028 00AA b1tprm: lda r2,procno ;entry for own processor
4A14 3028 5DCA b1tprs: sta r2,b1tpro ;temp, his procno
4A18 782A 5D6A l1dab r2,coutab(r2) ;who we're doing
4A1C 9603 ifnot f2 ;if not at end
4A1E 3028 5DCE sta r2,b1tdid ;remember for DDT
endif
4A22 4BA1 and r2,=1 ;odd bit
4A24 A2A5 s11 r2,5 ;;%cpu1-%cpu0
4A26 4E68 FF1E cmp r6,%cpu0+%ctrl ;proc reg?
4A2A 9C06 b1 b1tprx ;nope
4A2C 4E68 FFOO cmp r6,%cpu0
4A30 9203 bg b1tprx ;nope
4A32 4A62 add r6,r2 ;point right if odd
4A34 02C0 sst %F3 ;we need regs
4A36 4007 b1tprx: jmp (r7)

;now insert user reloader if any

rldins
.INSRT IMP.PKCORE
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 96
 IMP.PKCORE;1 PAGE 1

.comnt |
 ARPA Network reload subroutines

Implements packet core reloading of a dead PLURIBUS over a modem line.
 PLURIBUS requests core types that it needs by calls on TRYRLD at appropriate places in STAGE code. Block transfer polls POLRLD in order to support external reloading. PKCIC and PKCDC are callable from fake hosts for core transfers between live machines.

|
 Page Dummy

;core transfer parameters

```
00F0      pkcrat=6* D40    ;retry blank setup-send every 6 seconds
00C8      pkcsto=5* D40    ;retry setup-send if no input for 5 seconds
0640      pkclto= D40* D40    ;once active, be more lenient
          .if df PSE
0002      pkctry=2        ;# of tries per device.
          .iff   ;df PSE
          pkctry=6        ;# of tries per device.
          .endc   ;df PSE
000F      pkctmx= D15      ;lock onto a device for this long
FE00      cpmask= HFE00
B400      csetup= 0132000 :password for a setup
B600      cpcore=csetup+ 01000   :and for core
B800      cabort=cpcore+ 01000   :and aborts
0004      nfh=4
0048      rbflen= D72       ;words in reload modem buffers
```

;Pluribus modem interface

```
0000      .=0
0000      statd: .b1kw 1
          modid= H100
          hmodid= H500
          magmod= H8000
0002      starti: .b1kw 1
0004      endi: .b1kw 1
          0002      hrloff=words ;hardware stops a word short
          8000      mendf= H8000
0006      statim: .b1kw 1
          000E      lowmsk= HE      ;which bits of address go into status
          2000      mbusy= H2000
0008      starto: .b1kw 1
000A      endo: .b1kw 1
000C      statom: .b1kw 1
```

;words in a reload packet header

0000 .=0
0000 neth: .b1kb 1 ;network header
0001 cpsize: .b1kb 1 ;core piece size
0002 typh: .b1kw 1 ;special flags if core
C000 paktyp= HC000
C000 rldtyp= HC000
0004 chkh: .b1kw 1 ;checksum
0006 cpsats: .b1kw 1 ;:srch :satellite source
0008 cpsatd: .b1kw 1 ;:seqh ;and dest
000A pkth: .b1kw 1
000B cpdh=pkth+1 ;core dest Host num (byte)
000C dsth: .b1kw 1 ;core dest imp
000E cpd1: .b1kw 1 ;:midh ;core dest 1ink

0010 cmark: .b1kw 1 ;:data ;mark to distinguish core, setup
0100 cmmbit= H100 ;bit to say for magic modem (no special processing)
0080 crpbit= H80 ;repeat bit
0100 crpcnt= H100 ;how many times to repeat(32 seconds)
0007 cpline=7 ;which line to send to
0012 cmmbeg: .b1kw 0 ;where to copy magic modem data down

;data in a setup

0012 cforha: .b1kw 1 ;Foreign handling type
0013 cforhs=cforha+1 ;Foreign Host (byte addr)
0014 cfori: .b1kw 1 ;Foreign IMP number
0016 cformi: .b1kw 1 ;Foreign message-id
FFF0 messid= HFFF0
0018 ctype: .b1kw 1 ;core type
0019 cforl=ctype+1 ;and foreign line
001A cpkadd: .b1kw 1 ;start address
001C cpksiz: .b1kw 1 ;transfer size
001E cpkssf: .b1kw 1 ;send setup flag
0020 csrflg: .b1kw 1 ;send/receive flag (end of setup)

;data in core piece

0012 .=cmark+2
0012 ccaddr: .b1kw 1 ;this piece address (0=last piece)
0014 cctype: .b1kb 1 ;type and size
0015 FEO0 ccsze: .b1kb 1 ;size of this piece
00FF pkctmk= HFEO0 ;mask for type bits
00FF pkclmk= HFF ;mask for length field
0016 ccpiec: .b1kw 1 ;beginning of the piece

0018 .b1kw D59 ;words of data, plus ending 0

008E bufend: .b1kw 0 ;how much in a legit message on modem

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 98
 IMP.PKCORE;1 PAGE 3

```
;Reload variables

page relvars

5EAC      rlddev: .blkw 1 ;which device to reload from
          locdef pkclok.<;lock on packet core parameters - page 98>
5EBO      pkcst: .blkw 1 ;packet core state word
          7074      blters=blters ;bits reported with setups for errors
          8000      pkcist=sign ;set -> input buffer in progress
          0200      pkcost= H200 ;set -> output buffer being filled
          0100      pkcext= H100 ;set -> external reload using packet core
          0080      bitsdf=bitsdf ;set -> check spare for differences
          0008      bltcck=bltcck ;set whenever we get setup - fix checksums
          0002      pkcety=2 ;set -> received illegal packet core type
          0001      pkcact=1 ;set whenever pkt core is active
5EB2      pkctim: .blkw 1 ;packet core timeout
5EB4      pkcotm: .blkw 1 ;timer to slow pkt core output
5EB6      pkcrct: .blkb 1 ;retry counter for device
          .blkb 1 ;spare
5EB7
5EB8      pkcdon: .blkw 1 ;pkt core address after completing this piece
5EBA      pkcbfa: .blkw 1 ;present address in buffer

5EBC      pkccla: .blkw 0 ;begin zeroing here

5EBC      psetup: .blkw 1 ;saved copy of latest setup
5EBE      pkctyh: .blkw 1 ;=rldtyp in packet core
          .blkw 1 ;checksum
5EC0      pkcmyi: .blkw 1 ;source imp on dead line
5EC2      pkcnim: .blkw 1 ;neighbor IMP on dead line
5EC4      pkclha: .blkb 1 ;handling type in leader
          pkclho: .blkb 1 ;host number in leader
5EC7      pkclim: .blkw 1 ;IMP number in leader
5ECA      pkclid: .blkw 1 ;message-id in leader
5ECC      psdata: .blkb 1 ;password (setup or core),, line
          pkclln: .blkb 1 ;(dest line)
5ECD      pkcfha: .blkb 1 ;foreign handling type and host
          pkcfho: .blkb 1 ;byte address
5ECF      pkcfim: .blkw 1 ;foreign IMP in setup
5ED0      pkcfid: .blkw 1 ;foreign mess-id
5ED2      pkctyp: .blkb 1 ;setup: type*2+MM,,line number
          0040      plrpkc= H40 ;(old) Pluribus core type convention
          0040      plrcom= H40 ;(new) Pluribus common memory type convention
5ED5      pkcfln: .blkb 1 ;foreign line
5ED6      pkcadd: .blkw 1 ;core address
5ED8      pkclen: .blkw 1 ;and length
          0039      pkcmax= D57 ;max to send per message
5EDA      pkcssf: .blkw 1 ;send setup flag
5EDC      pkcsrf: .blkw 1 ;setup: send/receive flag
          0022      psetul=-psetup ;how much in a setup

5EDE      rldinb: .blkw rbflen ;reload input buffer
5F6E      rldotb: .blkw rbflen ;and output buffer
          0142      pkccll=-pkccla ;how much to zero at init
```

pluribus IMP Loader
IMP.PKCORE;1 PAGE 4

PLURIBUS V2.9B 25-Jun-87 11:20:52

PAGE 99

;First, include global device-input code

page LCode

.insert "XSIOIN",XSIOIN
.INSRT XSIOIN

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 100
 XSIOIN.PLR;1 PAGE 1

.comnt |
 XSIOIN - Start I/O input to buffer whose address is in r1, len in r3,
 from device in r5.

This routine starts input while checking for a subset of possible interface failures, in an attempt to prevent a buffer from being read into the wrong place. Will not detect failures in the high-order 7 bits of a 20-bit address, nor in the low 4, but will check the middle 9. After NSITRY tries fail, routine takes fail return.

Call xsioin with:
 lda r1,<packed buffer address>
 lda r3,<buffer length>
 lda r5,<io blk>
 call xsioin

```

| 0004      nsitry=4           ;how many retries before give up
OED6 1076      routine xsioin,local r2,arg r1,arg r3,arg r5
OED8 1026

OEDA 48A4      lda r2.=nsitry    ;init repeat counter
                repeat          ;store start pointer.
OEDC 301D 0002  sta r1,starti(r5)  ;buffer start address (high 16 bits).
OEE0 707D 0004  lda r7,endi(r5)   ;check start pointer store.
OEE4 A6F4      srl r7,4        ;adjust to match packed pointer.
OEE6 4D71      eor r7,r1       ;compare bits.
OEE8 4F78 01FF  while r7 .bit. =<packm_-4> ;write failed
OEEC 9A08
OEEE 49A1      sub r2,=1        ;count a try
OEOF 8A05      if zero          ;too many
                ; Trap 101.<:XSIOIN: 4 start ptr failures - page 100>
                fail return
OEF2 6026
OEF4 6076
OEF6 4FF0
OEF8 4007      endif
                ; Trap 100.<:XSIOIN: Start ptr write failed - page 100>
Oefa 90F1      endrepeat

OEFC A294      sll r1,4        ;now, store end pointer.
OEEF 4B18 1FFE  and r1.=packm
OFO2 4A13      add r1,r3       ;add buffer length
OFO4 301D 0004  sta r1,endi(r5)  ;buffer end address.

OFO8 6026      endroutine xsioin
OFOA 6006

```

```
;tables for packet core init <loc,value>  
  
FC00 0000      page relcode    ;back to code page  
  
4A38 FCO0      pkciid: 0374_D8          ;link for setups to NU.  
  
4A3A 5EBE      pkcitb: pkctyh,rldtyp  
4A3C C000      psdata,csetup  
4A40 B400      pkclha,<2-nfh>& HFF    ;:pkclho  
4A42 5EC6      pkclha,  
4A44 00FE      000C      pkcitl=-pkcitb ;length of table
```

.comnt |
POLRLD
Routine which performs stand-alone reloading of the Pluribus.
Most of the protocol work is done in PKCIC and PKCOC, below.
This routine manages the modem or Host interface for stand
alone loading or dumping, but the subroutines are also called
from the main IMP program, from Fake Host 2. POLRLD is polled
from BLT if PKCST says it is active (PKCACT).
|

```
4A46 1076      routine polrlid, uses r1-r5  
  
4A48 7078 BEAE  lck pkclok  
4A4C 9AFE      .  
4A4E 7028 5EBO  lda r2,pkcest      ;check PKC state  
4A52 8915      if odd  ;:pkcact  
4A54 7018 5EB2  lda r1,pkctim      ;active  
4A58 7118 00AO  sub r1,stime      ;timed out?  
4A5C 4F18 F000  tst r1,= HF000  
4A60 9A0D      ifnot zero      ;yes}  
4A62 4078 4F1C  call pkcupt,pkcrat  ;advance the timer  
4A66 00FO      .  
4A68 4891      lda r1,=1  
4A6A 3918 5EB6  subbm r1,pkcrct      ;too many tries?  
4A6E 8B04      if minus      ;yes  
        : ifnot r2 .bit. =pkcext ;worry if IMP transfer  
        : Trap 70,<;packet core timeout>  
        : endif  
4A70 4DA1      eor r2,=pkcact      ;clear active state  
4A72 3028 5EBO  sta r2,pkcest  
        .  
        endif  
4A76 3008 5EDA  sta r0,pkcssf      ;send a setup anyway  
        endif  
4A7A 4822      lda r2,r2      ;for a new odd test  
        endif  
4A7C 890B      if odd & r2 .bit. =pkcext      ;:pkcact  
4A7E 4F28 0100  
4A82 9A08  
4A84 4078 4A98  call r1dsub      ;check reloading stuff  
4A88 8A05      if fail      ;bad device or something  
4A8A 7078 BEBO  lda r3,m3#pkcest  
4A8E 7078 BEB6  ldb r3,m3#pkcrct  ;go idle  
        .  
        with new device
```

```
endif  
endif  
4A92 3008 BEAE unlock pkclok  
4A96 6006 endroutine polrid
```

```
;RLDSUB
;Check for any stand-alone packet core work to do.
;Only called of PKCACT and PKCEXT set, with PKCLOCK locked

4A98 1076    routine rldsub,uses r1-r5

        ;First make sure device's address and bus are okay .

4A9A 7028 5D5A  lda r2,usebus          ;I/O buses in system
4A9E 7058 5EAC  lda r5,r1ddev         ;current device to use
4AA2 4958 0010  sub r5,=devinc        ;try to keep same device
        repeat
            repeat
                add r5,=devinc           ;to next device
                repeat
                    repeat
                        lda r7,r5             ;bus for this device
                        and r7,=iomask
                        lda r1,=bsadil
                        repeat
                            cmp r7.bsadrs(-r1) ;bus exist?
                            until equal          ;maybe
                            until loop            ;never
                        endrepeat
                        until equal & r2 .bit. bittab(r1)
4ABA 8104
4ABC 7729 OEB6
4AC0 8A07
4AC2 485F 0800  lda r5,=businc(r7)   ;start at next bus
4AC6 9B03          ifnot minus        ;optimize over non-I/O
4AC8 7058 0320          lda r5,bsadrs  ;start of first bus
        endif
        endrepeat
        lda r7,r5             ;here when on good I/O bus
        and r7,=segmsk
        lda r1,=useiol
        repeat
            cmp r7.iobase(-r1) ;good seg address
            until equal          ;no such seg
            until loop
        endrepeat
        until equal          ;good bus and seg
        lda r5,=seginc(r7)   ;to next possible segment
        endrepeat
        lda r7,(r5)           ;here with good bus and segment
        until r7 = =modid    ;no device here
        endrepeat
        nop                  ;for backward compatibility
        and r7,=devtyp?magmod?hmodid?modid? HCO      ;;VDH .
        until r7 = =modid    ;got a useable modem
4AF4 81D9
        endrepeat
4AF6 3058 5EAC  sta r5,r1ddev         ;this is the (new) device
```

uribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 103
IMP.PKCORE;1 PAGE 6

```
;Now check if any output to be done

4AFA 707D 000C lda r7,statom(r5)           ;output busy?
4AFE 806D ifnot quit
4B00 4818 0007 lda r1,##<rldinb&lowmsk>_1 ;low order bits for status
4B04 301D 0006 sta r1,statim(r5)           ;hold input watchdog
        0000 .lif nz <rldinb&lowmsk>?<rldotb&lowmsk>
        1da r1,##<rldotb&lowmsk>_1 ;output status to hold
4B08 301D 000C sta r1,statom(r5)
4B0C 4F78 2000 tst r7,=mbusy               ;device done?
4B10 8A1D if zero
4B12 4818 5F6E 1da r1,=rldotb             ;output buffer to use
4B16 4078 4D58 call pkcoc,bitfex
4B1A 8000
4B1C 9A17 if success & equal            ;buffer to send
4B1E 8116
4B20 4879 A094 1da r7,=-<rldotb-hrdoff>(r1) ;bytes length
4B24 4827 1da r2,r7
4B26 A6F1 srl r7,1                  ;words length
4B28 3078 5F6E sta r7,rldotb            ;in first word
repeat
        sub r7,rldotb(-r2)      ;checksum buffer
        until loop
endrepeat
4B32 3278 5F72 addm r7,rldotb+chkh    ;adjust checksum
4B36 7038 00B0 1da r3,maprel          ;page buffer is on
4B3A 4C38 01F6 ior r3,=<0/rldotb>-4 ;offset packed
4B3E 303D 0008 sta r3,starto(r5)
4B42 4C18 8000 ior r1,=mendf         ;just this buffer
4B46 301D 000A sta r1,endo(r5)        ;start output
endif
endif
```

;Now check input

```
4B4A 707D 0006    lda r7,statim(r5)      ;check input
4B4E 4F78 2000    tst r7,=mbusy
4B52 8A42          if zero
4B54 4818 5EDE    lda r1,=r1dinb
4B58 7078 5EB0    lda r7,pkcst
4B5C 8B07          if minus ::pkcist
4B5E 4078 4BEA    call pkcic,bltfex
4B62 8000
4B64 8A02          if fail
4B66 6006          return
4B68 902B          endif
4B6A 707D 0004    lda r7,endi(r5)      ;still hold buffer
4B6E 8B28          if minus
4B70 4971          sub r7,r1
4B72 4B78 1FFF    and r7,=0/-1
4B76 4E78 008E    if r7 < =bufend
4B7A 8222
4B7C 4821          lda r2,r1
4B7E 483F 0002    lda r3,=2(r7)
4B82 A6B1          srl r3,1
4B84 6132          repeat
4B86 49F2          sub r3,(r2)+      ;buffer pointer
4B88 8BFE          sub r7,=2
4B8A 4EBO          until minus
4B8C 8119          endrepeat
4B8E 7078 5EE0    if r3 = =0          ;good checksum
4B92 4B78 C000    lda r7,r1dinb+typ
4B96 4E78 C000    and r7,=paktyp
4B9A 8112          if r7 = =r1dtyp
4B9C 7078 5EE6    lda r7,r1dinb+cpsatd
4BA0 7678 5EC2    if r7 = pkcmyi ;for me
4BA4 810D
4BA6 7078 5EE4    lda r7,r1dinb+cpsats
4BAA 3078 5EC4    sta r7,pkcnim ;remember neighbor
4BAE 48FF          lda r7=pkctmx ;retries max
4BB0 3878 5EB6    stab r7,pkcrct
4BB4 4078 4BEA    call pkcic,bltfex ;process this buffer
4BB8 8000
4BBA 8A02          if fail
4BBC 90D5          return ;don't read more
4BBD 8000          ;hold off more input
4BCE 8A02          endif
4BDF 90D5          endif
4BEE 8000          endif
4BF0 8000          endif
4BF2 8000          endif
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 105
IMP.PKCORE;1 PAGE 8

;Now start a new input

```
4BBE 4818 01ED    lda r1.=<0//rlldinb>_4
4BC2 7418 00B0    ior r1.maprel           ;packed start address
4BC6 4838 009C    lda r3.=<rbflen*words>+<rlldinb-<rlldinb&<0#-1_4>>>-hrdoff
                  ;buffer length plus fudge for XSIOIN
4BCA 4078 0ED6    call xsioin            ;start a new input
4BCE 8A04          if fail              ;bad device
4BD0 6076          fail return        ;so stop using it
4BD2 4FF0
4BD4 4007          endif
                  endif
4BD6 9002          else
4BD8 90FC          fail return        ;got a QUIT from STATOM
                  ;chuck device
                  endif
4BDA 7078 018A    lda r7.consol       ;I have a console?
4BDE 8B05          if minus            ;yes
4BE0 7018 5ED6    lda r1.pkcadd       ;display packet core address
4BE4 301F 0002    sta r1.%dreg1-%areg1(r7)
                  endif
4BE8 90BF          endroutine rlldsub
```

.comnt |
PKCIC - process input buffers to Packet Core.
Validates packets as they arrive, checking source IMP and Host if
we are "locked", and current address and type if we are active and
receiving a core message. Should be called repeatedly to process all
parts of a core message; signals that more must still be done with
the current buffer via a fail return. Arguments: R1 has the buffer
address of the current input buffer, which is assumed to be mapped
into map 2; inline is the proper state bit to turn on for BLT (either
BLTEXT for a reload, or BLTFAK for packet input from Fake Host 2).
Must be called with PKCLOCK locked.
|

4BEE 6027 routine pkcic, arg r1, inline r2, uses r1-r4
4BEC 1076

4BEE 7078 5EBO lda r7,pkcst ;check state
4BF2 9B66 ifnot minus ;:pkcist ;not in midst of buffer
4BF4 890D if odd ;:pkcact ;something happening
4BF6 7078 5EC8 lda r7,pkclim ;so check if locked
4BFA 9AOA ifnot zero ;we are locked
4BFC 7038 5EC6 lda r3,pkclha ;check handling type/host
4C00 7679 000C if r7 <> dsth(r1) } r3 <> pkth(r1)

4C04 8104
4C06 7639 000A
4COA 9102
4C0C 6006 return ;can't take it
endif
endif
endif
4COE 7039 0010 lda r3,cmark(r1) ;get password
4C12 4B38 FE00 and r3,=cpmask
4C16 4E38 B400 if r3 = =csetup ;got a setup
4C1A 8125
4C1C 4848 FFE8 lda r4,=-psetul+pkth ;how much to copy
4C20 4A9A add r1,=pkth ;and where to start
repeat
 lda r3,(r1)+ ;next buffer word
 sta r3,psetup+psetul(r4)+ ;into params
 until loop
endrepeat
4C22 6031
4C24 203C 5EDE
4C28 88FD

4C2A 7818 5ED5 l1dab r1,pkcf1n ;swap the line bytes
4C2E 7838 5ECD l1dab r3,pkcl1n
4C32 3818 5ECD stab r1,pkcl1n
4C36 3838 5ED5 stab r3,pkcf1n
4C3A 7018 BECE lda r1,m3//pkcfha ;:pkcfho
4C3E 7038 BED2 lda r3,m3//pkcfid
4C42 7048 BED0 lda r4,m3//pkcfim ;third-party sender?
4C46 9A07 ifnot zero ;yes
4C48 3018 5EC6 sta r1,pkclha ;:pkclho
4C4C 3038 5ECA sta r3,pkclid
4C50 3048 5EC8 sta r4,pkclim ;copy address stuff
endif

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 107
IMP.PKCORE;1 PAGE 10

```
;end of setup processing

4C54 4878 0089    lda r7,=pkcact}bltcck}bltsdf ;make packet core go
4C58 3478 5EBO    iorm r7,pkcst
4C5C 4078 4F1C    call pkcupt,pkclto      ;hold timer long
4C60 0640
4C62 90D5    return           ;all done with setup
endif

;here when got core packet or something else

4C64 7078 5EBO    lda r7,pkcst          ;if its core, must be active
4C68 8904    ifnot odd & r3 = =cpcore   ::pkcact ;if not active and a core
4C6A 4E38 B600
4C6E 9107
4C70 4E38 B800    if r3 = #abort        ;if it's an abort instead
4C74 8103
4C76 7078 BEBO    set r7 = m3#pkcst     ;stop packet core.
endif
return
endif
4C7A 90C9
4C7C 7038 5EC8    lda r3,pkclim       ;already locked?
4C80 8A0C    if zero            ;no, so remember this one
4C82 4879 000A    lda r7,=pkth(r1)    ;where to copy from
4C86 4838 FFFA    lda r3,=-3*words   ;and how much
repeat
4C8A 6047    lda r4,(r7)+      ;dsth pkth cpdl
4C8C 204B 5ECC    sta r4,pkclha+<3*words>(r3)+
4C90 88FD    until loop        ;pkclho pkclim pkclid
endrepeat
4C92 7037    lda r3,(r7)      ;cmark
4C94 3838 5ED5    stab r3,pkcf1n    ;line number to sender
endif
4C98 4A18 0012    add r1,=ccaddr   ;get to first piece
4C9C 7031    lda r3,(r1)      ;its address
4C9E 7638 5ED6    if r3 <> pkcadd  ;one we want?
4CA2 9105
4CA4 9203    ifnot g           ;no, too big?
4CA6 3008 5EDA    sta r0,pkcssf   ;yes, force setup
endif
return
endif
4CAA 90B1
4CAC 7079 0002    lda r7,cctype-ccaddr(r1) ;get type and length
4CBO 7578 5ED4    eor r7,pkctyp
4CB4 4B78 FE00    and r7,=pkctmk   ;type must also match
4CB8 9A02    ifnot zero
4CBA 90A9    return
endif
4CBC 901B    else
```

;Here if in the midst of a piece of core

```
4CBE 7078 5DBC    lda r7,bltst      ;be sure BLT is done.
4CC2 8904    if odd      ;;bltact
4CC4 6076    fail return ;not yet
4CC6 4FF0
4CC8 4007
4CCA 4818 7FFF    endif
4CCE 3318 5EBO    lda r1,=-1?pkcist
4CD2 4B78 7074    andm r1,pkcst      ;clear input pending bit
4CD6 9A06    ifnot zero
4CD8 3478 5EBO    iorm r7,pkcst      ;got some BLT error
4CDC 3008 5EDA    sta r0,pkcssf      ;save errors for later
4CEO 9096    return
4CEE 7018 5EB8    endif
4CE2 7018 5EB8    lda r1,pkcdon      ;force a setup
4CE6 3218 5ED6    addm r1,pkcadd
4CEA 3118 5ED8    subm r1,pkclen      ;update packet core address
4CEE 7018 5EBA    lda r1,pkcbfa      ;and remaining length
4CF2 6031    endif
4CF4 8A0D
4CF6 4078 4F1C    repeat
4CF6 4078 4F1C    lda r3,(r1)+      ;skip zero length pieces
4CF6 4078 4F1C    if zero
4CF6 4078 4F1C    call pkcupt,pkclto ;next piece address
4CF6 4078 4F1C    ;end this buffer
4CF6 4078 4F1C    ;hold timeout
4CFA 0640
4CFC 48FF    lda r7,=pkctmx      ;and stick with device
4CFE 3878 5EB6    stab r7,pkcrct
4D02 7078 5ED6    lda r7,pkclen      ;how much left in transfer
4D06 8A03    if zero
4D08 4078 4F36    call pkcclr      ;done it
4D08 4078 4F36    endif
4DOC 9080    return
4DOC 9080    endif
4DOE 6041    lda r4,(r1)+      ;reset some things
4D10 4873    lda r7,r3
4D12 7178 5ED6    sub r7,pkcadd
4D16 3278 5ED6    addm r7,pkcadd
4D1A 3178 5ED8    subm r7,pkclen
4D1E 4F48 0OFF    tst r4,=pkclmk
4D22 9AE8    while zero
4D22 9AE8    endrepeat
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 109
IMP.PKCORE;1 PAGE 12

;Now have R3: next piece address, R4: type and length,
;R1: address of core data in buffer

```
4D24 4B48 FEO0 and r4,=pkctmk
4D28 A3C7 r11 r4,7 ;just type
    0000 .if nz newpckc ;new-format packet core
        if r4 >= =anypro ;legal for us
            if r4 >= =plrcm ;convention for common
                eor r4,=plrcm?b1tnlc
            endif
        .iff ;nz newpckc
4D2A 4948 0040 sub r4,=plrpkc ;Pluribus convention
4D2E 9B0E ifnot minus ;get H16 type??
    .endc
4D30 4C24 ior r2,r4 ;set source type
4D32 1056 save r5 ;do our caller a favor
4D34 7059 FFFE lda r5,-words(r1) ;length of this piece
4D38 4078 4E94 call pkcb1t ;try to get BLT
4D3C 9A05 if success ;got it
4D3E 4878 8000 lda r7,=pkcist ;input being processed
4D42 3478 5EBO iorm r7,pkcst
    endif
4D46 6056 restore r5
4D48 90BE fail return ;to hold buffer
    endif
4D4A 48F3 lda r7,=pkcety}pkcact ;signal type error
4D4C 3478 5EBO iorm r7,pkcst
4D50 3008 5EDA sta r0,PKCSSF ;and force a setup with error

4D54 4008 4C0C endroutine
```

.comnt |

PKCOC

Check if any output from packet core to send. Call with R1: buffer address, R2: BLTFAK or BLTFEX, Map 2 set up. Returns fail if buffer is now in use (awaiting BLT completion), succeed with cc equal if a buffer to send (endpointer returned in R1), succeed with cc not equal, if nothing to send now. Must be called with PKCLOCK locked.

|

4D58 1076 routine pkcoc,arg r1,inline r2,local r2,uses r2-r4,result r1

4D5A 1026

4D5C 6027

4D5E 307E 0002

4D62 0281 sst %e ;assume we'll send something

4D64 7078 5EBO lda r7,pkcst

4D68 4F78 0200 tst r7,=pkcost

4D6C 9A2A ifnot zero

4D6E 7078 5DBC lda r7,bltst

4D72 8905 if odd

4D74 6026 fail return

4D76 6076

4D78 4FF0

4D7A 4007

endif

4D7C 4838 FD7F lda r3,=-1?pkcost?bltsdf

4D80 3338 5EBO andm r3,pkcst

4D84 4B78 70F4 and r7,=blters>bltsdf

4D88 3478 5EBO iorm r7,pkcst

4D8C 4F78 7074 tst r7,=blters

4D90 8A16 if zero

4D92 4078 4F1C call pkcupt,pkcsto

4D96 00C8

4D98 7078 5EB8 lda r7,pkcdon

4D9C 3278 5ED6 addm r7,pkcdadd

4DAO 3178 5ED8 subm r7,pkclen

4DA4 8A05 if zero

4DAG 7078 BEBO lda r7,m3//pkcost

4DAA 7078 BECA lda r7,m3//pkclid

endif

4DAE 48B0 lda r3,=0

4DB0 7018 5EBA lda r1,pkcbfa

4DB4 4078 4F2A call sendst

4DB8 6026 return

4DBA 6006

endif

4DBC 3008 5EDA sta r0,pkcssf

endif

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 111
IMP.PKCORE;1 PAGE 14

;Now check whether to send a setup

```
4DC0 7038 BEDA lda r3,m3#/pkcssf ;setup send flag?
4DC4 9A1C ifnot zero
4DC6 4838 FFDE lda r3,=-psetul ;how much to copy
repeat
4DCA 607B 5EDE lda r7,psetup+psetul(r3)+ ;leave with pkcsrf
4DCE 9803 until loop ;into buffer
4DD0 2071 sta r7,(r1)+ ;into buffer
4DD2 90FC endrepeat
4DD4 4937 sub r3,r7 ;flip sign
4DD6 9B02 ifnot minus ;we're receiving
4DD8 48B4 lda r3,=4 ;set his rate
endif
4DDA 7078 5EBO if pkcst .bit. #bilters+pkcety ;bit got errors- send abort
4DDE 4F78 7076
4DE2 9A05
4DE4 4838 B800 set r3 = #cabort ;return error flag
4DE8 4918 0010 sub r1,#pkcsrf-psdata ;reduce size of packet
endif
4DEC 4078 4F2A call sendst ;terminate buffer with state
4DF0 4F78 7076 tst r7,-bilters}pkcety
4DF4 9A03 ifnot zero ;if reporting an error
4DF6 4078 4F36 call pkcc1r ;stop further work
endif
4DFA 90DF return ;with buffer to send
endif
```

;No setup to send, maybe send core

```
4DFC 7078 5EBO  lda r7,pkcst
4E00 8948  if odd  ;;pkcact      ;pkt core is active
4E02 7078 5EDC  lda r7,pkcsrf   ;we sending?
4E06 9B45  ifnot minus    ;yes
4E08 7038 018C  lda r3.wdis   ;if system is off
4EOC 4F38 0400  if r3 .bit. =totstb
4E10 9A02
4E12 A6F1      srl r7,1       ;STIME is slow, so compensate
4E14 7038 5EB4  endif
4E18 7138 00AO  lda r3,pkcotm  ;time for next send
4E1C 4F38 FFOO  sub r3,stime
4E20 9A35  tst r3,= HFFOO    ;reached it?
4E22 1056  ifnot zero    ;yes, get ready for BLT
4E24 4842  save r5       ;protect caller's r5
4E26 7828 5ED4  ldatb r2,pkctyp  ;core type
4E2A 7058 5ED8  lda r5,pkclen  ;remaining length
4E2E 4E58 0039  if r5 > =pkcmax ;more than one BLT bite
4E32 8C03
4E34 4858 0039  lda r5,=pkcmax ;one BLT bufferful
4E38 A2A8      endif
4E3A 4C52      sll r2, H8     ;type to top
4E3C A3A7      ior r5,r2    ;plus length
        0000  r11 r2,7      ;type field
        .if nz newpckc  ;new pkc format
          if r2 >= =anypro
            if r2 >= =plrcm
              eor r2,=plrcm?bltnlc
            endif
        .iff :nz newpckc
          sub r2,=plrpkc   ;Pluribus convention
        .endifc ;nz newpckc ;complain if H16 type
        .endc ;nz newpckc
        sub r3,r7      ;:pkcsrf ;get next send time
4E44 4937      subm r3,pkcotm  ;copy length
4E46 3138 5EB4  lda r3,=-cmark-words
4E4A 4838 FFEE  repeat
        lda r7,psetup+cmark+words(r3)+  ;until loop
        until loop      ;exit with setup flag
        sta r7,(r1)+   ;into buffer
        endrepeat
        add r7,=cpcore-csetup  ;core flag
        sta r7,(r1)+   ;piece address
4E5E 7038 5ED6  lda r3,pkcad
4E62 2031      sta r3,(r1)+  ;store type,,length .
4E64 2051      sta r5,(r1)+  ;set dest type
4E66 4C42      ior r4,r2   ;grab BLT
4E68 4078 4E94  call pkcb1t
4E6C 9A05      if success   ;got it
4E6E 4878 0200  lda r7,=pkcost
4E72 3478 5EBO  iorm r7,pkcst
        endif
        else
          lda r7,=pkcety
          iorm r7,pkcst
        endif
4E76 9006      ;got a bad type
4E78 48F2
4E7A 5EBO
```

```
4E7E 3008 5EDC      sta r0,pkcsrf      ;send a setup next time
endif
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 113
IMP.PKCORE;1 PAGE 16

;core message is ready to go or got error
4E82 6056 restore r5
4E84 4008 4D74 fail return ;hold buffer in any case
;can't send yet - too soon
4E88 9004 else
4E8A 4078 4F1C call pkcupt,pkcsto ;hold timer while outputting slow
4E8E 00C8 endif
endif
4E90 0201 rst %e ;nothing to send
4E92 9093 endroutine

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 114
 IMP.PKCORE;1 PAGE 17

```
.comnt |
PKCBLT
Packet Core subroutine to try to get BLT going. Call with R1: buffer
address, R2: BLT source type, R3: PKC address, R4: BLT dest type,
R5: BLT transfer size (words). Fail return if BLT is busy. Smashes R3-R5.
|
4E94 1076      routine pkcblt, arg r1-r5, uses r3-r5

4E96 7078 BDBA  lock bltlok          ;now grab BLT
4E9A 9AFE
4E9C 7078 5DBC  lda r7,bltst
4EA0 8906      if odd :;bltact    ;must recheck
4EA2 3008 BDBA  unlock bltlok     ;can't get BLT
4EA6 6076      fail return
4EA8 4FF0
4EAA 4007      endif
4EAC 3018 5DC8  sta r1,bltbfa    ;set buffer address
4EB0 3028 5DD6  sta r2,bltsty    ; source type,
4EB4 A3B1      r11 r3,1        ; transform PKC -> BLT
4EB6 3038 5DC0  sta r3,bltadd   ; address,
4EBA 3048 5DDA  sta r4,bltdty   ; and dest type.
4EBE 4B58 0OFF  and r5,=pkclmk  ;length in words
4EC2 3058 5EB8  sta r5,pkcdon  ;how much in this transfer
4EC6 A2D1      s11 r5,1        ;now bytes
4EC8 3058 5DC2  sta r5,blt siz
4ECC 4A51      add r5,r1        ;address after this piece
4ECE 3058 5EBA  sta r5,pkcbfa  ;for next in buffer
4ED2 7078 808C  lda r7,m2#/slfptr
4ED6 3078 5DC6  sta r7,bltbm
4EDA 48FO      lda r7,=0        ;mask for common transfers
4EDC 4B48 007F  and r4,=pkctmk - H9
4EE0 4ECF      if r4 = =anyproc
4EE2 8103
4EE4 7078 5E28  lda r7,rckcon  ;use stage RC consensus
4EE8 3078 5DDE  endif
4EEC 3078 5DDC  sta r7,bldpi
4EOF 3078 5DD8  sta r7,bldpm
4EF4 7078 5EBO  sta r7,bltspm
4EF8 4B78 0088  lda r7,pkfst
4EFC 4CF1      and r7,=bltcck}bltsdf
4FEF 4F28 C000  ior r7,=bltact
4FO2 9AO2      if r2 .bit. =bltfex+bltfak
4FO4 4CF2      ;dest proc masks
4F06 3078 5DBC  sta r7,bltst
4FOA 7078 00AO  lda r7,stime
4FOE 4A78 0078  add r7,#bltrat
4F12 3078 5DBE  sta r7,bitto   ;give BLT some time to start up
4F16 3008 BDBA  unlock bltlok
|
4F1A 6006      endroutine
```

```
.comnt |
PKCUPT - updates PKCTIM using in-line argument
|
4F1C 6017        routine pkcupt, inline r1, uses r1
4F1E 1076

4F20 7218 00AO  add r1,stime
4F24 3018 5EB2  sta r1,pkctim                                          ;when pkc should go idle

4F28 6006        endroutine

.comnt |
SENDST
Sends the last protocol word of the buffer (from R3), and
appends the interesting state bits from BLT state (errors
and BLTSDF if spare matches) and PKC state (all bits).
Arguments: R1 is buffer pointer, R3 is last data word.
Returns state word bits in R7.
|
4F2A 1076        routine sendst, arg r1, arg r3, result r7
4F2C 2031        sta r3,(r1)+                                                  ;last data word
4F2E 7078 5EBO  lda r7,pkcst                                                  ;BLT errors and pkc errors
4F32 2071        sta r7,(r1)+                                                  ;at end of message

4F34 6006        endroutine

4F36 1076        routine pkcclr
4F38 7078 BEBO  lda r7,m3//pkcst                                                  ;read&clear current state
4F3C 4F78 0100  if r7 .nbit. =pkcext                                                  ;fake called us
4F40 8A03
4F42 7078 BEB6  lda r7,m3//pkcrct                                                  ;also force random reload
4F46 7078 4A38  endif                                                                          ;reinit msg-id to default
4F4A 3078 5ECA

4F4E 6006        endroutine pkcclr
```

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 116
 IMP.PKCORE;1 PAGE 19

```
.comnt |
TRYRLD
Routine to start a packet core transfer when a page is lost.
Called with R1: Type to reload. Inline argument is a pointer
to the checksum word for the logical page to reload, i.e., the
starting transfer address (limit word is assumed to follow it).
|
4F50 1076      routine tryrlid, arg r1, inline r2, local r3, uses r1-r2
4F52 1036
4F54 6027
4F56 307E 0002

4F5A 7078 BEAE  lock pkclok
4F5E 9AFE
4F60 7078 5EBO  lda r7,pkcst          ;packet core in progress?
4F64 9942  ifnot odd    ::pkcact   ;no, proceed
4F66 7078 5EB6  lda r7,pkcrct        ;tries this device
4F6A 8B1B  if minus           ;time to give up
4F6C 48F2  lda r7,=pkctry        ;tries per device
4F6E 3878 5EB6  stab r7,pkcrct       ;how much to clear
4F72 4878 0142  lda r7,=pkcc11
repeat
4F76 577F BEBC  tst r7,m3/pkcc1a(-r7)
4F7A 88FE  until loop
endrepeat
4F7C 7078 01A4  lda r7,1clock        ;some RTC
4F80 707F 0004  lda r7,rtcsws-rtcadd(r7) ;my IMP number
4F84 3878 5EC3  stab r7,pkcmyi+1
4F88 3018 5ED2  sta r1,pkcfid        ;type for neighbor to report
4F8C 48FC  lda r7,=pkc1t1        ;init table length
repeat
4F8E 503F 4A3A  lda r3,pkcitb(-r7)
4F92 10BF 4A3A  sta r3,@pkcitb(-r7) ;table-driven init
4F96 88FC  until loop
endrepeat
4F98 4878 0010  lda r7,=devinc        ;chuck this device
4F9C 3278 5EAC  addm r7,r1dddev
endif
4FA0 707A 0002  lda r7,words(r2)      ;end address
4FA4 4972  sub r7,r2                 ;length
```

;rest of TRYRLD

| | | |
|----------------|---|---|
| 0000 | .if nz newpkc if r1 <> =anypro ior r1,=plrcm and r7,=0/-1 and r2,=0/-1 ior r2,=bltlog endif | ;common transfer ;keep within window ;offset in page ;and logical page |
| 4FA6 4C18 0040 | .iff ;nz newpkc ior r1,=plrpkc if r1 <> =plrpkc+anypro | ;old Pluribus convention ;common reload |
| 4FAA 4E18 004F | | |
| 4FAE 9107 | | |
| 4FB0 4B78 1FFF | and r7,=0/-1 | ;keep within window |
| 4FB4 4B28 1FFF | and r2,=0/-1 | ;offset in page |
| 4FB8 4C28 8001 | ior r2,=bltnlc}bltlog endif | ;logical common address |
| 4FBC A6F1 | .endc ;nz newpkc srl r7,1 | ;words length |
| 4FBE 3078 5ED8 | sta r7,pkclen | |
| 4FC2 A291 | sll r1,1 | |
| 4FC4 3818 5ED4 | stab r1,pkctyp | ;core type to get |
| 4FC8 A7A1 | rrl r2,1 | ;word address |
| 4FCA 3028 5ED6 | sta r2,pkcad | |
| 4FCE 3008 5EDA | sta r0,pkcssf | ;force a setup |
| 4FD2 4878 FFFF | lda r7,=-1 | |
| 4FD6 3078 5EDC | sta r7,pkcsrf | ;receive a reload |
| 4FDA 4078 4F1C | call pkcupt,pkcrat | ;setup time |
| 4FDE 00FO | | |
| 4FE0 4878 0101 | lda r7,=pkcact}pkcext | |
| 4FE4 3078 5EB0 | sta r7,pk cst endif | ;start pkt core reload |
| 4FE8 3008 BEAE | unlock pkclok | |
| 4FEC 6036 | endroutine | |
| 4FEE 6006 | | |

; end of IMP.PKCORE

pluribus IMP Loader PLURIBUS V2.9B 25-Jun-87 11:20:52 PAGE 118
STAGEK.PLR;1 PAGE 81

.comnt |
Stage RC - Reliability page Code checksum

Check results of rely checksum from stage RK. Failure
triggers a reload and hangs. No consensus necessary.
If debugging mode is turned on, the pseudo-halt bit in the
snapshot area is checked, and if it's on, the processor will
hang itself right here.

```
|  
src00:  
repeat  
  
4FF0 4078 07DC    call wsleep           :rest a while  
4FF4 7018 80B0    lda r1,m2#comrel   ;check results of stage RK.  
4FF8 890C    if odd                 ;rely page is gone, reload it  
4FFA 4078 0D3E    call sbad            ;inhibit later stages.  
4FFE 7018 40DE    lda r1,limtab+relytyp ;known end of rely  
5002 3018 40B8    sta r1,tlimit        ;for rely page here  
5006 4890    lda r1,=relytyp       ;logical page  
5008 4078 4F50    call tryr1d,cksum    ;arg is limits  
500C 40B6  
500E 9003    else                  ;all's well  
5010 4078 0D50    call sokay            ;endif  
  
5014 90EE    endrepeat  
  
;End of STAGEK.PLR
```

;Now for final patching of STAGE
FC00 0000 \$FINKER

40C0 FEED
02FC FEED
FC00 0000 \$FINSTAGE
40DE 7D50
40E0 FFFF
40E2 FFFF
40E4 FFFF
40E6 FFFF
40E8 FFFF
40EA FFFF
40EC FFFF
40FA 0001
40FC 0001
40FE 0000
4100 4000
4104 0010
4106 0020
4108 0030
410A 0050
410C 0000
410E 0010
4110 0020
4112 002E
4114 0001
4116 E100
4118 E200
411A F100
411C F200
40BA 0000
40BC 0000
031A 04DC
0314 00C8
0316 0001
0318 0001
0320 E000
0322 F000
0324 8000
0326 8000

.if p2, .outsym implod.sym

0A1C .end wst ;start the loader code up

82 SECONDS RUN-TIME



P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 1
IMPOPS.PLR;1 PAGE 1

.title IMP Operating System

.comnt |

This file assembles the Operating System part of the IMP program.
Included herein are the non-kernel part of STAGE, the non-IMP part,
of DDT, and OPSYS (PID dispatch and some central routines). It is
intended that this assembly not need redoing every time the IMP
program is modified, thus saving assembly time in the long run.
In addition, it forms a common base for the specialized IMP
systems: the PSE IMP, the PTIP's IMP, etc. |

;First get IMPLOD (IMP Loader) Symbols

.insym IMPLOD.SYM

;Now for some additional macros

.insert "LPMAC"

.INSRT LPMAC

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
LPMAC.PLR;1 PAGE 1

PAGE 2

```
:LPMAC.PLR
:Eric Roberts 10-Aug-78
-----
; This file contains macros used in loop
; which do not easily fit into other categories.

:Revision history:
```

```
;System initialization

;      The STAGE and POPS systems call the routine
;SYSINI as the initialization routine for the WARM
;page.  The initialization system is table driven,
;and there are a number of macros which allow additional
;initialization code to be inserted into the tables.

;      The principal initialization specification is
;given by the macro $INIT which has one of the following
;general forms:
;
;      $INIT X,v           ;Initialize location X to
;                          ;The value v
;      $INIT X,1,v         ;Initialize the block of 1
;                          ;Bytes beginning at X to the
;                          ;Word value v

;      In addition, other macro forms are available for
;other standard initialization functions.
```

```
;The $INIT macro
;The $INIT macro stores entries on the deferred
;execution list %ILST (see GENSUB).

$LIST %ILST           ;Declare the %ILST
$LIST %RLST           ;Routine lists
$LIST %PLST           ;And the PID list

.MACRO $INIT ARG1,ARG2,ARG3
    .IF B <ARG3>
        $XINIT <ARG1>,2,<ARG2>
    .IFF
        $XINIT <ARG1>,<ARG2>,<ARG3>
    .ENDC
.ENDM

.MACRO $XINIT ADDR,COUNT,VALUE
    %M==<ADDR> - D13      ;Get map portion of address
    $MAPCHK \%M             ;And see if it changed
    $ENTER %ILST,<ADDR>
    $ENTER %ILST,<COUNT>
    $ENTER %ILST,<VALUE>
.ENDM

.MACRO $MAPCHK MP
    .IF DF %IMAP'MP       ;Is this mapped?
        $DOPPAGE <%KEY=>,<KEY>   ;Get the page key
        .IF NZ %KEY-%IKEY'MP     ;Same as last for this map?
            $ENTER %ILST,O       ;Mark a map change
            $ENTER %ILST,%IMAP'MP  ;And map address
            $DOPPAGE <$ENTER %ILST,>,<TYP>
            %IKEY'MP=%KEY
        .ENDC
    .ENDIF
.ENDM

;Define initial keys and maps

FC02    %IMAP3=%MAP1
FC04    %IMAP4=%MAP2
FC06    %IMAP5=%MAP3

FFFF    %IKEY3=-1
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
LPMAC.PLR;1 PAGE 3.1

PAGE 5

FFFF %IKEY4== -1
FFFF %IKEY5== -1

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
LPMAC.PLR:1 PAGE 4

PAGE 6

165

```
;Initialization routines (continued)

;The macro below assembles the initialization routine
;for constants

.MACRO $RINIT
    .WORD INITLIST          ;Call the INITLIST routine
    $APPLY .WORD,%ILST      ;Enter the initialization list
    .WORD NIL                ;And terminate with a nil
    $LIST %ILST
.ENDM

.MACRO $PINIT
    .WORD INITPID           ;Initialize the BASE table
    $APPLY $PID1,%PLST      ;Run through the list of PIDs
    .WORD NIL                ;Use an end marker
    $LIST %PLST
.ENDM

;Macro to enter a PID entry

.MACRO $PID1 NAME
    .WORD $'NAME
    .WORD NAME
.ENDM

;Macro to enter a new initialization routine

.MACRO $IROUTINE NAME
    .IF Z NAME & HCOOO      ;Is this in local?
    $ENTER %RLST,<$IR1 LOCAL> ;Yes, don't change map
    .IFF
        $DOPAGE |$ENTER %RLST,<$IR1 |. |>|
    .ENDC
    $ENTER %RLST,NAME
.ENDM

FFFF      M%LOCAL == -1
```

```
;Macro to build the init table

.MACRO $ITABLE
%FLG=0
$IFDF1 %ICHAIN,<%FLG==1>
.IF Z %FLG
ITABLE:
.IFF
%NDOT=.
.IF Z %NDOT & HCOOO
%NTYP==1
.IFF
$DOPPAGE <%NTYP==>,<TYP>
.ENC
SAVEPAGE TEMP
PAGE %IPAGE
PAGE DUMMY
.=%ICHAIN
.WORD 0.%NTYP.%NDOT
PAGE TEMP
.ENC
$APPLY <>,%RLST
SAVEPAGE %IPAGE
%ICHAIN=.
.WORD NIL
.WORD 0.0
$LIST %RLST
$ENTER %RLST,$RINIT
$ENTER %RLST,$PINIT
.ENDM

.MACRO $IR1 PG
.IF NZ M%'PG + 1
$DOPPAGE <%%PHYS == P%>,<>
.IF NZ %%PHYS - P%'PG
SAVEPAGE TEMP
PAGE PG
%NDOT=.
$DOPPAGE <%NTYPE==>,<TYP>
PAGE TEMP
.WORD 0.%NTYPE,%NDOT
PAGE PG
.ENC
.ENC
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 8
LPMAC.PLR;1 PAGE 5.1

.ENDM

\$ENTER %RLST,\$RINIT :Always do this initialization
\$ENTER %RLST,\$PINIT

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 9
LPMAC.PLR;1 PAGE 6

;Miscellaneous definitions

```
.MACRO LOCKDEF NAME
SAVEPAGE TEMP
PAGE VARS
NAME==M3//.
.BLKW 1
$INIT M1#NAME,1
PAGE TEMP
.ENDM
```

;Locks on vars page
;Locks defined through M3
;Leave space for lock
;Initialize the lock

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
IMPOPS.PLR;1 PAGE 1.1

PAGE 10

.insert "PROC"
.INSRT PROC

```
:PROC.PLR
:Eric Roberts 5-Oct-77
:-----
; This file contains the macros for defining
;processes to run under POPS.

;Revision history

;19-Jun-78 ESR:
;      Rewrote file to include the notion of MESSAGE PROCESS.

;10-Aug-78 ESR:
;      Added code to define process entities and primitives.

;15-Aug-78 ESR:
;      Restructured process entity mechanism to be current
;with HSMIMP documentation.

;28-Sep-78 ESR:
;      Redesigned INITFORK facility to provide for process
;initialization.

;14-Dec-78 ESR:
;      Added the CHAIN = xx directive to MESSAGE PROCESS.

;4-Jan-79 ESR:
;      Fixed the interaction between MESSAGE PROCESS and
;CLOCK processes.

;4-Apr-79 ESR: *** MAJOR REVISION ***
;      Revised the process package to correspond to the new
;process design.
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
PROC.PLR;1 PAGE 2

PAGE 12

;Miscellaneous definitions

DEAD STKPASS = HDEAD ;Password for stack checking

;Define Fork Control Block (FCB) entries

STRUCT SYSFCB

| | | |
|------|--------------------|----------------------------------|
| 0000 | FCBSTK: | ;Stack grows from here |
| 0002 | FCBCHAIN: PTR(FCB) | ;Chain field |
| 0004 | FCBSTATE: WORD | ;Fork state |
| 0006 | FCBSP: WORD | ;Save the stack pointer |
| 0008 | FCBCODE: PTR(CODE) | ;Back pointer to code (in local) |
| 000A | FCBRELY: PTR(FCB) | ;Queue pointer for reliability |
| | FCBTIME: WORD | ;Time this was last completed |

ENDSTRUCT SYSFCB

;Fork states

| | | |
|------|-------------|-------------------------------------|
| 0000 | F.LOCKED=0 | ;Fork is busy |
| 0001 | F.ILEG=1 | ;Fork has been unlocked by Stage |
| 0002 | F.RUNNING=2 | ;Fork is running |
| 0003 | F.QUEUED=3 | ;Fork is on run queue |
| 0004 | F.WAITING=4 | ;Fork is waiting for event |
| 0005 | F.RQUEUED=5 | ;Fork is running with event pending |

```
;Define PROCESS macro
;PROCESS <name>,[<args>]
;
;      Defines the beginning of a process.
;The argument forms are:
;
;      FCBAREA = xxx
;      STACKLIMIT = xxxx
;
;      The actual process code is preceded by a
;process data block which contains some constant
;parameters of the process, which are given by
;the structure below
;
STRUCT PDBBLK
    WORD -4          ;Start four words early
    PDBLIMIT: WORD   ;Limit of the stack
    PDBOFFSET: WORD  ;Offset of the FCB from data block
    PDBPKOFF: WORD   ;Packed version of PDBOFFSET
    PDBRATE: WORD    ;Rate at which to check this fork
ENDSTRUCT PDBBLK
```

```
;Actual PROCESS macro

.MACRO PROCESS NAME,ARG1,ARG2,ARG3,ARG4,ARG5
$STACKLIMIT == -1
$FCBAREA == -1
$RATE == D5000
%SRADIX == %RADIX
%RADIX == D10
.IRP ARG,<ARG1,ARG2,ARG3,ARG4,ARG5>
.IIF NB <ARG>, $OPT ARG
.ENDM
%RADIX == %SRADIX
.IF LT $FCBAREA
.IF LT $STACKLIMIT
$STACKLIMIT == 0
.ENC
$FCBAREA ==$STACKLIMIT + D32
.ENC
.IF LT $STACKLIMIT
$STACKLIMIT ==0
.ENC
.IF NZ $FCBAREA & HF
.ERROR FCBAREA not properly aligned in PROCESS NAME
.ENC
%'NAME=$FCBAREA+LSYSFCB
$LCCHK NAME
JSB R3,(R3)           ;Assume no initialization
.%INIT=                 ;INITFORK will change this
.ENC
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
PROC.PLR;1 PAGE 6

PAGE 16

175

```
;Process macro subroutines

:$LCCHK
;      Check to see that process header is in local

.MACRO $LCCHK NAME
    .IF GE . - H4000
    %DOT==.
    $DOPPAGE <%MAP==MAP>
    SAVEPAGE TEMP
    PAGE LCODE
    $PHEAD NAME
    LDA R7,%MAP
    STA R7,%MAPO
    JMP %DOT
    PAGE TEMP
    .IFF
    $PHEAD NAME
    .ENDC
    .ENDM

.MACRO $PHEAD NAME
$STACKLIMIT
$FCBAREA
$FCBAREA -4
<<$RATE * D10> / D256> / D10
NAME:
.ENDM
```

;OPTION routines

; These macros are functionally identical to
; those defined in GENSUB but work much better.

```
.MACRO $OPT X
    %FLG==TRUE
    $ARGTST'X
    .IF Z %FLG
        '$X == TRUE
    .IFF
        '$X
    .ENDC
.ENDM

.MACRO $NO X
    '$X == FALSE
.ENDM

.MACRO $ARGTST ARG
    .IF B <ARG>
        %FLG==FALSE
    .IFF
        %FLG==TRUE
    .ENDC
.ENDM

.MACRO $ ARG
    .IF NB ARG
        '$ARG
    .ENDC
.ENDM
```

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
PROC.PLR;1 PAGE 8

PAGE 18

```
;INIT headers

.MACRO INITFORK
    .IF NZ . - .%INIT
        .PRINT /INITFORK must immediately follow PROCESS line
    /
    .ENDC
    .=.-2
    PUSH R3           ;Back up over default code
    ;Save our "caller"
    .ENDM

.MACRO ENDINITFORK
    POP R3           ;Get back the return
    JSB R3,(R3)       ;And coreturn
    .ENDM
```

;Other PROCESS-related macros

;The EXIT macro

```
.MACRO EXIT NAME
JMP LOOPMV
$EXIT=.
.ENDM
```

```
.MACRO ENDPROCESS NAME
.IF NZ $EXIT-
    EXIT
.ENDC
.ENDM
```

0000 \$EXIT=0

;Macro to send a message to a process

```
.MACRO POST PROC
.ERROR Message processes have been eliminated
.ENDM
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
PROC.PLR;1 PAGE 10

PAGE 20

179

```
;Macro to allow sleeps inside a routine

.MACRO SLEEP ARG1
    JMP LOOPMV
    .IF NB <ARG1>
        SLP'ARG1
    .ENDC
.ENDM

.MACRO STARTUP ARG1
    SLP'ARG1
.ENDM

.MACRO SLPENTRY NAME
    NAME:
.ENDM
```

;Auxiliary routines (continued)

;Macro to poke a process

```
.MACRO POKE ARG1,ARG2
    .IF B <ARG2>
        POKE R7,ARG1
    .IFF
        .IF DF $'ARG2
            LDA ARG1,#$'ARG2
        .IFF
            LDA ARG1,#ARG2
        .ENDC
        STA ARG1,@PID
    .ENDC
.ENDM
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 22
PROC.PLR;1 PAGE 12

```
;Define process primitives [WAIT, DISMISS, WAKEUP]

.MACRO WAIT STVEC
    SAVE STVEC
    CALL WAIT
    RESTORE STVEC
.ENDM

.MACRO DISMISS STVEC
    SAVE STVEC
    CALL DISMISS
    RESTORE STVEC
.ENDM

.MACRO WAKEUP NAME
    CALL WAKEUP, NAME
.ENDM
```

```
;Define process primitives [FORK, ACTIVATE]

.MACRO FORK NAME
    LDA R2,R1          :Move arg to R2
    CALL ALLOC, SIZE %'NAME :Allocate a block
    ACTIVATE NAME       :Do an activate
    ENDM

.MACRO ACTIVATE NAME
    CALL ACTIVATE, NAME :Let ACTIVATE routine do this
    ENDM
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
IMPOPS.PLR;1 PAGE 1.2

PAGE 24

.insert "STRIP"
.INSRT STRIP

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 25
STRIP.PLR;1 PAGE 1

```
:STRIP.PLR
:Eric Roberts 9-Apr-79
-----
; This file contains the macros for defining
;strips to run under POPS.

:Revision history

:9-Apr-79 ESR:
; Separated this file from the PROC.PLR file to diambiguate
;the notions of STRIP and PROCESS (see HSMIMP #417).
```

```
;Define the STRIP macro
:STRIP <name>.[<args>]
;
:      Defines the beginning of a strip.
:The argument forms are:
;
:      PID = xxx
:      SLOW CLOCK or FAST CLOCK
;
:      The actual strip code is preceded by a
;strip data block which contains some constant
;parameters of the strip, which are given by
;the structure below
;
STRUCT SDBBLK
    WORD -2          ;Start at .-4 offset
    PDBPID: WORD     ;PID level for strip
    PDBCHAIN: PTR(STRIPI) ;Pointer to next timer chain
ENDSTRUCT SDBBLK
```

0000
FFFC
FFFF

```
;Actual STRIP macro

.MACRO STRIP NAME,ARG1,ARG2,ARG3,ARG4,ARG5
$PID == 0
$CLKFLG==0
.MACRO $CLOCK N
$CLKFLG==1
%CCHN==P$'N
P$'N==NAME-4
.ENDM
.IRP ARG,<ARG1,ARG2,ARG3,ARG4,ARG5>
.IIF NB <ARG>, $OPT ARG
.ENDM
$'NAME=$PID
.IF NZ $PID
.IIF MDF $%PLST, $ENTER %PLST,NAME
.IFF
.ERROR STRIP NAME has no declared PID level
.ENC
$LSCHK NAME
.MACRO ENDSTRIP NAME
.IF NZ $EXIT-
.EXIT
.ENC
.ENC
.ENC
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
STRIP.PLR;1 PAGE 4

PAGE 28

187

```
;STRIP macro subroutines

;$.LSCHK
;      Check to see that strip header is in local

.MACRO $.LSCHK NAME
  .IF GE . - H4000
    %DOT==.
    $DOPPAGE <%MAP==MAP>
    SAVEPAGE TEMP
    PAGE LCODE
    $SHEAD NAME
    LDA R7,%MAP
    STA R7,%MAPO
    JMP %DOT
    PAGE TEMP
  .IFF
    $SHEAD NAME
  .ENDC
.ENDM
```

```
;OPTION routines

; These macros are functionally identical to
; those defined in GENSUB but work much better.

.MACRO $OPT X
  %FLG==TRUE
  $ARGTST'X          ;;Assume argument if =
  .IF Z %FLG          ;;Check for blank arg
  '$X == TRUE
  .IFF
  '$X
  .ENDC
.ENDM

.MACRO $NO X
  '$X == FALSE
.ENDM

.MACRO $ARGTST ARG
  .IF B <ARG>
    %FLG==FALSE
  .IFF
    %FLG==TRUE
  .ENDC
.ENDM

.MACRO $ ARG
  .IF NB ARG
    '$ARG
  .ENDC
.ENDM
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
STRIP.PLR;1 PAGE 6

PAGE 30

189

;Auxiliary routines

.MACRO \$SHEAD NAME
\$'NAME
%CCHN
NAME:
.ENDM

.MACRO \$FAST DUMMY
\$CLOCK 1.6
.ENDM

.MACRO \$SLOW DUMMY
\$CLOCK 25.6
.ENDM

```
;Other STRIP-related macros  
  
;The EXIT macro  
  
.MACRO EXIT NAME  
    JMP LOOPMV  
    $EXIT=.  
.ENDM  
  
0000    $EXIT=0
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
STRIP.PLR;1 PAGE 8

PAGE 32

191

```
;Macro to allow sleeps inside a routine

.MACRO SLEEP ARG1
JMP LOOPMV          ;;Act like an EXIT
.IF NB <ARG1>      ;;But allow ENTRY field
    SLP'ARG1
.ENDC
.ENDM

.MACRO STARTUP ARG1
    SLP'ARG1
.ENDM

.MACRO SLPENTRY NAME
    NAME:
.ENDM
```

;Auxiliary routines (continued)

;Macro to poke a strip

```
.MACRO POKE ARG1,ARG2
    .IF B <ARG2>
        POKE R7,ARG1
    .IFF
        .IF DF $'ARG2
            LDA ARG1,#$'ARG2
        .IFF
            LDA ARG1,#ARG2
        .ENDC
        STA ARG1,@PID
    .ENDC
.ENDM
```

```
;Assign page parameters for DDT, V2 pages

0200      ddtlod= H200
80CO      v2st=m2#/pagebc ;V2 page origin

5800      .if df PSE
           ddtvst= H5800   ;DDT and VHA vars
           .iff   ;df PSE
           ddtvst= H5B00   ;DDT vars
           .endc   ;df PSE

;This is the physical DDT page and V2 page

CODEPAGE DDT, PHYSICAL DDTLOD, LIMIT DDTVST

VARSPAGE V2

;Now define the Logical Pages

Defpage V2, org v2st, limit m3

Defpage DDTCode, map code, limit ddtvst, physical ddt, org *
Defpage DDTVars, org ddtvst, limit m1

;Get Stage Common code, DDT, and OPSYS code

.INSERT "STAGEC"
.INSRT STAGEC
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 35
STAGEC.PLR;1 PAGE 1

;STAGEC.PLR - Non-kernel Stages and associated Code

;*** Stage Rely Page non-Kernel Code ***

FC00 0000 PAGE RelCode ;back to common code
page Dummy

;***** reliability page *****

;set up page limits, timeout table, init routine, etc.

40B8 7D50 cpage O.rellod,relvst,relini,reltab,reltyp

40BA 0000

40BC 0000

40BE 0000

;set up dispatches for last few stages

defents s1c00,smm00,sid00,sar00

FC00 0000 page RelCode ;to start of common area

.comnt |
Stage LC - Local memory Checksum

Checksums all of local memory, including the kernel. If
Bad checksum, try to reload (either from other procs or
via external reload)

|
;Maybe check my checksum against rest of system here?
s1c00: ; Local memory Checksum.
repeat
5020 4078 0E10 call cksub,localc,locend ; checksum local memory
5024 02F8
5026 4000
5028 8104 if equal ; cksum good.
502A 4078 0D22 call sclrok ; pass this stage.
502E 900D else ; cksum bad; fix somehow.
5030 4078 0D2E call sfxbad ; inhibit later guys and fix}
5034 810A if equal ; we can fix}
5036 E00E Trap 16,<Stge LC: Local code cksum broken (H/S) - page 36>
5038 7028 40DC lda r2,ihotlm ; initial limit of local
503C 3028 02FA sta r2,hotlim ; fix it in case broken
5040 489F lda r1,=anypro ; destination type
5042 4078 4F50 call tryrld,localc ; try local reload
5046 02F8
endif
endif
5048 90EC endrepeat

;Stage MM - Prototype empty page

| | |
|-----------|------------------------------------|
| 504A 0000 | imemt: O ;intime |
| 504C 9F40 | -<m1//pagebc> ;cksum |
| 504E 60C0 | m1#/pagebc ;tlimit |
| 5050 0000 | O ;pginit |
| 5052 0000 | O ;topntr |
| 5054 0000 | O ;type4k |
| 000C | 1imemt=-imemt ;length of this area |

```
;Stage MM - Common Memory Management

smm00:

repeat
5056 48D0    lda r5,=0          ;current type, cmap index
5058 7048 40FE lda r4,bsmapm ;low bus
505C 7038 4100 lda r3,bsmapm+words ;high bus(ses)
5060 7028 40A6 lda r2,memtot ;total pages in machine

repeat
5064 4813    lda r1,r3          ;assume higher bus
5066 9904    if odd } r5 < ncodep ;none, or doing code
5068 7658 4104
506C 8204
506E 4814    lda r1,r4          ;so try lower bus
5070 8902    if odd
5072 4813    lda r1,r3          ;none, must use higher
      endif
      endif
      ;lda r1,r1          ;set success, test r1
      ;ifnot odd
      ;call memst          ;page to try at all?
      ;this page exist?
5074 9903
5076 4078 0E8A
507A 9A2E    if success         ;exists or no page (bum})
507C 4078 5112 call smmsearch   ;look for the needed page
5080 9A04    if fail } r1 <> cmap(r5) ;not perfect
5082 761D 5E40
5086 9109
5088 301D 00B0 sta r1,lmap(r5) ;so that loader knows where
508C 4078 0D2E call sfxbad     ;must fix now
5090 8140    break ifnot equal ;no consensus
5092 4078 5290 call simfix     ;repair as required
5096 9A3D    break if fail    ;couldn't do it
```

```
5098 201D 00B0      endif
509C 990D      sta r1,lmap(r5)+      ;set up our copy of maps
509E 3018 FC06      ifnot odd          ;got a real page
50A2 7078 A08E      sta r1,%map3       ;get to it locked
50A6 9AFE          lock r7,s1flk       ;now fix up SLFLK with type
50A8 4B78 FFO0      and r7,= HFFOO      ;just map, 100} bits
50AC 4C7D FFFE      ior r7,=-2(r5)     ;add type too
50B0 3078 A08E      unlock r7,s1flk      ;and restore SLFLK
50B4 49A1          sub r2,=1           ;used a page
50B6 7658 410A      endif
50BA 810E          if r5 = novarp      ;filled table
50BC 7028 4100      lda r2,bsmapm+words   ;end of lower bus
50C0 4814          lda r1,r4          ;;odd ;next unused page
50C2 4078 5380      call smmsmash      ;kill its remaining stuff
50C6 4828 8000      lda r2,=nmseg* H8* H200
50CA 4813          lda r1,r3          ;;odd ;also for higher bus(es)
50CC 4078 5380      call smmsmash
50D0 4078 0D22      call sc1rok         ;succeed stage
50D4 901E          break
50D6 901E          endif
50D8 901E          endif
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 40
STAGEC.PLR;1 PAGE 5

```
50D6 4E18 FFFF      if r1 <> -1           ;used a real page
50DA 9113
50DC 4E13      if r1 = r3           ;just used highest page
50DE 8108
50EO 4A38 0200    add r3,= H200       ;next higher page
50E4 4E38 8000    if r3 = =nmseg+ D8* H200   ;last page
50E8 8103
50EA 4838 FFFF    lda r3,=-1
50EE 4E14      endif
50FO 8108      if r1 = r4           ;just used lowest page
50F2 4A48 0200    add r4,= H200       ;get the next higher
50FG 7648 4100    if r4 = bsmapm+words  ;end of low bus
50FA 8103
50FC 4848 FFFF    lda r4,=-1
5100 4E34      endif
5102 8106      if r3 = r4           ;was it last? (both = -1)
5104 7658 4110    if r5 < nrequp     ;too few found?
5108 8203
510A EEOF      Trap 17,<;Stage MM: Not Enough Memory (H) - page 40>
510C 9002      break
510E 90AB      endif
5110 90A3      endrepeat
5110 90A3      endrepeat
```

```
;Stage MM Subroutines

;SMMSEARCH - check proper page for type, checksum, quits.
;If this page isn't needed or wanted, set R1 to -1, return.
;If not okay, search for a good copy, a spare, and a free page

5112 1076    routine smmsearch, arg r1/r3-r5, result r1, uses m1

5114 4E18 FFFF  if r1 <> =-1           ;no bother if no page to use
5118 9160
511A 48F1      lda r7,=1             ;assume no spare initially
511C 7658 4104  if r5 >= ncodep
5120 9220
5122 7658 4106  if r5 < nsparp       ;doing a spare?
5126 820E
5128 4875      lda r7,r5
512A 7178 4104  sub r7,ncodep
512E 707F 00B0  lda r7,1map(r7)     ;see if there's a main page
5132 9904      if odd } r2 <= ndvars ;no page or too few free
5134 7628 4114
5138 9C04
513A 4818 FFFF  lda r1,=-1         ;so no allocate any
513E 6006      return
                endif
5140 9010      else
5142 7078 5E3E  if buff1g & r5 > nrequp ;check for M/I configuration
5146 9A0C
5148 7658 4110
514C 8C09
514E 4E14      if r1 = r4 & mysegs+6 ;this is O bus & 6000 exists
5150 8107
5152 7078 01B2
5156 9A04
5158 4818 FFFF  lda r1,=-1         ;yes, no allocate
                return
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
STAGEC.PLR;1 PAGE 6.1

PAGE 42

515C 90F1

```
        endif
        endif
515E 48F1      lda r7.=1          ;no spare yet
        endif
        endif
5160 3078 01C4 sta r7.smmspa    ;remember main as spare
5164 48F1      lda r7.=1          ;no page found yet
5166 3078 01C2 sta r7.smmok     ;nor free page
516A 3078 01C6 sta r7.smmfre    ;worst type for free page
516E 48FO      lda r7.=0          ;worst type for free page
5170 3078 01C8 sta r7.smmfty
```

```
5174 4078 51DA    call smmccheck      ;check default page
5178 8A21    if fail           ;all is not well
517A 1016    save r1-r2       ;want these later
517C 1026
517E 7028 4100   lda r2,bsmapm+words  ;limit of low bus
5182 4814    lda r1,r4        ;;odd      ;now look at rest of pages
5184 4078 5272   call smmscan      ;check low pages
5188 4828 8000   lda r2,=nmseg* H8* H200 ;last page in system.+ 200
518C 4813    lda r1,r3        ;;odd      ;now for high bus(es)
518E 4078 5272   call smmscan
5192 6026    restore r1-r2
5194 6016
5196 7078 01C2   lda r7,smmok      ;found page if any
519A 7378 01C4   and r7,smmspa    ;and found spare
519E 890B    if odd           ;no page nowhere
51A0 48F8    lda r7,=1mmbas    ;for list of categories
      repeat
      until r5 >= smmbas(-r7) ;where we are
51A2 565F 4102   endrepeat
51A6 92FE
51A8 765F 410C   if r5 > smmins(r7) ;got min this type?
51AC 8C04
51AE 4818 FFFF   lda r1,=-1      ;yes, so all's well
51B2 90C6    return
      endif
      endif
      fail return      ;didn't succeed anyway
51B4 6076
51B6 4FF0
51B8 4007
      endif
51BA 7078 01C4   lda r7,smmspa    ;target a spare for a good page?
51BE 990D    ifnot odd      ;yes
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 44
STAGEC.PLR;1 PAGE 7.1

```
51C0 3078 FCO2    sta r7,%map1      ;get to main page
51C4 7078 60B6    lda r7,m1#cksum  ;its checksum
51C8 3018 FCO2    sta r1,%map1      ;back to spare
51CC 7178 4104    sub r7,ncodep    ;this should be spare's checksum
51D0 7678 60B6    if r7 <> m1#cksum
51D4 9102
51D6 E025        Trap 45,<Spare pg cksum differs (H/S) - page 44>
                  endif
                  endif
                  endif
51D8 90B3        endroutine
```

```
:SMMCHECK - check given page's type, checksum, and variables.

51DA 1076    routine smmcheck, arg r1, arg r5, local r2-r4      ,
51DC 1026
51DE 1036
51EO 1046

51E2 4875    lda r7,r5          :type we want
51E4 7658 4104 if r5 < ncodep   ;looking for code?
51E8 8203
51EA 7278 4104 add r7,ncodep   ;yes, accept spare
      endif
51EE 4828 FFFF lda r2,=-1      ;type if QUIT
51F2 7028 60BE lda r2,m1/type4k ;type of target page
51F6 8026      if nquit & r2 = r5 } r2 = r7 ;right type or spare?

51F8 4E25
51FA 9103
51FC 4E27
51FE 8122
5200 4ED0      if r5 <> =reltyp } r1 >= maprel      ;ignore bad com kernels
5202 8104
5204 7618 00B0
5208 921B
520A 1056      save r5
520C 4078 0E10 call cksub,m1/cksum,m2      ;do its checksum
5210 60B6
5212 8000
5214 6056      restore r5
5216 8114      if equal
5218 4078 539E call smmqch      ;check page for quits
521C 7658 60BE if r5 = m1/type4k ;find perfect page?
5220 8107
5222 3018 01C2 sta r1,smmok      ;remember it
      return
```

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 46
STAGEC.PLR;1 PAGE 8.1

```
5226 6046
5228 6036
522A 6026
522C 6006
      endif
522E 3018 01C4     sta r1,smmspa      ;no, remember spare
5232 6046     fail return
5234 6036
5236 6026
5238 6076
523A 4FF0
523C 4007
      endif
      endif
      lda r2,=-1      ;signal bad type
endif
```

```
5242 7078 01C8 lda r7,smmfty :type of free page
5246 9B15 ifnot minus :no free page yet
5248 7628 4104 if r2 >= ncodep & r2 < nsparp :spare type?
524C 9206
524E 7628 4106
5252 8203
5254 7128 4104 sub r2,ncodep :yes, count it like code
      endif
5258 4E25 if r2 < r5 :type we've already seen?
525A 8203
525C 4828 FFFF lda r2,=-1 :yes, can smash
      endif
5260 4EA0 if r2 < =0 } r2 > r7 ;smmfty :better free page?
5262 9203
5264 4E27
5266 8C05
5268 3028 01C8 sta r2,smmfty :save its type
526C 3018 01C6 sta r1,smmfre :and page
      endif
      endif
5270 90E1 fail return

endroutine smmcheck

:SMMSCAN
:Scan a range of pages, checking for useful types
:Calls SMMCHECK to do all the work
:R1,R2 are limits of search

5272 1076 routine smmscan,arg r1-r2,arg r5

5274 4811 lda r1,r1 :got a range to search?
5276 990C ifnot odd :if any to check
      repeat
5278 4078 0E8A call memtst :this page there?
      if success :yes, so check it out
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 48
STAGEC.PLR;1 PAGE 9.1

```
527C 9A05
527E 4078 51DA      call smmcheck
5282 4078 07DC      call wsleep      ;rest occasionally
                     endif
5286 4A18 0200      add r1,= H200
528A 4E12            until r1 = r2
528C 81F6
                     endrepeat
                     endif

528E 6006      endroutine smmscan
```

```
;SMMFIX - Stage MM routine to move memory around as needed
;Called after achieving consensus. Movement is governed by the
;local variables SMMOK, SMMSPA, and SMMFRE. If the missing page
;is a code page and there is no good copy or spare, trigger a
;reload of that page by calling TRYRLD. Repair CMAP when done.
```

```
5290 1076      routine smmfix, arg r1, arg r5, local <r1-r2>, uses <m1-m2>
5292 1016
5294 1026

5296 4811      lda r1,r1
5298 9955      ifnot odd
529A 0888          inh .L4
529C 7028 01C2      lda r2,smmok
52A0 8903      if odd
52A2 7028 01C4      lda r2,smmspa
52A6 4E12      endif
52A8 9139      if r1 <> r2
52AA 7078 01C6      lda r7,smmfre
52AE 990A      if even & r1 <> r7
52B0 4E17
52B2 9108
52B4 3078 FCO2      sta r7,%map1
52B8 3018 FCO4      sta r1,%map2
52BC 4078 534C      call smmcop,O//--intime
52C0 1F4C      endif
```

IMP Operating System STAGEC.PLR;1

PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 50
PAGE 11

```
52C2 3018 FCO2    sta r1.%map1          ;target is now dest
52C6 4822    lda r2,r2  ;smmsok smmspa
52C8 9907    ifnot odd           ;got a page to copy
52CA 3028 FCO4    sta r2.%map2          ;this is source page
52CE 4078 534C    call smmcop.O//intime  ;and move it
52D2 1F4C
52D4 9023    else
52D6 7658 4106   if r5 < nsparp        ;must reload us
52DA 8213
52DC 4078 OF88    call fixjif
52E0 4815    lda r1,r5          ;type to load
52E2 48FO    lda r7,=0
52E4 3078 60B4    sta r7,m1#intime
52E8 707D 40DE    lda r7,l1mtab(r5)    ;fix up target
52EC 3078 60B8    sta r7,m1#tlimit
52FO 4078 4F50    call tryr1d,m1#cksum  ;start external reload
52F4 60B6
52F6 6026    fail return        ;force MM to quit now
52F8 6016
52FA 6076
52FC 4FF0
52FE 4007
5300 3018 FCO4    endif
5304 3018 FC06    sta r1.%map2          ;this is also source
5308 4078 534C    sta r1.%map3          ;and will be cleared
530C 5F4C    call smmcop,m3-m1+<0//intime>  ;by using map3 for compare
530E 48AC    1da r2,=l1memt        ;set up the type header, etc.
5310 507A 504A    repeat
5314 307A 60B4    1da r7,imemt(-r2)
5318 88FC    sta r7,m1#intime(r2)    until loop
                                endrepeat
                                endif
                                endif
```

```
531A 3018 FCO2    sta r1,%map1          :get to target page
531E 4ED0          if r5 = =reltyp      ;(necc?)
5320 8105
5322 3018 FC00    sta r1,%map0          :go run there}
5326 3018 40B0    sta r1,comrel
endif
532A 4875          lda r7,r5
532C 7178 60BE    sub r7,m1#type4k      :adjust type if needed
5330 3278 60BE    addm r7,m1#type4k
5334 3178 60B6    subm r7,m1#cksum
Trap 3,<;Ctd STAGE mem mgmnt--OK if Strtup/Rstrt (H) - page 51>
5338 E003
51
533A 4078 OF88    call fixjif
533E 4078 07DC    call wsleep          ;now take a rest, fix maps
endif
5342 301D 5E40    sta r1,cmap(r5)      ;finally fix cmap
5346 6026          endroutine
```



```
;SMMCOP - routine to copy or clear a page. Clears if the
;in line argument adds an offset from map 1 to map 3. Copies
;from map 2 to map 1 or 3.

534C 1076      routine smmcop, local <r1-r2>, inline r2
534E 1016
5350 1026
5352 6027
5354 307E 0004

5358 4818 1F4C  lda r1,=0//intime          ;total to copy
repeat
535C 5079 80B4  lda r7,m2//intime(-r1)    ;source word
5360 8001        ifnot quit                ;ignore quits
endif
5362 107A 60B4  sta r7,m1//intime(-r2)    ;store it away
5366 767A 60B4  cmp r7,m1//intime(r2)     ;check (and clear) word
536A 8002        if quit } NEqual         ;trouble
536C 9105
536E 7048 0064  lda r4,quitad            ;Note failing location
; Trap 57,<Stage MM: Copy/clear failed (H/S) - page 52>
; Disable the following, as MOS memories fail on read-after-lock
; lda r1,m1#slfptr           ;the bad page
; call memdis               ;stop using the page
; jsb r4,wst                ;and restart
5372 4048 0A1C  endif
5376 4E90        until r1 = =0
5378 81F2        endrepeat

537A 6026        endroutine smmcop
537C 6016
537E 6006

;SMMSMASH
;Clobber the TYPE4K words on all the rest of the pages
;within the given range. R1-R2 specify range

routine smmsmash,arg r1-r2
```

5380 1076

```
5382 4811    lda r1,r1          ;check range start
5384 990C    ifnot odd           ;some to smash
               repeat
5386 4078 0E8A    call memtst      ;this page exist?
538A 9A05    if success          ;yes
538C 4878 FFFF    lda r7,=-1      ;invalid type
5390 3078 60BE    sta r7,m1#type4k  ;clobber page
               endif
5394 4A18 0200    add r1,= H200    ;next page
5398 4E12    until r1 = r2
539A 81F6

               endrepeat
               endif

539C 6006    endroutine smmsmash
```

```
.comnt |
SMMQCH - check unchecksummed part of page in r1 for
quits. r2 points at first word to check.

Takes page in r2 and scans all of the part after the checksummed
region specified by the cksum block for quits. Fix if find any.
Disable the bad page and restart if unrecoverable error.

|
539E 1076      routine smmqch,arg r1-r2,uses r2
53AO 4F28 1FFE      repeat while r2 .bit. =packm
53A4 9AOE
53A6 4078 07DC      call wsleep
53AA 3018 FC02      sta r1.%map1
repeat
53AE 6072          lda r7,(r2)+           ;load a word
53BO 8002          if quit
53B2 9003
53B4 4078 53D0      call smmqfx           ;diagnose QUIT
endif
53B8 4F28 003E      while r2 .bit. = H3E
53BC 8AF9          endrepeat
53BE 90F1          endrepeat
53CO 4828 60B4      lda r2,=m1/intime
53C4 6072          lda r7,(r2)+           ;check INTINE too}
53C6 8002          if quit
53C8 9003
53CA 4078 53D0      call smmqfx           ;look at its QUIT
endif
53CE 6006          endroutine smmqch

.comnt |
```

SMMWRD - check one word that got QUIT and fix.

Call with r1 current page, r2 pointing 2 past word. returns if
quit fixable; disable memory and restart if couldn't fix quit.

```
53D0 1076      routine smmqfx,arg r1-r2

53D2 48F0      lda r7.=0
53D4 1072      sta r7,(-r2)
53D6 8006      ifnot quit
53D8 6772      tst r7,(r2)+
53DA 8004      ifnot quit
53DC 3078 60B4 ; Trap 46,<;fixed bad mem parity (H) - page 55>
53E0 6006      sta r7,m1#intime
                return
                endif
                endif
                Trap 47,<;Solid mem parity err *** CALL BBN MAINT *** (H) - page 55>

53E2 EO27
page 55
53E4 4048 OA1C jsb r4,wst           ;and restart

53E8 90FC      endroutine smmqfx
```

.commt |
Stage ID Discover the I/O
Look for I/O interfaces in common. Build a bit-table
of what ones exist in useio, subject to removal in
iokill. Don't use a device if its PID isn't there.
prohng gives processors to never run (they hang here).

5424 4078 07DC

endif
endif
until r5 = =0 : until done this segment.

5428 4ED0
542A 81FO

endrepeat
endif

542C 4078 07DC

call wsleep : catch up on sleep.

```
5430 753A 5E98      eor r3,useio(r2)          ; r3 = our view of useio.
5434 9A20          ifnot zero                ; we match system?
5436 773A 5E98      if r3 .bit. useio(r2)       ; no, our useio <> system useio.
543A 9A03          call sbad                 ; system see things we don't?
543C 4078 0D3E      endif
5440 7078 00A8      lda r7,procbt           ; my vote
5444 7478 5E96      ior r7,iofix            ; get fixit word
5448 3078 5E96      sta r7,iofix            ; include my vote in fixit
544C 7048 5E90      lda r4,iocon            ; consensus word
5450 4B74          and r7,r4               ; ignore dead voters
5452 4B4C FFFF      repeat
5456 4B7F FFFF      and r4,=-1(r4)         ; calc. how majority votes.
545A 9A04          and r7,=-1(r7)         ; remove one voter...
545C 4B4C FFFF      until zero             ; and one vote to fix
5460 8AF9          and r4,=-1(r4)         ; no more votes to fix
5462 4EC0          until zero             ; now another pair of voters
5464 8107          endrepeat              ; no more voters left.

5462 4EC0          if r4 = =0            ; Majority votes to fix
5464 8107          sta r4,sidflg           ; signal I/O config change
5466 3048 620A      sta r4,iofix            ; restart fixit to stop race
546A 3048 5E96      eorm r3,useio(r2)       ; give the system our picture.
546E 353A 5E98      endif
5472 9011          break                  ; restart this stage
5474 49A2          endif

5476 8B0E          sub r2,=2              ; select next i/o segment
5478 4078 0D90      if minus              ; none left
547C 7078 00A8      call sclear            ; take our bit out of fixit.
547E 49A2          1da r7,procbt          ; do i exist?
```

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 59
STAGEC.PLR;1 PAGE 15.1

```
5480 7778 40C8      if r7 .bit. prohng    ; me disabled past here?  
5484 9A04  
5486 4078 0D3E      call sbad           ; Yes, hang later stages.  
548A 9003      else                ; me enabled, so be happy.  
548C 4078 0D50      call sokay          ; let next stage run .  
5490 9002      endif  
                 break  
                 endif  
  
5492 90AD      endrepeat  
  
5494 90AB      endrepeat
```

.comnt |
Stage AR - find any un-running processors and get them
loaded and Started
ampman, ampsys and ampty are the current state of defined
amputees. the abus word gives processor buses which
should be amputated and the aproc words give the processors
that should not be started up
prior is ior of 3 aproc words, busior of abus words

|
sar00:
repeat
5496 4078 07DC call wsleep
549A 7038 5DBC lda r3,bltst ;blt busy?
549E 9938 ifnot odd ;;bltact
54A0 7028 5E34 lda r2,locfix ;unhappy procs
54A4 9A07 ifnot zero ;are there any?
54A6 4078 55C8 call locblt,bltact,locfix,memcon
54AA 0001
54AC 5E34
54AE 5E36
54B0 902F else
: look to see if any processors need starting from time to time
: do this if BLT process is idle and no one hung in stage LC
54B2 7028 5EA0 lda r2,pptime ;time for processor stuff?
54B6 7038 00A0 lda r3,stime ;system time
54BA 4923 sub r2,r3
54BC 4F28 F000 tst r2,= HF000
54C0 9A27 ifnot zero ;time now?
54C2 7078 0160 lda r7,bltmvc ;do nothing if debug-halted
54C6 8906 if nodd } debugm .nbit. procbt ;okay to restart
54C8 7078 030C
54CC 7778 00A8
54D0 8A1F
54D2 4A38 0078 add r3,=prrate ;set time ahead now

```
54D6 3038 5EA0      sta r3,pftime
54DA 7048 5D66      lda r4,proior
54DE 7018 4096      lda r1,segcon      ;who's already running
54E2 4C41           ior r4,r1        ;ignore the running guys
54E4 7448 40C6      ior r4,prokil     ;see who's left
54E8 4D48 FFFF      eor r4,=-1       ;and who exists
54EC 7348 5D64      and r4,procex    ;anybody?
54F0 9A0F           ifnot zero
54F2 4F48 5555      tst r4,= H5555   ;any even to do?
54F6 9A05           ifnot zero
54F8 A291           sll r1,1        ;buddies of live processors
54FA 4C18 5555      ior r1,= H5555   ;plus even ones
54FE 4B41           and r4,r1       ;select these only
                           endif
5500 3048 017E      sta r4,temp1    ;these need restarting
5504 4078 55C8      call locbit,b1tact}b1trun,temp1,rckcon
5508 0401
550A 017E
550C 5E28
                           endif
                           endif
                           endif
endif
```

```
550E 4878 DEAD      set lstack-lstkln-lstkln = =stkpass ;init stack password
5512 3078 0288
5516 4078 5624      call sarwdg          ;check application watchdog
551A 4078 OFOE      call sarpoll         ;poll various init routines
551E 9ABC           next if fail       ;hanging on some fix
5520 4078 55AO      call sarpcnt        ;enough procs running?
5524 9AB9           next if fail       ;nope - hang here
5526 4078 0D50      call sokay          ;on to next stage
552A 7018 0056      lda r1,quitrt     ifnot zero      ;any recent quit retries?
552E 9A08           lda r2,rtryad     ;R2 reports QUIT address
5530 7028 0058      lda r3,rtrypc     ;R3 reports QUIT PC .
5534 7038 005A
5538 E02C           Trap 54,<;Quit retries ok on 2nd mem ref (H) - page 62>
553A 3118 0056      subm r1,quitrt
553E 7018 005C      endif
5542 9A07           lda r1,qqhct      ifnot zero      ;any QUITs in QUIT handler?
5544 7028 005E      lda r2,qqhad     ;R2 reports QUIT address
5548 7038 0060      lda r3,qqhpc     ;R3 reports QUIT PC
554C 3118 005C      ; Trap 52,<;QUITs in hndlir (MEM PROB-- CALL MAINT) (H) - page 62>
5550 7018 00AA      subm r1,qqhct
5554 990F           endif
5556 7018 0050      lda r1,procno    ifnot odd       ;should I have jiffies?
555A 9A03           lda r1,clokrt    ifnot zero      ;any RTC re-reads?
555C 3118 0050      ; Trap 55,<;RTC read retries succeeded (H) - page 62>
5560 4078 05D4      subm r1,clokrt
5564 7128 01A6      call rclock       ;read the clock, reliably
5568 3118 0050      sub r2,jtime     ;when jiffy last looked
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE ,63
STAGEC.PLR;1 PAGE 17.1

5568 9B05 ifnot minus :jiffy didn't nail us?
556A 4E28 2710 if r2 > = D10000 :1 second gone
556E 8C02
5570 E00A Trap 12,<;Jiffy clock stopped (H) - page 63>
endif
endif
endif

5572 7048 00D0 lda r4,mapvar :get application vars page
5576 3048 FCO2 sta r4,%map1
557A 7058 0100 lda r5,locipt :local info to common
repeat
557E 4958 0010 sub r5,=ibuf1*words ;most recently used buffer
5582 4B58 0030 and r5,=<nibus-1>*ibuf1*words ;keep within buffer
5586 702D 0102 lda r2,loci11(r5) ;Trap here?
558A 9AO8 until zero ;no more in local
558C 352D 0102 eorm r2,loci11(r5) ;removing this one
5590 481D 0104 lda r1,=loci11+words(r5) ;point to regs
5594 4078 0734 call i11cnt
5598 90F3 endrepeat

559A 4008 5496 endrepeat

PAGE LCode :back to local

OFOE 1076 ;Stage AR Init Routine Poller
OF10 4890 routine sarpoll
1da r1,=0 ;for all code
repeat
OF12 6029 00BO 1da r2,1map(r1)+ ;next map setting
OF16 9931 ifnot odd ;page exists
OF18 3028 FCOO sta r2,%map0 ;get to it
OF1C 7078 40BA 1da r7,pinit ;pointer to init routine
OF20 9A2C ifnot zero ;there is one
OF22 1016 push r1 ;save 1map index
repeat
OF24 1026 push r2 ;save map setting
OF26 0201 rst %e ;done condition
OF28 4077 call (r7) ;coreturn to routine
OF2A 8A1D if fail ;wants consensus
OF2C 4817 1da r1,r7 ;remember coreturn
OF2E 7078 00BO setmap m0,maprel ;need rely page for consensi
OF32 3078 FCOO
OF36 6026 pop r2 ;remember map setting for later
OF38 6076 pop r7 ;forget 1map index
OF3A 4078 0D2E call sfxbad
OF3E 8110 if equal ;got it
OF40 3028 FCOO sta r2,%map0 ;allow long init routine
OF44 0888 inh .L4
OF46 4876 1da r7,sp
OF48 4868 02C8 1da sp,=s1stack ;use system local stack
OF4C 1076 push r7
OF4E 4071 call (r1)

```
OF50 6066          pop sp           ;back to AR stack
OF52 4078 OF88      call fixjif
OF56 7078 OOBO      setmap m0,maprel
OF5A 3078 FC00
                     endif
OF5E 6076          fail return   ;start over either way
OF60 4FF0
OF62 4007
                     endif
OF64 6026          pop r2           ;map setting
OF66 9108          until equal    ;this routine done?
OF68 1076          push r7           ;save coreturn
OF6A 4078 07DC      call wsleep
OF6E 6076          pop r7           ;(restores map.0 = maprel)
OF70 3028 FC00      sta r2,%map0  ;fetch coreturn again
OF74 90D8          endrepeat
OF76 6016          pop r1           ;page to call on
                     endif
                     restore lmap index
OF78 7078 OOBO      setmap m0,maprel
OF7C 3078 FC00
OF80 7618 4104      until r1 = ncodep ;returning to rely
OF84 81C7
                     endrepeat
OF86 6006          endroutine sarpoll
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 66
STAGEC.PLR;1 PAGE 19

;Routine to fix up jiffy timing and reenable level 4

OF88 1076 routine fixjif
OF8A 70F8 01A4 lda r7,@1clock
OF8E 3078 01A2 sta r7,ltime
OF92 0808 enb .L4
OF94 6006 endroutine fixjif

FC00 0000 page RelCode

;move this into kernel params in new IMPLOD }}}

559E 0000 minpro: 0 ;patched by FINSTAGE

;SARPCNT
;count procs that are running, fail if too few

55AO 1076 routine sarpcnt, uses r1

55A2 4078 0D90 call sclear
55A6 7078 5EA2 lda r7,inifix ;hanging procs don't count
55AA 4D78 FFFF eor r7,=-1
55AE 7378 5EA4 and r7,inicon ;and procs must be in this stage
55B2 7018 559E lda r1,minproc ;how many required
repeat
55B6 4991 sub r1,=1 ;count procs now
55B8 8A02 if zero ;count one
55BA 6006 return ;got enough
endif
55BC 4B7F FFFF and r7,=-1(r7) ;all's well
55CO 8AFB until zero ;count off a running proc
55C2 6076 endrepeat
55C4 4FF0 fail return
55C6 4007

endroutine sarpcnt

.comnt |
Subroutine to try to set up a BLT local-to-local transfer
In-line args: blt state, dest proc mask, and source proc mask
Grabs blt process if not busy and starts a transfer
Assumes: use reload buffer on rely page. Limits = local limits.
Source type = dest type = all processors (anypro).

|
55C8 6037 routine locblt,inline r3,indirect r1,indirect r2,uses r5
55CA 6097
55CC 60A7
55CE 1076

55D0 7078 BDBA lock bltlok ;lock blt params
55D4 9AFE
55D6 7078 5DBC lda r7,bltst ;blt busy?
55DA 9922 ifnot odd
55DC 4858 02F8 lda r5,=localc ;starting address to load
55E0 3058 5DC0 sta r5,bltadd ;remember addr for outside
55E4 48FF lda r7,=anypro ;this is local-local
55E6 3078 5DD6 sta r7,bltsty ;source type
55EA 3078 5DDA sta r7,bltdty ;dest type
55EE 3038 5DBC sta r3,bltst ;new blt state
55F2 7078 02FA lda r7,hotlim ;get finish
55F6 4975 sub r7,r5 ;length of transfer
55F8 3078 5DC2 sta r7,bltsiz ;total transfer
55FC 3028 5DD8 sta r2,bltspm ;source procs
5600 4D28 FFFF eor r2,=-1 ;not source procs
5604 4B12 and r1,r2 ;only these get core
5606 3018 5DDE sta r1,bltdpi ;dest procs
560A 4878 5DE0 lda r7,=bltbuf
560E 3078 5DC8 sta r7,bltbfa ;buffer address
5612 7078 00AO lda r7,stime
5616 4A78 0078 add r7,=bltrat
561A 3078 5DBE sta r7,bltto ;set timeout
endif

561E 3008 BDBA unlock bltlok
endroutine locblt

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
STAGEC.PLR;1 PAGE 20.1

PAGE 69

5622 6006

```
;check system watchdog

;define watchdog variables
    Page Vars
625A      wdgtim: .blkw 2           ;lo-hi order sytime at which to trigger

FC00 0000      page RelCode
5624 1076      routine sarwdg, uses r1-r2
5626 7078 00A8      if procbt .nbit. debugm   ;don't check if debugging
562A 7778 030C
562E 8A10
5630 7018 625A      lda r1,wdgtim        ;get lo order current time
5634 9A0D      ifnot zero
5636 7028 625C      lda r2,wdgtim+words
563A 7118 00A0      sub r1,stime        ;see if time's up
563E 9402      ifnot carry
5640 49A1      sub r2,=1
                endif
5642 7128 00A2      sub r2,stim2
5646 8B04      if minus          ;time got ahead too far}
5648 E050      Trap 120,<;S/W watchdog timer expired>} - page 70>
564A 4048 0A08      jsb r4,ws          ;restart just for now
.comnt |
    lda r1,# HDOOB        ;calling card
    sta r1,m0#cksum      ;:maprel;smash rel page
    lda r2,maprel+ H10    ;:nsparp
    ifnot odd            ;have to get spare too
        sta r2,%map2
        sta r1,m2#cksum
    endif
    |
    endif
    endif
    endif
564E 6006      endroutine sarwdg
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 71
IMPOPS.PLR;1 PAGE 1.4

.INSERT "DDT"
.INSRT DDT

.comnt |

Pluribus IMP Minimal DDT
Written for IMP 1110, April 1980- Andy Huang
Revised for IMP 1112, 28 April 80- Andy Huang

1. Numbers

The new DDT works solely in hexadecimal. The radix commands <esc> 0, H, D and the radix specifiers "'", "}" , "." have all been removed or applied to other purposes. Another character (\$) is now used to specify decimal input rather than hexadecimal and is the only exception to the hexadecimal rule.

2. Commands

- x,y/ This is the basic examine command of DDT to return the contents of a memory location. There are two cases of this command depending on the value of Y. If Y is a local address (ie Y is less than 4000) then X is the mask of processors whose memory is to be examined (this means that the answer returned will be from one of the processors specified by the mask X). A special case is a negative mask value and sets the processor mask to be all those that are known to exist to stage BD. In the other case, when Y is greater than 4000 and therefore a reference to common memory, a then specifies the map setting to use in the reference. In this case a can be either a logical page (x < 200) or a physical page (x is odd or x > 200).
- x/ This is a simplified case of the above and does an examine of the address X using the last processor mask specified if X is a local address or the last map specified if X is a common memory address.
- x,y<cr> Carriage return is used to insert new values into memory and close the location currently examined. X and Y will be inserted into the current location and the next location respectively if the current location is still open and then the location is closed. A location is open when it has been examined but not closed with a carriage return or linefeed.
- x<cr> This is just the one argument form of the above and stores just one number into memory

<cr> This is the no argument form of the above and stores nothing but closes the location.

<lf> Linefeed closes the current location and examine the next location

x y Space adds two arguments together and the result becomes one argument. eg "x y/" will open the location at x+y.

 Delete (=rubout) will zero all current input and will restore DDT to the state it was at the last typeout.

.

Dot has the value of the current address and can be used instead of typing the current address. It can no longer be used to specify a decimal number.

\$ Dollar sign is now used to say that the number just typed in is a decimal number.

3. Typeout

All typeout from DDT is four digit hexadecimal.

Examine Formats

a,b/x y

This is the usual format of an examine. If a,b is a local memory reference, then only x will be printed. Note that the processor that did the reference is no longer specified. If a,b is a common memory address then x and y are the contents of the main and spare pages if they differ. If some kind of error occurs in the reference then the x and y are replaced by the appropriate error messages for the corresponding pages.

Error Messages

There are two formats of error messages for the two situations where an error can occur. A system wide error can occur causing Stage to put the system into the stand-alone DDT mode (if enabled by DEBUGM). In this situation the bit of the processor reporting the error is printed first followed by the error followed by the location of the error. The second type of error is a store or read error as a result of a DDT reference. In these cases the error is typed first followed by a number specifying the mask of processors that failed the reference.

Errors

- | | |
|------|--|
| QUIT | The location referenced by DDT resulted in a QUIT or an unexpected QUIT occurred in the running of the system. In the system QUIT the address returned by DDT is not the address of the instruction producing the QUIT but the address of the F001 Trap specifying the quit. The location of the QUIT can be found in the snapshot area in the Stage variables area. |
| NX | BLT returned a non-existent memory code as a result of some DDT reference. |
| FRMT | BLT returned a format error in response to some DDT reference. This usually means that DDT had set up parameters improperly for BLT or that a reference to a nonexistent processor was made. |
| TO | Timeout- BLT took too long to complete a reference and aborted |

IL An attempt was made to execute a non-instruction

FADE The halt all processors trap was encountered in the running of the
system

```
.stitle DDT and TTY code
0010 .radix H10

;Teletype interface definitions
page Dummy
0000 .=0
0000 psbsta: .blkw 1 ;status register
0001 psbfree=1 ;PSB available
0004 psbovr=4 ;overrun
0008 psbbrk= H8 ;got a break
0002 .blkw 2
0006 psbctl: .blkw 1 ;control register
0001 psbact=1 ;psb active bit
0002 psbout=2 ;psb in output mode
0004 psbint=4 ;psb should cause interrupts
0008 psbech= H8 ;psb should echo typein (half-duplex)

0008 psbdat: .blkw 1 ;data register
;vistar cursor control characters:
000B cclear= 013 ;clear current line
0019 cright= 031 ;move cursor right
001C cup= 034 ;move cursor up line
001D cdown= 035 ;move cursor down line

FA00 psbone= HFA00
EA00 psbtwo= HEA00

;some useful ascii constants
002C asiccomma= 054
0020 ascifspce= 040 ;space
000A ascilf= 012 ;linefeed
000D ascicr= 015 ;carriage return
000C asciff= 014 ;formfeed

page DDTVars
;ddt stack vars
stack dd, D15
5B00
5B1E
5B1E
5B1E ddtsp: .blkw 1 ;ddt stack pointer when sleeping
5B20 dsyssp: .blkw 1 ;system called ddt with this in sp
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 77
DDT.PLR;1 PAGE 3.1 DDT and TTY code

5B22 ;tty stack vars
5B36 stack tt, D10
5B36
5B36
5B36 ttysp: .blkw 1 :tty's stack saved pointer
5B38 ttsyssp: .blkw 1 :system called tty with this

```
    page vars      ;vars on imp vars page

    locdef d2f1
    locdef f2d1
    locdef tty1ok
    locdef ddt1ok

6266    poltim: .blkw 1      ;time at last poll from operational pgm
6268    dxtflag: .blkw 1      ;flag: ddt should be polled (dxt ran)

626A    dspflag: .blkw 1      ;if zero, don't display traps

626C    d2fb: .blkw 1      ;ddt to fake buffer
626E    d2fpok: .blkw 1

    ;display, ddt, tty poke flags
6270    ttpoke: .blkw 1
6272    ddpoke: .blkw 1
6274    dsppok: .blkw 1

6276    f2db: .blkw 1      ;fake to ddt buffer
6278    f2dpok: .blkw 1

627A    f2tpok: .blkw 1
627C    t2fpok: .blkw 1

    ;define format of ring buffer and pointers
    0014    bufsiz= D20          ;how many bytes in buffer
            page dummy
            .=0
0000    rbuf: .blkb bufsiz     ;buffer
0014    rbufs: .blkb 1         ;pointer to start of buffer
0015    rbufe: .blkb 1         ;pointer to end of buffer
    4016    rbflok= .+foo
0016    .blkw 1               ;locks the block
0018    rbuflen: .blkw 0       ;length of buffer and pointers

    page vars
627E    ttyibf: .blkb rbuflen ;tty puts stuff here
6296    ttyobf: .blkb rbuflen ;tty writes from here

62AE    anom: .blkw 1 ;anomalies word
1000    hdcbit= H1000 ;host data checksum is bad
0400    msgbit= H400  ;message generator bit
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 79
DDT.PLR;1 PAGE 4.1 DDT and TTY code

```
0200      cumbit= H200 :cumulative statistics bit
0020      HTSBIT= H20 :HEAT SENSOR SWITCH USE TO BE SAT
0010      ovrbit= H10 :override switch bit in anom
0004      SS2BIT=4   :SENSE SWITCH 2 BIT
0001      ss4bit=1  :sense switch 4 bit in anom
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 5 DDT and TTY code

PAGE 80

239

```
;DDT variables
page Vars

62B0      ddvarst:     .blkw 0          ;start of area to clear on init
           ;argument flags
62B0      pfx:        .blkw 1          ;non-zero means two args typed
62B2      something:   .blkw 1          ;non-zero means something typed

           ;argument variables
62B4      ddaccum:    .blkw 1          ;most recently typed arg
62B6      sum:        .blkw 1          ;for addition
62B8      decnum:    .blkw 1          ;argument in hex
62BA      prefix:    .blkw 1          ;first arg of two
62BC      ddivarend:  .blkw 0          ;end of input variables

           ;deposit/examine variables
62BC      locopen:    .blkw 1          ;non-zero means open
62BE      curadd:    .blkw 1          ;alternative to band-aid
62CO      mapbits:   .blkw 1          ;mapbits of curadd
62C2      dmap:       .blkw 4          ;map values for four current maps
62CA      ddtproc:   .blkw 1          ;processor mask for this access
62CC      lastnum:   .blkw 1          ;last number typed out
62CE      ddvarend:  .blkw 0          ;end of init vars

           Page DDTVars
;blk interface buffers on DDT page
ddtbf2: .blkw 1          ;buffer for two word store
ddtbuf: .blkw 1          ;buffer for examines. one word stores
```

```
0000          .parity 0           ;7 bit ascii
              Page LCode
              ;POLLER- Stage's connection to ddt/tty
              ;  called with s1stack out of wsleep so as to not interfere
              ;  with saved system stack if application not running
OF96 1076    routine poller
OF98 4078 OFBC  call finddt      ;set up maps
OF9C 7078 40B4  if intime       ;can run this stuff
OFA0 9A0D
OFA2 49F1    sub r7,#1 ;intime   ;count down timer
OFA4 3078 40B4  sta r7,intime
OFA8 4078 42B8  call ddtpol    ;check for ddt stuff
OFACT 4078 40C0  call dxtchk    ;check crosspatch
OFBO 4078 4134  call tttypol   ;do tty
OFB4 48F3    set intime = #3   ;hold off timer
OFB6 3078 40B4
              endif
OFBA 6006    endroutine poller

routine finddt, nosave      ;don't need much so save an instr
OFBC 7018 0000  lda r1,mapvar
OFC0 3018 FC02  sta r1,%map1
OFC4 3018 FC06  sta r1,%map3
OFC8 7018 00B2  lda r1,mapddt
OFC0 3018 FC00  sta r1,%map0
OFDO 4807    endroutine finddt

$dopatch jpoll, LCode      ;connet poller with system
0310 OF96    poller
```

FC00 0200 Page DDTCode
40B8 7B00
40BA 4236
40BC 0000
40BE 0002

;DXTCHK- checks and does the crosspatching of tty and ddt
; if a) tty hasn't been polled for a long time; or
; b) the sign bit is set in DEBUGM.

40C0 1076 routine dxtchk, uses r1/r2
40C2 7078 6266 lda r7,poltime ;last time polled
40C6 7178 00AO sub r7,stime ;time now
40CA 7478 030C tor r7,debugm
40CE 4F78 FF00 if r7 .bit. #OFF00 ;too long or always crosspatch
40D2 9A21
40D4 7078 A260 lock f2d1 ;give stuff to ddt?
40D8 9AFE
40DA 7878 6276 ifnot byte f2db ;ddt took last, so get more
40DE 8A09
40E0 4078 1008 call rbfget, ttyibf ;get stuff from tty
40E4 627E
40E6 9105 ifnot equal ;something really happened
40E8 3818 6276 stab r1,f2db ;put result away
40EC 3008 6268 set dxtflag ; so make sure ddt goes
 endif
 endif
40FO 3008 A260 unlock f2d1
40F4 7078 A25E lock d2f1
40F8 9AFE
40FA 7818 626C ldab r1,d2fb ;stuff from ddt?
40FE 9A09 ifnot zero ;yup, give to tty
4100 4078 OFD2 call rbfput, ttyobf ;put on tty's output buffer

```
4104 6296
4106 9105           ifnot equal      ;buffer ok
4108 3918 626C       subbm r1,d2fb   ;clear ddt's buffer
410C 3008 6268       set dxtflag    ; and make sure ddt runs
                      endif
                      endif
4110 3008 A25E       unlock d2f1
                      endif
4114 6006           endroutine dxtchk
```

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 8 TTY handler

PAGE 84

```
.stitle TTY handler

:tty poller connections

;TTSLEEP to put tty handler to sleep
; can accept any map settings in m2,m3
; returns standard m0-m3
4116 1076 routine ttsleep, local r1-r5, arg m1, result m0-m3
4118 1016
411A 1026
411C 1036
411E 1046
4120 1056
4122 4078 40CO    call dxtchk          ;check crosspatching
4126 3068 5B36    sta sp,ttysp        ;save our stack
412A 7068 5B38    lda sp,ttsyssp      ; and get system's stack
412E 3008 6262    unlock m1#ttylok     ;release both stacks
4132 6006          pop r0             ;return to system
4134 1076          ttypol:           ;entry from system
4136 7078 A262    save r7            ;save return
413A 9AFE          lock r7,ttyleok   ;get control of stacks
413C 3068 5B38    sta sp,ttsyssp      ;save system stack
4140 4877          lda r7,r7       ;:ttyleok
4142 9909          bo ttyini        ;bad lock- reset
4144 7068 5B36    lda sp,ttysp        ;get tty's stack
4148 6056          endroutine ttsleep
414A 6046
414C 6036
414E 6026
4150 6016
4152 6006
```

```
:TTYINI to start tty and reset tty on quits or resets
: entered at first run of tty when ttylck odd or when
: reference to psb gives quit
ttyini:
4154 4878 0006    lda r7,#1psbtab      :how many to check
                  repeat      :find good psb
4158 505F 4190    lda r5,psbtab(-r7)   :try another
415C 9805          until loop      :no good psb's
415E 7015          lda r1,(r5)      :see if there
4160 80FC          next if quit    :not this one
4162 9002          break
4164 90FA          endrepeat
4166 4E54          if r5 <> r4      :new psb}
4168 910E

Trap 300,<;DDT changed psbs (VISTAR using new BUS) - page 85>

416A E000
5
416C 4845    lda r4,r5      :use new one
416E 8AOB    if zero       :got none}
4170 7078 A2AC    lock ttyobf+rbf1ok  ;reset some stuff
4174 9AFE
4176 7078 A2AA    lda r7,ttyobf+rbufs+foo  ;clear out buffer
417A 3008 A2AC    unlock ttyobf+rbf1ok
417E 4078 4116    call ttsleep
4182 90E9    br ttyini      :try again
                  endif
                  endif
4184 3004    set (r4)      :reset interface
4186 80E7    qutpat ttyini
4188 4868 5B36    lda sp,#ttstack  ;correct stack pointer
418C 900B    BR FVSTAR
418E 8000    NOP

;table of psbs we may use
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 9.1 TTY handler

PAGE 86

```
4190      table psbtab
4190 0000      0          ;zero flag to mark end
4192 EA00      psbtwo
4194 FA00      psbone
endtable psbtab
```

```
;TTY- main loop to check for output and input to do.  
; Wait for input unless output to do  
tty:                                :initial entry from TTYINI  
repeat  
4196 4078 4116      call ttsleep  
419A 4078 1008      call rbfget. ttyobf      ;try to get output to do  
419E 6296  
41AO 9105      ifnot equal      :got some to do .  
FVSTAR:  
41A2 4078 41FA      call ttyout      :send output  
41A6 90F8          next      :check for more output  
41A8 900C          else      :no output to do .  
41AA 7074          lda r7,(r4)  ;:psbstat      ;check status  
41AC 890A          if odd  ;:psbfree      ;not busy so avail  
41AE 707C 0006      if psbctl(r4) .bit. #psbout      :was output.  
41B2 4FF2  
41B4 9A04  
41B6 48F1          set psbctl(r4) = #psbact      ; can change .  
41B8 307C 0006  
41BC 4078 41C2      endif  
                     call ttystuff      :get input  
                     endif  
                     endif  
41CO 90EB          endrepeat  
  
;handle inputs  
41C2 1076          routine ttystuff, arg r4, uses r1  
41C4 7074          lda r7,(r4)  ;:psbstat      ;check status  
41C6 4FF8          if r7 .bit. #psbbrk      :got break  
41C8 9A03
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 88
DDT.PLR;1 PAGE 10.1 TTY handler

```
41CA 4078 4154    call ttyini          ; so reset
                   endif
41CE 4FF1          if r7 .bit. #psbfree   ;not busy
41D0 9A14
41D2 701C 0008    lda r1,psbdat(r4)  ;get char
41D6 4B18 007F    and r1.#7F        ;just these bits
41DA 4E18 0060    if r1 < # 0140      ;upper limit of printables
41DE 8206
41EO 4E9A          if r1 >= #asciLF
41E2 9204
41E4 4078 OFD2    call rbfput, ttyobf ;echo printing chars
41E8 6296          endif
                   endif
41EA 4078 OFD2    call rbfput, ttyibf   ;stuff char into buffer
41EE 627E          set @pid = t2fpok   ;poke fake to take char
41F0 7078 627C
41F4 30F8 00AC    endif
41F8 6006          endroutine ttystuff
```

```
        ;output handler
41FA 1076    routine ttyout, arg r1/r4
41FC 7078 A274    lda r7,dspok+foo      ;poke whoever poked us
4200 8A03    if zero
4202 7078 A27A    lda r7,f2tpok+foo
4206 30F8 00AC    endif
420E 707C 0006    sta r7,@pid
4210 8A03    repeat
4212 8A08    lda r7,psbct1(r4)      ;now writ char
4214 1016    lda r7,psbact          ;get cntl reg
4216 4078 41C2    qutpat ttyini      ;lost psb- reset
421A 6016    if r7 .nbit. #psbout   ;input mode
421C 48F3    save r1              ;save what to type
421D 4078 41C2    call ttystuff      ;use input handler
421E 6006    pop r1
4220 48F3    set psbct1(r4) = //psbact+psbout   ;set output mode
4222 7074    endif
4224 9906    lda r7,(r4)          ;:psbstat
4226 3008 6270    until odd      ;:psbfree   ;until psb not busy
422A 4078 4116    set ttpoke       ;get poked later
422E 90EE    call ttsleep
4230 301C 0008    endrepeat
4234 6006    sta r1,psbdat(r4)      ;put data into psb
4234 6006    endroutine ttyout
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 12 TTY handler

PAGE 90

249

;ring handling routines

Page LCode ;this code used by fakes also
;RBFPUT-put r1 onto ring
; inline pointer to buffer
; returns EQUAL if FAIL ({}{})
routine rbfput, arg r1, local r2, inline r2

OFD2 1076
OFD4 1026
OFD6 6027
OFD8 307E 0002
OFDC 707A 4016 lock rbflok(r2) ;lock buffer
OFE0 9AFE
OFE2 787A 0015 ldat r7,rbufe(r2) ;get end pointer
OFE6 4A72 add r7,r2 ;get into block
OFE8 2817 stab r1,(r7)+ ;insert into buffer
OFEA 4972 sub r7,r2 ;fix before saving
O FEC 7E7A 0014 if byte r7 <> rbufs(r2) ;still room left
OFF0 9108
OFF2 4E78 0014 if r7 >= #bufsiz ;wrapped
OFF6 9203
OFF8 48F0 lda r7,#0 ;reset pointer
OFFA 0201 rst %e ;fix cc for return
endif
OFFC 387A 0015 stab r7,rbufe(r2) ;save pointer
endif
1000 300A 4016 unlock rbflok(r2)
1004 6026 endroutine rbfput
1006 6006

;RBFGET-get r1 from ring
; inline pointer to buffer
; returns EQUAL if FAIL ({}{})
routine rbfget, local r2, inline r2, result r1

1008 1076
100A 1026
100C 6027
100E 307E 0002

```
1012 707A 4016    lock rbflok(r2)          ;control block
1016 9AFE
1018 787A 0014    l1dab r7,rbufs(r2)      ;get start pointer
101C 4E78 0014    if r7 >= #bufsiz        ;wrapped
1020 9202
1022 48FO          lda r7,#0              ;reset pointer
1024 7E7A 0015    endif
1028 9106
102A 4A72          add r7,r2              ;get into buffer
102C 6817          l1dab r1,(r7)+       ;remove from buffer
102E 4972          sub r7,r2              ;restore just offset
1030 387A 0014    stab r7,rbufs(r2)      ;save pointer
1034 300A 4016    endif
1038 6026          unlock rbflok(r2)
103A 6006          endroutine rbfget
```

FC00 0200

Page DDTCode

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 13 DDT Code

PAGE 92

```
.stitle DDT Code

:DDTINI- Stage AR's init of DDT code
4236 1076 routine ddtini
4238 7078 40B4      ifnot intime
423C 8A08
423E 6076      fail coreturn      :get consensus
4240 4FF0
4242 4077
4244 1076
4246 48F3      set intime = #3      ;do just once
4248 3078 40B4
424C 6076      endif
424E 0281      return equal
4250 4007
endroutine ddtini
```

```
;DDRESET- clears buffers, sets init ddt maps, clears args
4252 1076          routine ddreset
4254 7078 A294      lock ttyibf+rbfl0k           ;lock and then clear this ring
4258 9AFE
425A 7078 A292      lda r7,ttyibf+rbufs+foo ;;rbufe      ;clear both start and end pntrs
425E 3008 A294      unlock ttyibf+rbfl0k
4262 7078 A2AC      lock ttyobf+rbfl0k
4266 9AFE
4268 7078 A2AA      lda r7,ttyobf+rbufs+foo ;;rbufe
426C 3008 A2AC      unlock ttyobf+rbfl0k
4270 4890          lda r1,#0                  ;clear some vars
4272 4828 62B0      lda r2,#ddvarst          ;where to start clearing from
repeat
        sta r1,(r2)+       until r2 = #ddvarend
427C 81FD
endrepeat
427E 4878 8000      set ddtproc = #sign        ;default is all procs
4282 3078 62CA
4286 4878 62C2      lda r7,#dmap            ;set up init ddt maps
428A 4894          lda r1,#4                ;mapcod
428C 2017          sta r1,(r7)+           ;map0
428E 4828 0020      lda r2,#20              ;mapvar
4292 2027          sta r2,(r7)+           ;map1
4294 4818 0022      lda r1,#22              ;map2
4298 2017          sta r1,(r7)+           ;mapvar
429A 2027          sta r2,(r7)+           ;mapvar, map3
429C 6006          endroutine ddreset
```

```
;ddt's poller stuff here
429E 1076    routine dsleep, local r1-r5, arg m1, result m0-m3
42A0 1016
42A2 1026
42A4 1036
42A6 1046
42A8 1056
42AA 3068 5B1E    sta sp,ddtsp          ;save DDT stack pointer
42AE 7068 5B20    lda sp,dsyssp        ;restore system stack pointer
42B2 3008 6264    unlock m1#ddtlok      ;release
42B6 6006    pop pc                  ;return to system
42B8 1076    ddtpol:                ;system entry
42BA 7078 A264    push r7                ;save caller
42BE 9AFE
42C0 3068 5B20    lda r7,r7            ;lock died}- reset ddt
42C4 4877
42C6 9909    bo ddienter           ;get our stack
42C8 7068 5B1E    lda sp,ddtsp          ;endroutine dsleep
42CC 6056
42CE 6046
42DO 6036
42D2 6026
42D4 6016
42D6 6006

;DDIENTER- initial entry to ddt to get stack and
; initial content right
ddienter:
42D8 4868 5B1E    lda sp,#ddstack       ;start at beginning of stack
42DC 4078 4252    call ddreset          ;fix up variables
42E0 4078 4820    call itextout, herald   ;type herald
42E4 42EC
42E6 4078 44D4    call argreset         ;do some more reset and type tab
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 95
DDT.PLR;1 PAGE 15.1 DDT Code

42EA 9031 br disploop ;and start ddt here
42EC 44 herald: .asciz /DDT} /
42ED 44
42EE 54
42EF 21
42FO 20
42F1 00
.even

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 16 DDT Code

PAGE 96

```
;stage error decoder
42F2 1076    routine ddtste, arg r1, local r2
42F4 1026
42F6 7078 0154    lda r7,bltmyr< D9*words>    ;Location of error
42FA 1076    push r7    ;save it to print later
42FC 3D18 0160    eorbm r1,bltmyc    ;clear error flag
4300 1016    push r1    ;save for later
4302 7018 00A8    lda r1,procbt    ;processor that got it
4306 1016    push r1    ;save over multi-proc calls
4308 4078 479C    call iddtout, ascispace
430C 0020
430E 6016    pop r1    ;get back proc bit
4310 4078 4830    call numout
4314 6016    pop r1    ;bltmyc    ;error that happened
4316 4B97    and r1,#locqut}locilo}locfad    ;possible errors
4318 A291    sll r1,1    ;offset to table
431A 7029 4338    lda r2,stetab(r1)    ;get pointer to text
431E 4078 481C    call textout
4322 4078 479C    call iddtout, '@'
4326 0040
4328 6016    pop r1    ;bltmyr    ;where it happened
432A 4078 4830    call numout
432E 4078 4820    call itextout, %3space
4332 465E
4334 6026    endroutine ddtste
4336 6006

4338    table stetab
4338    notext
```

```
433A 4698      quitxt    ::locqut*2
433C 4342      iloptxt   ::locilo*2
433E 46AA      notext
4340 4346      fadtext   ::locfad*2
                           endtable stetab

4342 20        iloptxt: .asciz / IL/
4343 49
4344 4C
4345 00
4346 20        fadtext:  .asciz / FADE/
4347 46
4348 41
4349 44
434A 45
434B 00
                           .even
```

```
;command dispatch loop

disploop:
repeat
434C 4078 47D0    call ddtget ;get char from wherever
4350 4E18 0060    if r1 >= # 0140
4354 9209
4356 4E18 007D    if r1 >= = 0175
435A 9204
435C 4918 001D    sub r1,# 0175- 0140
4360 9003    else
4362 4918 0020    sub r1,#'a-'A
                    endif
endif
4366 4871    lda r7,r1
4368 A2F1    s11 r7,1           ;dispatch table offset
436A 707F 43AO    lda r7,ddtdispatch(r7)
436E 9A16    ifnot zero        ;there is dispatch
4370 890A    if odd   ;;ovrrid  ;needs override
4372 49F1    sub r7,#1         ;a cheap AND
4374 7028 62AE    lda r2,anom
4378 4F28 0010    if r2 .nbit. #ovrbit ;isn't on,
437C 8A04
437E 4078 44CA    call rubout      ;so can't do
4382 90E5    next
                    endif
endif
4384 7028 62B4    lda r2,ddaccum ;get current args for routines
4388 7038 62B6    lda r3,sum
438C 4A32    add r3,r2
438E 7048 62BA    lda r4,prefix
4392 7058 62B0    lda r5,pfx
4396 4077    call (r7)          ;go to dispatch
next
```

4398 90DA
endif
439A 4078 44CA call rubout
439E 90D7 endrepeat

```
.comnt |
dispatch table
 0 - illegal char
|
 0001      ovrid=1           ;if odd, need override
43AO          table ddtdispatch
43AO 0000    .word 0        : @    000    00
43A2 0000    .word 0        : A    001    01
43A4 0000    .word 0        : B    002    C2
43A6 0000    .word 0        : C    003    03
43A8 0000    .word 0        : D    004    04
43AA 0000    .word 0        : E    005    05
43AC 0000    .word 0        : f    006    06
43AE 0000    .word 0        : G    007    07
43BO 0000    .word 0        : H    010    08
43B2 0000    .word 0        : I    011    09
43B4 44F8    lf              : J    012    0A (line feed)
43B6 0000    .word 0        : K    013    0B
43B8 0000    .word 0        : L    014    0C
43BA 4522    cr              : M    015    0D (car ret)
43BC 0000    .word 0        : N    016    0E
43BE 0000    .word 0        : O    017    0F
43CO 0000    .word 0        : P    020    10
43C2 0000    .word 0        : Q    021    11
43C4 0000    .word 0        : R    022    12
43C6 0000    .word 0        : S    023    13
43C8 0000    .word 0        : T    024    14
43CA 0000    .word 0        : U    025    15
43CC 0000    .word 0        : V    026    16
43CE 0000    .word 0        : W    027    17
43DO 0000    .word 0        : X    030    18
43D2 0000    .word 0        : Y    031    19
43D4 0000    .word 0        : Z    032    1A
43D6 0000    .word 0        : [    033    1B
43D8 0000    .word 0        : \    034    1C
43DA 0000    .word 0        : ]    035    1D
43DC 0000    .word 0        : _    036    1E
43DE 0000    .word 0        : "    037    1F
43EO 448E    space          : space 040    20
43E2 0000    .word 0        : }    041    21
43E4 0000    .word 0        : "    042    22
```

| | | | | | |
|-----------|---------|---|----|-----|----|
| 43E6 0000 | .word 0 | : | # | 043 | 23 |
| 43E8 4472 | decimal | : | \$ | 044 | 24 |
| 43EA 0000 | .word 0 | : | % | 045 | 25 |
| 43EC 0000 | .word 0 | : | & | 046 | 26 |
| 43EE 0000 | .word 0 | : | , | 047 | 27 |
| 43FO 0000 | .word 0 | : | (| 050 | 28 |
| 43F2 0000 | .word 0 | : |) | 051 | 29 |
| 43F4 0000 | .word 0 | : | * | 052 | 2A |
| 43F6 0000 | .word 0 | : | + | 053 | 2B |
| 43F8 447E | comma | : | . | 054 | 2C |
| 43FA 0000 | .word 0 | : | - | 055 | 2D |
| 43FC 4466 | dot | : | . | 056 | 2E |
| 43FE 45CA | slash | : | / | 057 | 2F |

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 19 DDT Code

PAGE 102

| | | | | | |
|-----------|---------|---|---|-----|----------------------|
| 4400 44A4 | digit | : | 0 | 060 | 30 |
| 4402 44A4 | digit | : | 1 | 061 | 31 |
| 4404 44A4 | digit | : | 2 | 062 | 32 |
| 4406 44A4 | digit | : | 3 | 063 | 33 |
| 4408 44A4 | digit | : | 4 | 064 | 34 |
| 440A 44A4 | digit | : | 5 | 065 | 35 |
| 440C 44A4 | digit | : | 6 | 066 | 36 |
| 440E 44A4 | digit | : | 7 | 067 | 37 |
| 4410 44A4 | digit | : | 8 | 070 | 38 |
| 4412 44A4 | digit | : | 9 | 071 | 39 |
| 4414 0000 | .word 0 | : | : | 072 | 3A |
| 4416 4798 | ddtout | : | : | 073 | 3B (just echo semis) |
| 4418 0000 | .word 0 | : | < | 074 | 3C |
| 441A 0000 | .word 0 | : | = | 075 | 3D |
| 441C 0000 | .word 0 | : | > | 076 | 3E |
| 441E 0000 | .word 0 | : | : | 077 | 3F |
| 4420 0000 | .word 0 | : | @ | 100 | 40 |
| 4422 449E | hexletr | : | A | 101 | 41 |
| 4424 449E | hexletr | : | B | 102 | 42 |
| 4426 449E | hexletr | : | C | 103 | 43 |
| 4428 449E | hexletr | : | D | 104 | 44 |
| 442A 449E | hexletr | : | E | 105 | 45 |
| 442C 449E | hexletr | : | F | 106 | 46 |
| 442E 0000 | .word 0 | : | G | 107 | 47 |
| 4430 0000 | .word 0 | : | H | 110 | 48 |
| 4432 0000 | .word 0 | : | I | 111 | 49 |
| 4434 0000 | .word 0 | : | J | 112 | 4A |
| 4436 0000 | .word 0 | : | K | 113 | 4B |
| 4438 0000 | .word 0 | : | L | 114 | 4C |
| 443A 0000 | .word 0 | : | M | 115 | 4D |
| 443C 0000 | .word 0 | : | N | 116 | 4E |
| 443E 0000 | .word 0 | : | O | 117 | 4F |
| 4440 0000 | .word 0 | : | P | 120 | 50 |
| 4442 0000 | .word 0 | : | Q | 121 | 51 |
| 4444 0000 | .word 0 | : | R | 122 | 52 |
| 4446 0000 | .word 0 | : | S | 123 | 53 |
| 4448 0000 | .word 0 | : | T | 124 | 54 |
| 444A 0000 | .word 0 | : | U | 125 | 55 |
| 444C 0000 | .word 0 | : | V | 126 | 56 |
| 444E 0000 | .word 0 | : | W | 127 | 57 |
| 4450 0000 | .word 0 | : | X | 130 | 58 |
| 4452 0000 | .word 0 | : | Y | 131 | 59 |

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
DDT.PLR;1 PAGE 19.1 DDT Code

```
4454 0000 .word 0 : Z 132 5A
4456 0000 .word 0 : [ 133 5B
4458 0000 .word 0 : \ 134 5C
445A 0000 .word 0 : ] 135 5D
445C 0000 .word 0 : _ 136 5E
445E 0000 .word 0 : - 137 5F

4460 0000 .word 0 : code 175 folded to 140
4462 0000 .word 0 : code 176 folded to 141
4464 44CA rubout : rubout, code 177 folded to 142

endtable ddtdispatch
```

```
.comnt |
character dispatches
  character in r1 for all dispatches
|  
  
4466 1076      routine dot
4468 7078 62BE    set ddaccum = curadd
446C 3078 62B4
4470 6006      endroutine  
  
.comnt | Decimal to convert the current argument into decimal |
4472 1076      routine decimal
4474 7078 62B8    set ddaccum = decnum
4478 3078 62B4
447C 6006      endroutine decimal
@  
  ;handle argument seperator
447E 1076      routine comma, arg r3
4480 3038 62BA    sta r3,prefix          ;put into prefix
4484 3008 62B0    set pfx                ;say we have one
4488 4078 44E0    call accreset         ;reset ddaccum etc
448C 6006      endroutine comma  
  
        ;add two args
448E 1076      routine space, arg r2-r3
4490 3128 62B4    subm r2,ddaccum       ;clear accumulator
4494 7078 A2B8    lda r7,decnum+foo   ;clear decimal number
4498 3038 62B6    sta r3,sum           ; and save sum
449C 6006      endroutine space
```

```
;handle alphabetic hex digits
449E 1076    routine hexletr, arg r1-r2
44AO 4997    sub r1,#<'A-0A>-'0          ;convert to something useful
44A2 9002    entry digit, arg r1-r2      ;for numerals
44A4 1076
44A6 4918 0030   sub r1,#'0              ;convert to number
44AA A2A4    s11 r2,4                ;shift accumulated argument
44AC 4A21    add r2,r1               ;include new digit
44AE 3028 62B4   sta r2,ddaccum
44B2 3008 62B2   set something
44BG 7028 62B8   lda r2,decnr        ;remember something typed
44BA 4872    lda r7,r2               ;accumulate decimal number
44BC A2A2    add r2,r1               ;first multiply old by 10.
44BE 4A27    s11 r2,2                ;(4x+x)+2=10x
44CO A2A1    s11 r2,1
44C2 4A21    add r2,r1               ;add new decimal digit
44C4 3028 62B8   sta r2,decnr
44C8 6006    endroutine hexletr       ;;digit

;rubout to reset input
44CA 1076    routine rubout
44CC 4078 4820   call itextout, rubtxt
44DO 44F4
44D2 9002    entry argreset         ;reset args and flags
44D4 1076
44D6 7078 A2B0   lda r7,pfx+foo
44DA 7078 A2BA   lda r7,prefix+foo
44DE 9002    entry accreset         ;reset current input
44EO 1076
44E2 7078 A2B4   lda r7,ddaccum+foo
44EG 7078 A2B6   lda r7,sum+foo
44EA 7078 A2B2   lda r7,something+foo
44EE 7078 A2B8   lda r7,decnr+foo
44F2 6006    endroutine rubout
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 106
DDT.PLR;1 PAGE 21.1 DDT Code

```
44F4 23      rubtxt: .asciz /* / ;# and two spaces
44F5 20
44F6 20
44F7 00
        .even

44F8 1076    routine lf, uses r1
44FA 7078 62BE lda r7,curadd           ;get curr address :
44FE 4AF2     add r7,#words          ;next one to look at
4500 4855     lda r5,r5             ;:pfx
4502 9A02     ifnot zero            ;doing two word store
4504 4AF2     add r7,#words          ;so skip two words :
4506 1076     endif
4508 4078 4522 save r7              ;save over call
450C 6016     pop r1               ;do deposit, close loc as req'd
450E 4831     lda r3,r1             ;recover new address
4510 4078 4830 call numout          ;argument for slash
4514 4078 479C call iddtout, '/'   ;type new location
4518 002F     call slash             ;and slash
451A 48D0     clear r5              ;no prefix
451C 4078 45CA call slash             ;use slash to do the work
4520 6006     endroutine lf
```

```
;handle carriage returns
4522 1076 routine cr, arg r3-r5
4524 4078 4820 call itextout, crlftxt
4528 458E
452A 7078 62BC If locopen :location still open
452E 9A2D
4530 7078 62B2 if something :something typed
4534 9A27
4536 7078 00A8 if procbt .bit. debugm } anom .bit. #ovrbit
453A 7778 030C
453E 8AO6
4540 7078 62AE
4544 4F78 0010
4548 9A1B
454A 3038 5B3C sta r3,ddtbuf ;ddaccum sum :what to deposit
454E 3048 5B3A sta r4,ddtbf2 ;prefix :use second buffer
begin deposit
    call bltget :get control of params
    call setdep
    lda r1,#bltact}bltcck}bltfu1 :to turn on blt
    455C 4078 4720 call bitdo :and do it
    4560 4078 4592 call setdep :do spare now
    4564 4878 080B set m3//bltst = //bltact}bltfu1}bltcck}bltspa
    4568 3078 BDBC
    456C 3008 6272 set ddpoke :have poller poke BLT
    4570 4078 45AC call derrprint :errors for main
    4574 4078 475E call bltget :check spare store
    4578 4078 45AC call derrprint :spare errors
end deposit
else
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 108
DDT.PLR;1 PAGE 22.1 DDT Code

457C 9003
457E 4078 44CA call rubout ;can't do this
endif
endif
4582 48F0 clear locopen
4584 3078 62BC
endif
4588 4078 44D4 call argreset
458C 6006 endroutine cr

458E 0D crlftxt: .byte ascicr,ascilf,0,0
458F 0A
4590 00
4591 00

```
:set up parameters for deposit and differentiate
; between one/two -word stores. r5 = pfx
4592 1076 routine setdep, arg r5
4594 4078 46AC call setblk          ;get blk and set up parameters
4598 4875 if r5      ;:pfx      ;two words
459A 9A08
459C 4878 9B3A set m3#blkbf = #m2#ddtbf2      ;use second buffer
45A0 3078 BDC8
45A4 48F4 set m3#bltsiz = #2^words ;how many bytes
45A6 3078 BDC2
        endif
45AA 6006 endroutine setdep

:type errors from deposit
routine derrprint, arg r1, uses r5
45AC 1076    lda r5,m3#blkbf      ;in case error
45AE 7058 BDCC unlock blklok
45B2 3008 BDBA setmap m3, mapvar      ;for ttylek etc
45B6 7078 0ODO
45BA 3078 FC06
45BE 4F18 7074 if r1 .bit. #blkters   ;errors
45C2 9A03
45C4 4078 4662 call errprint
        endif
45C8 6006 endroutine derrprint
```

```
;open a location
45CA 1076 routine slash, arg r3-r5, uses r1-r5
45CC 3038 62BE sta r3,curaddr ;save for this ref and future
45D0 4B38 E000 and r3,#OE000 ;just map bits
45D4 A3B4 r11 r3,4 ;map#+4
45D6 49B4 sub r3,#4 ;map#
45D8 3038 62C0 sta r3,mapbits ;save it for others
45DC 4875 if r5 :: pfx ;prefix typed
45DE 9AOA
45E0 4833 lda r3,r3 ::mapbits ;what to do with it
45E2 9B03 if minus } r3 > #6 ;local address
45E4 4EB6
45E6 8C04
45E8 3048 62CA sta r4,ddtproc ;prefix ; so save processor mask
45EC 9003 else ;common reference
45EE 304B 62C2 sta r4,dmap(r3) ;prefix ; save map in DDT's reg
endif
endif
begin examine
call bltget ;control params
call setbit
set r1 = #bltact+bltsdf
call bltdo
if r1 .nbit. #bltsdf ;bltst ;spare differs
4606 8A0D
4608 4078 475E call setbit ;fix parameters again
460C 3008 6272 set ddpoke ;get poller to poke
4610 4878 0801 lda r7,#bltact+bltspa ;read spare
4614 3078 BDBC sta r7,m3#bltst ;activate
4618 4078 462E call answer ;release BLT & type answer
461C 4078 475E call bltget ;get control when blt done
```

```
4620 4078 462E      endif
                     call answer
                     end examine
4624 3008 62BC      set locopen
                     ;typ spare if dif, else typs ans
4628 4078 44D4      call argreset
                     ;allow stores
462C 6006      endroutine slash
                     ;prepare for next
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 112
DDT.PLR;1 PAGE 25 DDT Code

```
:interpret error bits from blt examine and
: print text as req'd
462E 1076 routine answer, arg r1/r4/r5, uses r1
4630 7058 BDCC    lda r5,m3#/bltbmk      ;mask of procs failed
4634 7048 5B3C    lda r4,ddtbuf        ;answer is here
4638 3008 BDBA    unlock bltlok       ;release blt
463C 7078 0ODO    setmap m3,mapvar   ;for ttyleok, ddtlok
4640 3078 FC06
4644 4F18 7074    if r1 .bit. #blters     ;blt got errors
4648 9A04
464A 4078 4662    call errprint        ;r3-err bits, r5-procs
464E 9004    else
4650 4814    lda r1,r4
4652 4078 4830    call numout         ;print answer out
endif
4656 4078 4820    call itextout, %3space   ;type spaces
465A 465E
465C 6006    endroutine answer

465E 20    %3space:      .asciz /  /
465F 20
4660 20
4661 00    .even

;print errors
4662 1076 routine errprint, arg r1/r5, uses r1/r2
4664 4878 0074    lda r7,#OFF&blters   ;low byte mask of errors
4668 4B71    and r7,r1      ;:bltst      ;low byte errors
466A 4828 46A7    lda r2,#totxt     ;assume timeout error
466E 4FF4    if r7 .nbit. #blteto   ;blt didn't time out
4670 8A06
4672 A698    srl r1,8
```

```
4674 4C17      ior r1,r7          :get errors together
4676 A693      sr1 r1,3          ;error table offset-LESS THAN 8}}}
4678 7029 468E  lda r2,etab(r1)   ;get text pointer
467C 4078 481C  endif
4680 4815      call textout      ;type text
4682 4078 4830  lda r1,r5      ;b1tmk
4686 4078 4820  call numout      ;procs failing this way
468A 465E      call itextout, %3space
468C 6006      endroutine errprint

468E           table etab
468E 46AA      notext    ;:0      ;no error
4690 4698      quitxt   ;:b1tesq b1tedq ;source/dest quit
4692 469E      nxistxt  ;:b1tesk b1tdsk ;non-existent reference
4694 46AA      notext    ;:6      ;not error
4696 46A1      formerr  ;:b1tesf b1tedf ;format error
469E           endtable etab      ;room for just 0-8}}}
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 114
DDT.PLR;1 PAGE 26 DDT Code

```
;error text:  
4698 20     quitxt: .asciz / QUIT/  
4699 51  
469A 55  
469B 49  
469C 54  
469D 00  
469E 4E     nxistxt: .asciz /NX/  
469F 58  
46A0 00  
46A1 46     formerr: .asciz /FRMT /  
46A2 52  
46A3 4D  
46A4 54  
46A5 20  
46A6 00  
46A7 54     totxt: .asciz \TO \  
46A8 4F  
46A9 20  
46AA 00  
46AA    notext=-1 ;bum to print no text  
        .even
```

```
.comnt |
SETBLT to set up blt parameters from ddt arguments
|
46AC 1076      routine setblt, uses r2/r3
46AE 7078 62BE  lda r7.curadd          ;get address to look at
46B2 7038 62CO  lda r3.mapbits        ; and map reg to use
46B6 9B11       ifnot minus } r3 > #6   ;restrict to map0 - map3
46B8 4EB6
46BA 9C0F
46BC 4B78 1FFE  and r7,#packm         ;just page offset for common
46C0 4CF1       ior r7,#bltnlc        ;mark as common ref
46C2 702B 62C2  lda r2,dmap(r3)      ;get map we're supposed to use
46C6 9904       if odd } r2 >= #200    ;physical map
46C8 4E28 0200
46CC 9203
46CE A6A9       srl r2,9            ;leave just map setting
46D0 9003       else               ;logical page
46D2 4C78 8000  ior r7,#bltlog        ;tell BLT
endif
else
46D6 900D       lda r2,ddtproc      ;local examine
46D8 7028 62CA  if minus           ;set processor mask words
46DC 8B03       lda r2,m2#procex    ;if "all procs" flag is set
46DE 7028 9D64  endif              ;use all existing processors
46E2 3028 BDD8  sta r2,m3#bltspm    ;source proc mask
46E6 3028 BDDC  sta r2,m3#bltdpm    ;dest proc mask
46EA 3028 BDDE  sta r2,m3#bltdpi    ;initial dest proc mask
46EE 48AF       lda r2,#OF          ;mark as using mask
endif
46FO 3028 BDD6  sta r2,m3#bltsty    ;set proc mask or map
46F4 3028 BDDA  sta r2,m3#bltdty    ;address to look at
46F8 3078 BDC0  sta r7,m3#bltadd    ;set up some more parameters
46FC 7078 00AO  lda r7,stime       ;timeout rate
4700 4A78 0078  add r7,#bltrat
4704 3078 BDBE  sta r7,m3#bltto    ;always just one word
4708 48F2       set m3#bltsiz = #words
470A 3078 BDC2
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 116
DDT.PLR:1 PAGE 27.1 DDT Code

```
470E 7078 00B2      set m3#/b1tbfm = mapddt      ;put answer onto ddt page buffer
4712 3078 BDC6
4716 4878 9B3C      set m3#/b1tbfa = #m2#/ddtbuf   ;address of buffer
471A 3078 BDC8
471E 6006      endroutine setbit
```

```
.comnt |
BLTDO to start a blt operation and wait until it is
    complete before returning
r1: bits for BLTST at entry, returns finished BLTST
|
4720 1076      routine bltdo, arg r1, result r1, arg m3, result m3/m2
                repeat
4722 3008 6272      set ddpoke          ;tell poller to poke blt
4726 3018 BDBC      sta r1,m3#bltst   ;activate blt as directed
472A 3008 BDBA      unlock bltlok     ;unlock and let it start
                    setmap m3,mapvar  ;for ddtlok $$$maybe don't need if unlock/m1
472E 7078 0000
4732 3078 FC06      repeat
4736 4078 429E      call dsleep        ;rock a bye ddt
473A 7078 0000      setmap m2,maprel  ;through m2 so we can check
473E 3078 FC04
4742 7018 9DBC      lda r1,m2#bltst  ;ready?
4746 99F8      until even  ;;bltact  ;until ready
                endrepeat
                setmap m3,maprel  ;so can lock bltlok
4748 7078 0000      lock bltlok      ;do so
474C 3078 FC06
4750 7078 BDBA
4754 9AFE
4756 7018 9DBC      lda r1,m2#bltst  ;got to check again
475A 99E4      until even  ;;bltact
                endrepeat
475C 6006      endroutine bltdo
```

```
.comnt |
BLTGET gets control of blt parameters when blt is free (not busy)
returns with bltlok locked and m2.m3 set to maprel, r1 has BLTST
|
routine bltget, result m2/m3/r1
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 118
DDT.PLR;1 PAGE 28.1 DDT Code

475E 1076

```
repeat
    repeat
        setmap m2.maprel      :get to blt's page
        lda r7,m2#bltst       ;check if busy
        while odd   ;:bltact
            call dsleep         ;sleep until free
        endrepeat
        setmap m3.maprel      :to lock bltlok
        lock bltlok
        lda r1,m2#bltst       :check again
        while odd   ;:bltact
            ;lost it}
            unlock bltlok        ;try again later
            setmap m3,mapvar     ;fix for sleep
        endrepeat
    endroutine bltgt
```

.comnt | Character at a time IO routines for DDT |

4798 1076 routine ddtout, arg r1
479A 9003 entry iddtout, inline r1
479C 6017
479E 1076
repeat
repeat
 while byte d2fb ;can't give char away yet
47A0 7878 626C
47A4 9AO4
47A6 4078 429E call dsleep
47AA 90FB endrepeat
47AC 7078 A25E lock d2f1
47B0 9AFE
47B2 7878 626C while byte d2fb ;lost while locking?
47B6 9AO4
47B8 3008 A25E unlock d2f1
47BC 90F2 endrepeat
47BE 3818 626C stab r1,d2fb ;give char away
47C2 7078 A26E set @pid = m3//d2fpok
47C6 30F8 00AC
47CA 3008 A25E unlock d2f1
47CE 6006 endroutine ddtout

47D0 1076 routine ddtget, result r1
repeat
 if debugm .bit. procbt
47D2 7078 030C
47D6 7778 00A8
47DA 9AO7
47DC 7818 0160 1dab r1,b1tmyc ;anything happen?

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 120
DDT.PLR;1 PAGE 29.1 DDT Code

```
47E0 4B97      and r1,=<locqut>1ocilo>1ocfad>
47E2 9A03      ifnot zero
47E4 4078 42F2  call ddtste      ;yes--report&clear
                  endif
                  endif
                  repeat
47E8 7878 6276  until byte f2db      ;buffer not empty
47EC 8A04      call dsleep
47EE 4078 429E  endrepeat
47F2 90FB      lock f2d1      ;lock input buffer
47F4 7078 A260
47F8 9AFE
47FA 7818 A276 1dab r1,f2db+foo    ;get input
47FE 8A04      while zero      ;nothing yet?
4800 3008 A260  unlock f2d1      ;try again later
4804 90E7      endrepeat
4806 7078 A278 1da r7,m3#/f2dpok   ;poke ddt fake host
480A 30F8 00AC  sta r7,@pid
480E 3008 A260  unlock f2d1
4812 4B18 007F  and r1,= H7F      ;we found a character
4816 4078 429E  call dsleep
481A 6006      endroutine ddtget
```

```

.commt | Multiple character routines for ddt and others | '
                                ;type text from r2 until zero
481C 1076      routine textout, arg r2, uses r1/r2
481E 9003      entry itextout, inline r2, uses r1/r2
4820 6027
4822 1076

                                repeat
4824 6812      1dab r1,(r2)-
4826 9A04      until zero
4828 4078 4798   call ddout
482C 90FC      endrepeat
482E 6006      endroutine textout

4830 1076      routine numout, arg r1, uses r1/r4/r5
4832 3018 62CC   sta r1.lastnum           ;save numbers output
4836 4078 103C   call hexout, ddout       ;type in hex using ddout
483A 4798
483C 6006      endroutine numout

.commt | this routine to convert word into ascii hex numbers
and send result to output routine. output routine to use is
inline argument, number to print is r1 |
                                page LCode    ;used by many
103C 1076      routine hexout, arg r1, local r2, inline r2, uses r1/r4
103E 1026
1040 6027
1042 307E 0002
1046 48C4      1da r4,#4
                                repeat
1048 A394      r11 r1,4
104A 1016      push r1

```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 122
DDT.PLR;1 PAGE 30.1 DDT Code

```
104C 4B9F      and r1,#0F          ;digit at a time
104E 4E9A      if r1 >= #OA        ;alphabetic digit
1050 9202
1052 4A97      add r1,#'A-'9-1   ;make it ascish
                endif
1054 4A18 0030 add r1,#'0          ;make ascii
1058 4072      call (r2)         ;send it wherever
105A 6016      pop r1           ;get remaining digits
105C 49C1      sub r4,#1        ;count digits
105E 8AF5      until zero       ;until all done
                endrepeat
1060 6026      endroutine hexout
1062 6006
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
IMPOPS.PLR;1 PAGE 1.5 DDT Code

PAGE 123

.INSERT "OPSY" .
.INSRT OPSYS

```
;OPSYS.PLR
;Pluribus Operating System Code.
;Base Dispatch and Routines to handle it.

        Page V2

;*** This really depends on the hardware pid levels ***

;Base dispatch table, indexed by pid.
;Pids 0, FC, and FE are reserved by the system.

80C0      base: .blkw D128
0100      1base=-base ;base length

;Mblk is a table parallel to base. Used by modems, hosts
;and fakes, plus others.

81C0      mblk: .blkw D128

        page Vars

;fakpid: .blkw 1 ;simulate PIDs here if nothing else to do
;idlec: .blkw 1 ;how many times PID empty

62CE      watch1: .blkw 1 ;pointer for first console lights (address)
62D2      watm1: .blkw 1 ;and map for same
62D4      watch2: .blkw 1 ;ptr for console data lights
62D6      watm2: .blkw 1 ;and its map

62D8      watchs: .blkw 1 ;word for procs to blink their lights
```

```
;IMP local code

defloop loopmv           ;entry from STAGE

Page Lcode

;Special dispatches from LOOP

routine bad,nosave,arg r1,arg r4

1064 E042      Trap 102,<;got illegal pid value - page 125>

1066 4807      endroutine bad

routine nopids,nosave,uses r1

.comnt |       ;someday
    lda r1,=2
    subbm r1,fakpid+1
    ifnot zero
        ldab r1,fakpid+1
        lda r3,base(r1)
        cmp r3,=bad
        bne loop4  ;good dispatch
    endif
|
1068 4891      lda r1,=1
106A 3218 62CE  adddm r1,idlec

106E 4807      endroutine nopids
```

P Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 126
OPSYS.PLR;1 PAGE 3 DDT Code

```
1070 00B4      defm0: mapddt+words ;default map 0 setting
1072 7028 00D0  1oopmv: 1da r2,mapvar ;Enter here to set up all maps
1076 7038 00D2  1da r3,mapv2
107A 3028 FC02  sta r2,%map1
107E 3028 FC06  sta r2,%map3
1082 3038 FC04  sta r3,%map2
1086 70A8 1070  1oopm: 1da r2,@defm0 ;:mapcod ;reset default map 0
108A 3028 FCO0  sta r2,%map0
108E 7068 030C  1oop:   1da r6.debugm ;*$ debugging mode?
1092 7768 00A8  tst r6.procbt ;** for me?
1096 9A37       bz loop1 ;*$ no, proceed
1098 7068 0160  1da r6,b1tmvc ;*$ am I "halted"?
109C 9960       bo gosj6 ;:pchalt ;*$ yes, no run sys
109E 7048 6208  1da r4,dhalt ;*$
10A2 4E48 1ADE  cmp r4,=dhpass ;*$ all-hlt password?
10A6 8105       bne loop5 ;*$ no password
10A8 48E1       1da r6,=pchalt ;*$ halt ourselves now
10AA 3068 0160  sta r6,b1tmvc
10AE 9057       br gosj6
10B0 7078 0288  1oop5: if 1stack-1stkln-1stkln <> =stkpass
10B4 4E78 DEAD
10B8 9105       ; Trap 104,<;LSTACK overflow - page 126>
                  set 1stack-1stkln-1stkln = =stkpass
10BA 4878 DEAD
10BE 3078 0288
                  endif
10C2 7018 408C  1da r1,s1fptr ;*$
10C6 801D       qutpat loope ;*$
10C8 7028 608C  1da r2,m1//s1fptr ;*$
10CC 801A       qutpat loope ;*$
10CE 7038 808C  1da r3,m2//s1fptr ;*$
10D2 8017       qutpat loope ;*$
10D4 7698 1070  cmp r1,@defm0 ;:mapcod ;*$
10D8 8114       bne loope ;*$
10DA 7628 00D0  cmp r2,mapvar ;*$
```

```
10DE 8111      bne loope    :*$  
10EO 7638 00D2  cmp r3,mapv2  :*$  
10E4 810E      bne loope    :*$  
10E6 7048 A08E 1da r4,s1f1k  :$$*$  
10EA 800B      qutpat loope  :$$*$  
10EC 8A04      if zero     :$$*$  
10EE 7048 A08E  lock r4,s1f1k :$$*$  
10F2 9AFE      endif  
10F4 3048 A08E  sta r4,s1f1k  :$$*$  
10F8 4B48 FEO0  and r4.=mapmsk :$$*$  
10FC 4E42      cmp r4,r2    :$$*$  
10FE 9103      be loop1    :$$*$  
loop1: ; Trap 103.<;map error - page 127>  
1100 4048 0A1C  jsb r4,wst   ;better check the world
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 128
OPSYS.PLR;1 PAGE 4 DDT Code

```
;*** to activate strip time measurement, next instruction
;*** should be JSB R7,MSRPAT (MAPO MUST POINT TO WARM)
1104 70C8 01A4    loop1: lda r4.@1clock ;time to run stage?
1108 7148 01A2    sub r4,1time
110C 8B22          bnm loop2   ;yes
110E 7048 018A    lda r4,consol ;:$
1112 8B15          bnm loop3   ;:$ i don't have a console.
1114 7028 62D2    lda r2,watm1 ;:$ diddle maps now too
1118 3028 FCO4    sta r2,%map2 ;:$
111C 70A8 62D0    lda r2,@watch1 ;:$
1120 8001          qutpat .+2  ;:$
1122 2024          sta r2,(r4)+ ;:$
1124 7028 62D6    lda r2,watm2 ;:$
1128 3028 FCO4    sta r2,%map2 ;:$
112C 70A8 62D4    lda r2,@watch2 ;:$
1130 8001          qutpat .+2  ;:$
1132 3024          sta r2,(r4)  ;:$
1134 7028 00D2    lda r2,mapv2 ;:$
1138 3028 FCO4    sta r2,%map2 ;:$
113C 48C0          loop3: lda r4,=0
113E 4868 02F8    lda r6,_1stack ;support RATMAC Routines
1142 609C 0174    emty: lda r1,@pidget(r4)+ ;try PIDs in order
1146 3018 00A4    lop4:  sta r1,oldp
114A 40F9 80C0    call @base(r1) ;nominal dispatch
114E 909C          br loopm
1150 3008 00A4    loop2: sta r0,oldp ;:$
1154 7048 00A8    stagep: lda r4,procbt ;fake PID level for stage
1158 3548 62D8    eorm r4,watchs ;blink my bit
115C 4008 0836    gosj6: tr sj6
```

: Routine to add an entry to BASE

.COMNT |

The INBASE subroutine accepts a PID level (in R1), a control block address (in r2), and a routine pointer (inline argument). It adds the routine to the table BASE and the control block to MBLKs. If the table entry is the same as the arguments or if it is empty (=BAD, LOOP, or 0), the success return is taken. If a different routine already exists in BASE or MBLKs, the fail return is taken. %map2 is saved and restored.

The calling sequence is:

```
1da r1,<pid level>
1da r2,<control block>
call inbase
<routine>
```

```
|
```

| | |
|----------------|--|
| 1160 1076 | routine inbase, arg r1-r2,inline r3,local m2 |
| 1162 7078 80BE | |
| 1166 1076 | |
| 1168 707E 0002 | |
| 116C 6037 | |
| 116E 307E 0002 | |
| 1172 9005 | entry inbas2,arg r1-r3,local m2 |
| 1174 1076 | |
| 1176 7078 80BE | |
| 117A 1076 | |
| 117C 7078 00D2 | 1da r7,mapv2 ; address base, mblk |
| 1180 3078 FCO4 | sta r7,%map2 |
| 1184 7079 80CO | 1da r7,base(r1) :get current base entry. |
| 1188 9A14 | ifnot z } r7 ==loop } r7 ==bad } r3 = r7 & r2 = mblk(r1) |
| 118A 4E78 108E | |
| 118E 9111 | |
| 1190 4E78 1064 | |
| 1194 910E | |
| 1196 4E37 | |
| 1198 8104 | |

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
OPSYS.PLR;1 PAGE 5.1 DDT Code

PAGE 130

119A 7629 81CO
119E 9109
11AO 6076 fail return
11A2 707F OOBO
11A6 3078 FC04
11AA 6076
11AC 4FF0
11AE 4007
11BO 3029 81CO endif
11B4 3039 80CO sta r2,mb1ks(r1)
11B8 6076 endroutine
11BA 707F OOBO
11BE 3078 FC04
11C2 6006

```
;subroutine to see if a device is in useio
;r7 return, r1 device address, r2,r3 clobbered
;low-order 4 bits of device address ignored
;map0 is saved and restored,using temp4

11C4 7038 408C findev: lda r3,m0+<s1fptr&packm>           ;save m0.
11C8 3038 0184          sta r3,temp4
11CC 7038 00B0          lda r3,maprel
11D0 3038 FCOO          sta r3,%map0             ;setup map0 for routine.
11D4 48B8              lda r3,=useiol  ;search fake also
11D6 4821          finde1: lda r2,r1    ;device we seek
11D8 512B 4116          sub r2,ibase(-r3)   ;base of this segement
11DC 4F28 FFO0          tst r2,=- D16* H10   ;in these 16 devices?
11E0 9A04              bz finde2      ;yes, now look at the bit
11E2 88FA              bnlp finde1     ;no, check more
11E4 48AO              lda r2,=0       ;zero cc for fail
11E6 9006              br findex

11E8 A6A3          finde2: srl r2,3      ;convert to bittab index
11EA 702A 0EB6          lda r2,bittab(r2)  ;get right bit
11EE 732B 5E98          and r2,useio(r3)  ;in useio?
11F2 7038 0184          findex: lda r3,temp4
11F6 3038 FCO0          sta r3,%map0      ;restore map0
11FA 4822          lda r2,r2       ;set condition code
11FC 4007          jmp (r7)
```

```
;DOINIT

;      The DOINIT routine performs table-driven
;initialization for the POPS/IMP system.  The
;argument for DOINIT is a pointer (through M0)
;to an initialization table on the page to
;be initialized.  The table has the general
;form

;      <cmd block>
;      <arg>
;      <arg>
;      ...
;      <arg>
;      <cmd block>
;      <arg>
;      ...
;      <nil>

;where the <cmd block> takes one of two forms.
;The most common form is a routine address
;which is just coded in-line.  Each routine
;is passed the current table pointer
;in R1 and is responsible for reading
;its own arguments.

;The other possible form for the <cmd block>
;is the pair 0 / <chain ptr> which
;indicates that the initialization should
;be continued from <chain ptr>.
```

;The DOINIT routine

11FE 1076 ROUTINE DOINIT, ARG R1, LOCAL MO
1200 7078 40BE
1204 1076
1206 4851 LDA R5,R1 ;Move arg pointer to R5
 REPEAT ;For each routine or ptr
1208 6075 LDA R7,(R5)+ ;Get the next word
120A 9919 UNTIL NIL ;Odd end marker
120C 8AOB IF ZERO ;If this is a jump block
120E 6075 LDA R7,(R5)+ ;Get new page type
1210 8903 IF ODD ;Should we set a map?
1212 7055 LDA R5,(R5) ;No, just get new address
1214 9006 ELSE ;Move to the new address
1216 7055 LDA R5,(R5) ;Move to new map
1218 707F 00BO SETMAP MO,LMAP(R7) ;Move to new map
121C 3078 FCOO
 ENDIF
1220 900D ELSE ;Use standard variables page
1222 7078 0ODO SETMAP <M1,M3>,MAPVAR
1226 3078 FCO2
122A 3078 FCO6
122E 7078 0OD2 SETMAP M2,MAPV2 ;And extra variables page
1232 3078 FCO4
1236 40FD FFFE CALL @-2(R5) ;Call the routine
 ENDIF
123A 90E7 ENDREPEAT
123C 6076 ENDROUTINE DOINIT
123E 707F 00BO
1242 3078 FCOO
1246 6006

;Routine to initialize from the \$INIT table

| | | |
|----------------|--------------------------|---------------------------|
| 1248 1076 | ROUTINE INITLIST, ARG R5 | |
| | REPEAT | ;For each block in list |
| 124A 6025 | LDA R2,(R5)+ | ;Get address of block |
| 124C 990E | UNTIL NIL | ;Are we at end |
| 124E 8A07 | IF ZERO | ;Is this a map change |
| 1250 6025 | LDA R2,(R5)+ | ;Load which map |
| 1252 6035 | LDA R3,(R5)+ | ;And logical page |
| 1254 707B 00BO | LDA R7,LMAP(R3) | ;Convert to physical page |
| 1258 3072 | STA R7,(R2) | ;Store in map |
| 125A 9006 | ELSE | ;Normal reference |
| 125C 6075 | LDA R7,(R5)+ | ;And count |
| 125E 6035 | LDA R3,(R5)+ | ;And initial value |
| | REPEAT | ;For each cell in block |
| 1260 2032 | STA R3,(R2)+ | ;Initialize the word |
| 1262 49F2 | SUB R7,=2 | ;Count two more bytes |
| 1264 8AFE | UNTIL ZERO | ;Until we hit zero |
| | ENDREPEAT | |
| | ENDIF | |
| 1266 90F2 | ENDREPEAT | ;Back for more |
| 1268 6006 | ENDROUTINE INITLIST | |

:Routine to initialize the BASE table

```
126A 1076      ROUTINE INITPID
                REPEAT
126C 6015          LDA R1,(R5)+           ;Get next PID level
126E 9907          UNTIL NIL
1270 48A1          LDA R2,=NIL
1272 6035          LDA R3,(R5)+           ;And dispatch entry
1274 4078 1174    CALL INBAS2            ;Enter into the base table
1278 8A01          IF FAIL              ;Did this fail
                    ;     Trap 105.<:INBASE failed - page 135>
                    ;ENDIF
127A 90F9          ENDREPEAT
127C 6006          ENDROUTINE INITPID
```

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
OPSYS.PLR;1 PAGE 11 DDT Code

PAGE .136

;Timer routines for PID poking

127E 1076 ROUTINE D025.6, LOCAL R1
1280 1016
1282 7018 0316 LDA R1,L\$25.6
1286 4078 129E CALL DOCLOCK
128A 6016 ENDROUTINE D025.6
128C 6006

128E 1076 ROUTINE D01.6, LOCAL R1
1290 1016
1292 7018 0318 LDA R1,L\$1.6
1296 4078 129E CALL DOCLOCK
129A 6016 ENDROUTINE D01.6
129C 6006

129E 1076 ROUTINE DOCLOCK, ARG R1
REPEAT :For each item in clock list
 UNTIL R1 = =NIL ;Until we reach the end
12A0 4E91
12A2 9106
12A4 6071 LDA R7,(R1)+
12A6 30F8 OOAC STA R7,@PID
12AA 7011 LDA R1,(R1)
12AC 90FA ENDREPEAT
12AE 6006 ENDROUTINE DOCLOCK

;Build the initial table entry (blank)

12B0 1248 \$ITABLE
12B2 0001
12B4 126A
12B6 0001
12B8 0001

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
OPSYS.PLR;1 PAGE 11.1 DDT Code

PAGE 137

12BA 0000
12BC 0000

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42
IMPOPS.PLR;1 PAGE 1.6 DDT Code

PAGE 138

297

;Next, finish up the STAGE stuff

FC00 0000 \$finstage
40DE 7D50
40EO 7B00
40E2 FFFF
40E4 FFFF
40E6 FFFF
40E8 FFFF
40EA FFFF
40EC FFFF
40FA 0001
40FC 0001
40FE 0000
4100 4000
4104 0010
4106 0020
4108 0030
410A 0050
410C 0002
410E 0012
4110 0022
4112 002E
4114 0002
4116 E100
4118 E200
411A F100
411C F200
5016 5020
5018 5056
501A 53EA
501C 5496
501E 1072
40BA 0000
40BC 0000
031A 04DC
0314 00C8
0316 0001
0318 0001
0320 E000
0322 F000
0324 8000

IMP Operating System PLURIBUS V2.9B 25-Jun-87 11:22:42 PAGE 139
IMPOPS.PLR;1 PAGE 1.7 DDT Code

0326 8000

;Now save our symbols
.iif p2, .outsym impops.sym
contload ;load system onto us

64 SECONDS RUN-TIME



299

CONCORDANCE FOR 1: <PSE-1301-1>IMPLOD.COM.1, 3-Dec-86 15:56:16
2: <PSE-1301-1>IMPOPS.COM.1, 10-Dec-86 13:29:59
3: <PSE-1301-1>IMP.COM.2, 25-Jun-87 11:05:10

A - 1:3, 1:3, 1:3, 1:3, 1:3, 1:3, 1:3, 1:7, 1:7, 1:8, 1:8, 1:8, 1:10,
1:10, 2:7, 2:98, 2:98, 2:105, 2:122
ABUS 1:68 1:78
ACCRES 2:105 2:104
ACT - 2:32
ACTIVA 2:23 2:23, 2:23
ADDR - 2:4, 2:4, 2:4
AGAIN - 2:7
AGE 3:32 3:263, 3:263, 3:265, 3:265, 3:277, 3:288, 3:308, 3:308, 3:308,
3:308, 3:309, 3:309, 3:310, 3:310, 3:311, 3:313, 3:404, 3:404,
3:405
AGEO 3:32 3:32, 3:261, 3:264, 3:265, 3:265, 3:277, 3:280, 3:288, 3:288,
3:308, 3:310, 3:310, 3:310, 3:310, 3:404, 3:404
AGEING 3:404 3:403
AGETM 3:404 3:403
AGETX 3:404 3:404, 3:404
AKBITS 3:18 3:312
ALLOC 3:18 2:23, 3:274, 3:305, 3:305
ALLOW - 2:32
ALTIO 3:11, 3:13 3:108, 3:108, 3:113, 3:113, 3:113, 3:113, 3:113
AMEMIO 1:68 1:73, 1:73, 1:73, 1:78
AMPCom 1:70 1:78, 1:81
AMPMan 1:68 1:73, 1:78, 1:78, 1:78
AMPROC 1:70 1:78, 1:81, 1:82, 1:82, 1:93
AMPSYS 1:68 1:73
AMPTRY 1:68 1:73
AN - 2:32
ANOM 2:78 2:98, 2:107, 3:65, 3:65, 3:316, 3:316, 3:362, 3:362, 3:375,
3:402, 3:411, 3:411
ANOMSP 3:361
ANSWER 2:112 2:110, 2:111, 2:112
ANYPRO 1:16 1:92, 1:109, 1:112, 1:114, 1:117, 1:117, 2:36, 2:68
APROC 1:68 1:78
AR - 1:23
ARE - 2:7
ARG1 - 1:4, 1:4, 1:5, 1:5, 1:9, 1:9, 2:4, 2:4, 2:4, 2:4, 2:15, 2:15, 2:20,
2:20, 2:20, 2:20, 2:21, 2:21, 2:21, 2:21, 2:21, 2:27,
2:27, 2:32, 2:32, 2:32, 2:32, 2:32, 2:33, 2:33, 2:33, 2:33,
2:33
ARG2 - 1:4, 1:4, 1:5, 1:5, 1:9, 1:9, 2:4, 2:4, 2:4, 2:4, 2:15, 2:15, 2:21,
2:21, 2:21, 2:21, 2:21, 2:27, 2:27, 2:33, 2:33, 2:33, 2:33,
2:33
ARG3 - 1:4, 1:4, 1:5, 1:5, 2:4, 2:4, 2:4, 2:4, 2:15, 2:15, 2:27, 2:27
ARG4 - 1:4, 1:4, 1:5, 1:5, 2:15, 2:15, 2:27, 2:27
ARG5 - 1:4, 1:4, 2:15, 2:15, 2:27, 2:27
ARGRES 2:105 2:94, 2:108, 2:111, 3:64, 3:69
ARGUME - 2:29
ARPANO 3:9 3:9
ARSTAC - 1:26
ASCICO 2:76 3:66
ASCICR 2:76 2:108, 3:80, 3:345
ASCIFF 2:76 3:79
ASCILF 2:76 2:88, 2:108, 3:80, 3:345
ASCISP 2:76 2:96, 3:79, 3:79

ASSUME - 2:29
ATTNPA 1:24 1:11,1:33
AUXCNT 3:11 3:131,3:131,3:138,3:139,3:147
B1 - 3:8
B2 - 3:8
B2PBLK 3:320 3:320,3:320,3:422,3:425,3:427
B2PEND 3:320 3:320,3:422,3:427
B3 - 3:8
B4 - 3:8
B5 - 3:8
B510 3:301 3:301,3:302
B511 3:303 3:303
B513 3:304 3:303,3:303
B514 3:303 3:302,3:303
B515 3:304 3:303,3:303
B516 3:302 3:302
B517 3:303 3:303
B518 3:304 3:304
B519 3:302 3:302
B51B 3:304 3:304
B51D 3:302 3:302
B51E 3:302 3:302
B51F 3:303 3:302
B51G 3:303 3:302
B51T 3:302 3:302
B52 3:301 3:301
B53 3:301 3:301
B54 3:301 3:301
B54A 3:301 3:301
B55 3:303 3:302
B57 3:303 3:303
B58 3:302 3:301
B59 3:301 3:301
B6 - 3:8
B62 3:306 3:306
B64 3:306 3:306
B65 3:306 3:306
B66 3:306 3:306,3:306
B67 3:305 3:306
B68 3:305 3:305
B6A 3:305 3:305
B6B 3:305 3:305
B6D 3:305 3:305
B702 3:308 3:308
B71 3:308 3:308
B72 3:308 3:308
B73 3:308 3:308,3:308,3:308,3:308
B74 3:308 3:308
B75 3:309 3:309
B76 3:309 3:309
B77 3:309 3:309,3:309
B79 3:309 3:308
B7SUB 3:310 3:308,3:309
B7SUB1 3:310 3:310
B94 3:313 3:313
B95 3:313 3:313,3:313,3:313
B96 3:313 3:313,3:313
B97 3:313 3:313,3:314

B98 3:314 3:313, 3:313
BACK 3:300 2:7, 2:7, 3:425
BACK5 3:301 3:303, 3:304, 3:425
BACK6 3:306 3:306, 3:425
BACK7 3:308 3:309, 3:425
BACK7J 3:309 3:309
BACK9 3:313 3:425
BAD 2:125 2:125, 2:129, 3:97, 3:414
BADLIT 3:401 3:401, 3:401
BAKLEN 3:15 3:34, 3:34, 3:34, 3:34
BAKSET 3:311 3:302, 3:302, 3:303
BANOM 3:360 3:375, 3:402
BASE 2:124 2:124, 2:128, 2:129, 2:130, 3:97, 3:97, 3:97, 3:410, 3:410,
3:410, 3:414, 3:425, 3:428, 3:428
BASEC 3:426 3:428, 3:428
BASEP 3:426 3:428, 3:428
BASEP2 3:426 3:428
BASEPO 3:426 3:428, 3:428
BASETO 3:428 3:421
BASETP 3:428 3:380, 3:428
BBCANS 1:17 1:81
BBCBAD 1:71 1:95
BBCBAK 1:17 1:82, 1:93
BBCFOR 1:17 1:78, 1:78, 1:81
BBCLOK - 1:70, 1:73, 1:81, 1:81, 1:82, 1:83, 1:84, 1:93, 1:94, 1:95
BBCLMAP 1:17 1:82, 1:93, 1:93, 1:94, 1:94, 1:94, 1:94
BBCMSK 1:17 1:17, 1:82, 1:82, 1:93, 1:93, 1:93, 1:94, 1:94, 1:94, 1:94,
1:94, 1:95
BBCODD 1:17 1:93, 1:93, 1:94
BBCPAS 1:17 1:78, 1:78, 1:81
BBCRES 1:17
BBCRST 1:71 1:94
BBCWIN 1:17 1:28, 1:79, 1:80, 1:82, 1:93, 1:93, 1:93, 1:94, 1:94, 1:94,
1:94, 1:94, 1:94, 1:95
BBCWMK 1:17 1:82, 1:93, 1:94, 1:94, 1:94, 1:94, 1:94, 1:94, 1:94,
1:95
BBKO 3:34 3:320, 3:425
BBK1 3:34 3:320, 3:425
BBK2 3:34 3:320, 3:425
BBK3 3:34 3:320, 3:425
BBN63 3:9 3:46
BBNTIP 3:9
BCOMT 3:425 3:427
BD - 1:23
BDATA 3:15 3:307, 3:311
BDSTAC - 1:26
BDSTH 3:15 3:312
BEG 1:23
BEGIN - 2:107, 2:110, 3:52, 3:169, 3:390
BESCLK 1:3 3:120
BFAIL - 3:247, 3:248, 3:251, 3:260, 3:276, 3:322
BFIJC 3:312 3:311
BFIJR 3:312 3:309
BFIJT 3:312 3:305, 3:305, 3:305, 3:308, 3:314
BFMPND 3:414 3:389, 3:412, 3:412
BGETA 3:304 3:302, 3:302, 3:303
BGETAO 3:304 3:304
BGETA1 3:304 3:304

BLTBB4 1:93 1:94
BLTBB5 1:94 1:93
BLTBB6 1:94 1:93
BLTBB7 1:94 1:94, 1:94, 1:94, 1:94
BLTBB9 1:93 1:93
BLTBBC 1:93 1:92
BLTBBK 1:95 1:93
BLTBBQ 1:95 1:93, 1:93, 1:93
BLTBBR 1:95 1:94
BLTBBS 1:94 1:93
BLTBBT 1:95 1:94
BLTBBY 1:95 1:95
BLTBBZ 1:94 1:94
BLTBCQ 1:86 1:85, 1:95
BLTBFA 1:71 1:87, 1:87, 1:114, 2:68, 2:109, 2:116
BLTBFM 1:71 1:87, 1:114, 2:116
BLTBGO 1:86 1:86
BLTBIT 1:90 1:86
BLTBMK 1:71 1:86, 2:109, 2:112
BLTBMQ 1:86 1:85, 1:88, 1:88, 1:88, 1:90, 1:95
BLTBPR 1:89 1:86, 1:90
BLTBU1 1:85 1:85
BLTBUC 1:86 1:86, 1:86, 1:94
BLTBUD 1:85 1:92
BLTBUE 1:86 1:86, 1:91
BLTBUF 1:71 2:68
BLTBUG 1:86 1:86
BLTBUS 1:86 1:85
BLTBUT 1:86 1:86
BLTBUZ 1:86 1:86
BLTC1 1:88 1:89
BLTC2 1:89 1:91
BLTC3 1:88 1:85, 1:88, 1:88, 1:88, 1:91, 1:91, 1:91, 1:91
BLTC4 1:88 1:89
BLTC6 1:89 1:89, 1:89, 1:89
BLTC7 1:89 1:91
BLTCAL 3:84 3:426
BLTCC 1:90 1:88
BLTCC1 1:90 1:90, 1:90, 1:90
BLTCCK 1:71, 1:98 1:90, 1:98, 1:107, 1:114, 2:107, 2:107
BLTCE 1:90 1:90, 1:90, 1:90
BLTCE1 1:90 1:88
BLTCE2 1:90 1:90, 1:90
BLTCEQ 1:88 1:90
BLTCF 1:88 1:88
BLTCOM 1:89 1:87
BLTCTL 1:71 1:86, 1:86, 1:89, 1:93, 1:94, 1:94
BLTDDT 1:71 1:87
BLTDID 1:71 1:95
BLTDO 2:117 2:107, 2:110, 2:117
BLTDON 1:71 1:86, 1:87, 1:88, 1:90
BLTDPI 1:71 1:86, 1:89, 1:114, 2:68, 2:115
BLTDPM 1:71 1:89, 1:90, 1:92, 1:114, 2:115
BLTDSK - 2:113
BLTDTY 1:71 1:87, 1:87, 1:114, 2:68, 2:115
BLTEDF 1:71 1:71, 1:71, 1:91, 2:113
BLTEDK 1:71 1:71, 1:71, 1:86
BLTEDQ 1:71 1:71, 1:71, 1:86, 2:113

TEND 1:90 1:90
 BLTEOR 1:89 1:86,1:87,1:90
 BLTERS 1:71,1:98 1:98,1:108,1:110,1:110,1:111,1:111,2:109,2:112,
 2:112
 BLTESF 1:71 1:71,2:113
 BLTESK 1:71 1:71,2:113
 BLTESQ 1:71 1:71,2:113
 BLTETO 1:71 1:71,1:87,2:112
 BLTFAK 1:71 1:87,1:114,3:353,3:357
 BLTFDN 1:90 1:87
 BLTFER 1:91 1:92,1:92,1:93
 BLTFEX 1:71 1:87,1:103,1:104,1:104,1:114
 BLTFGO 1:87 1:87
 BLTFUL 1:71 1:87,1:89,1:90,1:114,2:107,2:107
 BLTGET 2:117 2:107,2:107,2:110,2:110,2:118
 BLTLOG 1:71 1:89,1:89,1:117,1:117,2:115
 BLTLOK - 1:71,1:73,1:87,1:89,1:114,1:114,1:114,2:68,2:68,2:109,
 2:112,2:117,2:117,2:118,2:118
 BLTM1 1:91 1:91
 BLTM2 1:91
 BLTM3 1:91 1:91
 BLTMAX 1:16 1:87,1:87,1:87
 BLTME 1:91 1:92
 BLTME1 1:91 1:91
 BLTMK - 2:113
 BLTMSK 1:89 1:86,1:87,1:88
 BLTMYB 1:90 1:89,1:89
 BLTMYC 1:21 1:39,1:45,1:50,1:51,2:60,2:96,2:96,2:119,2:126,2:126
 BLTMYM 1:21 1:91,1:91
 BLTMYR 1:21 1:89,1:91,2:96,2:96
 BLTMYS 1:89 1:90
 BLTNLC 1:71 1:87,1:89,1:109,1:112,1:117,2:115
 BLTPCH 1:92 1:87
 BLTPCO 1:92 1:92,1:92
 BLTPCS 1:92 1:92
 BLTPID 3:24 3:63,3:84,3:426
 BLTPOK 1:71 1:90,3:353,3:357
 BLTPRM 1:95 1:88,1:91
 BLTPRO 1:71 1:86,1:89,1:94,1:95
 BLTPRS 1:95 1:85,1:93
 BLTPRX 1:95 1:95,1:95
 BLTRAT 1:16 1:89,1:114,2:68,2:115,3:63
 BLTRUN 1:71 1:86,1:93,1:94,2:61
 BLTSDF 1:71,1:98 1:88,1:88,1:98,1:107,1:110,1:110,1:114,2:110,
 2:110
 BLTSIZ 1:71 1:87,1:90,1:114,2:68,2:109,2:115
 BLTSPA 1:71 1:89,2:107,2:110
 BLTSPM 1:71 1:86,1:92,1:114,2:68,2:115
 BLTST 1:71 1:86,1:86,1:87,1:88,1:89,1:93,1:94,1:108,1:110,1:114,
 1:114,2:60,2:68,2:68,2:107,2:110,2:110,2:112,2:117,
 2:117,2:117,2:118,2:118,3:84
 BLTSTB 1:16 1:46
 BLTSTY 1:71 1:87,1:87,1:114,1:114,2:68,2:115
 BLTTCH 1:87 1:92,1:92,1:92,1:92
 BLTTO 1:71 1:87,1:89,1:114,2:68,2:115
 BLTTOG 1:89 1:86,1:90
 BLTX 1:89 1:87,1:87
 BMESSB 3:15 3:300,3:301,3:301,3:304,3:306,3:312

BMIDH 3:15 3:311,3:312,3:312,3:314
BMINUS - 3:322
BODD - 3:255
BPKTH 3:15 3:302,3:303,3:304,3:305,3:312,3:312,3:314
BREINI 3:425 3:422,3:425
BSADIL 1:25 1:64,1:73,1:79,1:80,1:81,1:93,1:102,3:103,3:120,3:410
BSADIT 1:3 1:11,1:25
BSADML 1:25 1:73,1:81
BSADMT 1:3 1:11,1:25
BSADRM 1:25 1:25,1:81,1:81
BSADRS 1:25 1:11,1:25,1:64,1:74,1:80,1:81,1:81,1:93,1:102,1:102,
3:103,3:410
BSEND 3:307 3:304,3:305,3:309,3:314
BSEQH 3:15 3:302,3:302,3:303,3:307,3:309,3:311,3:312,3:312,3:314
BSMAPM 1:69 1:81,2:38,2:38,2:39,2:40,2:43
BSMAPS 1:69 1:11,1:74
BSMPTB 1:3 1:11,1:69
BSMPTM 1:3 1:11,1:69
BTO 3:403 3:380
BTO1 3:404 3:404
BTO2 3:404 3:404
BTO4 3:404 3:404,3:404,3:404
BTOA1 3:28
BTOA5 3:28
BTOA7 3:28
BT1 3:403 3:403
BT2 3:403 3:403
BT3 3:403 3:403
BT4 3:405 3:405
BT6 3:405 3:404,3:404,3:404,3:404
BTC 3:320 3:320,3:426
BTYPH 3:15 3:307,3:312
BUFB 3:17 3:91,3:283,3:285,3:322,3:322
BUFCYC 3:390 3:390
BUFE 3:17 3:86,3:86,3:153,3:155,3:156,3:157,3:175,3:221,3:260,
3:282,3:282,3:283,3:285,3:290,3:300,3:307,3:322,3:332,
3:338,3:352,3:352,3:355
BUFEND 1:97,3:17 1:104,3:154,3:155,3:156,3:241,3:247,3:257,
3:260
BUFFLG 1:72 2:41,3:99
BUFINI 3:412 3:412,3:422
BUFLEN 3:17 3:376,3:388,3:388,3:389,3:389,3:412,3:413,3:413
BUFMAP 3:414 3:389,3:389,3:412
BUFSIZ 2:78 2:78,2:90,2:91
BUFT 3:390 3:380,3:390
BUFTIC 3:390 3:390,3:390
BUFTIM 3:27 3:390,3:390
BUSCON 1:70 1:27
BUSFIX 1:70 1:27
BUSINC 1:17 1:102
BUSKIL 1:25 1:33,1:64,1:73,1:73,1:73
BUT - 2:32
C - 3:64,3:72
C.VD 3:47 3:47,3:47
CABORT 1:96 1:107,1:111
CANOM 3:361 3:362
CARRY - 1:31,1:35,2:70,3:117,3:169,3:169,3:170,3:175,3:178,
3:268,3:268,3:359,3:370,3:370

| | | |
|---------|-----------|---|
| CAWL | 3:361 | 3:361,3:362,3:362 |
| CAWLSP | 3:361 | |
| CBLOCK | 3:21 | 3:251,3:252 |
| CCA | 3:9 | 3:40 |
| CCADDR | 1:97 | 1:107,1:107 |
| CCBASE | 3:47 | 3:47,3:47,3:97 |
| CCBSPA | 3:47 | |
| CCCHECK | 3:47 | 3:47,3:47,3:97 |
| CCHSPA | 3:47 | |
| CCKTAB | 1:68 | 1:88,1:88 |
| CCLEAR | 2:76 | |
| CCLED | 3:62 | 3:64,3:64,3:343,3:343,3:343,3:415,3:415,3:417 |
| CCPIEC | 1:97 | |
| CCSIZE | 1:97 | |
| CCTYPE | 1:97 | 1:107 |
| CD | - | 1:23 |
| CDELAY | 3:175 | 3:149,3:171,3:175 |
| CDESTD | 3:21 | 3:252,3:271,3:291,3:394 |
| CDOWN | 2:76 | |
| CDSTAC | - | 1:26 |
| CERR32 | 3:21 | 3:256 |
| CERRLD | 3:21 | 3:252 |
| CERROR | 3:21 | 3:251 |
| CFORHA | 1:97 | 1:97 |
| CFORHS | 1:97 | |
| CFORI | 1:97 | |
| CFORL | 1:97 | 3:355 |
| CFORMI | 1:97 | |
| CHACC | 3:21 | 3:287 |
| CHAIN | 3:37 | 3:50,3:50,3:50,3:50,3:51,3:51,3:51,3:53,3:53,3:53, 3:54,3:56,3:56,3:56,3:57,3:58,3:59,3:91,3:91,3:91, 3:100,3:100,3:131,3:131,3:141,3:147,3:150,3:168,3:169, 3:169,3:169,3:212,3:215,3:216,3:217,3:217,3:218,3:225, 3:229,3:238,3:238,3:239,3:239,3:249,3:255,3:255,3:278, 3:285,3:286,3:286,3:286,3:290,3:322,3:332,3:332,3:332, 3:332,3:338,3:352,3:354,3:388,3:388,3:388,3:388,3:389, 3:389,3:389,3:390,3:390,3:390,3:390,3:391,3:391,3:392, 3:392,3:412,3:422 |
| CHAN | 3:37 | 3:56,3:91,3:91,3:91,3:147,3:150,3:168,3:169,3:169, 3:182,3:212,3:215,3:217,3:218,3:225,3:229,3:255,3:278, 3:285,3:286,3:290,3:352,3:354,3:388 |
| CHANGE | - | 2:6,2:7 |
| CHANNE | 3:43 | 3:43,3:136 |
| CHANUM | 3:18 | 3:146,3:151,3:167,3:167 |
| CHCUP | 3:21 | 3:311 |
| CHECK | - | 2:7,2:29 |
| CHINI | 3:21 | 3:13 |
| CHKDUP | 3:274 | 3:274 |
| CHKFRE | 3:391 | 3:390,3:391 |
| CHKH | 1:97,3:17 | 1:103,3:144,3:144,3:146,3:151,3:152,3:152, 3:164,3:221,3:259,3:260,3:263,3:276,3:278,3:300,3:307, 3:351,3:351,3:352 |
| CHKPNT | 3:388 | 3:390,3:391 |
| CHKTC | 3:319 | 3:317,3:319,3:320 |
| CHNBSY | 3:12 | 3:137,3:148,3:148,3:150,3:150,3:171,3:171,3:382,3:382, 3:382 |
| CHNOTI | 3:21 | 3:294,3:311 |
| CHOSIN | 3:47 | 3:47,3:96 |

CHOST 3:106 3:104,3:106
CHOSTE 3:47 3:47,3:96
CHSTAT 3:21 3:271
CHSTD 3:21 3:252,3:287,3:294,3:296
CILBUF 1:29 1:29
CILCNT 1:29 1:42,1:42,1:43
CILEND 1:29 1:41,1:41,3:72,3:79,3:373
CILLGL 3:21 3:252
CILLOC 1:29 1:43,3:402
CILNUM 1:16 1:29,1:29
CILOPS 1:29 1:41,1:41,3:79,3:79,3:79,3:373
CILOVF 1:29 1:42,1:42,3:72,3:72
CILPRO 1:29 1:42,1:42,1:43
CILREG 1:29 1:43
CIMPD 3:21 3:252,3:296,3:394
CIMPDN 3:21 3:396
CIMPER 3:21 3:251
CINCTR 3:21 3:251,3:252,3:291,3:305
CKCLOC 3:120 3:117,3:120
CKCOUN 3:52 3:52,3:53
CKERRS 3:11 3:131,3:154
CKILL 3:29 3:95,3:99,3:114
CKLOCK - 1:58,3:440
CKNICE 3:122 3:117,3:122
CKPASS 1:16 1:12,1:12,1:12,1:12,1:66,1:90
CKQPUT 3:85 3:85,3:148,3:157
CKSJAM 3:376 3:369,3:369,3:369,3:369,3:369,3:370,3:370,3:371,3:371,
3:371,3:371,3:371,3:371,3:371,3:371,3:372,3:372,
3:372,3:372,3:373,3:373,3:374,3:374,3:374,3:374,
3:375,3:375,3:375,3:375,3:375,3:375,3:375,3:376,3:376,
3:376,3:376
CKSUB 1:65 1:52,1:60,1:60,1:66,2:36,2:45
CKSUM 1:28 1:12,1:60,1:118,2:37,2:44,2:44,2:45,2:50,2:51,3:423,
3:423
CLDRP 3:21 3:246,3:287
CLINEI 3:136 3:132,3:136,3:137
CLK1UP 3:25 3:120,3:120
CLK2UP 3:25 3:120,3:120,3:120
CLKLOK - 3:25,3:120,3:120,3:120,3:120,3:417
CLLED 3:62 3:343
CLOCK 3:25 3:55,3:120,3:120,3:121,3:148,3:152,3:154,3:183,3:184,
3:236,3:260,3:283,3:285
CLOCKM 3:11 3:136,3:175,3:178,3:370
CLOKRT 1:19 1:36,2:62,2:62
CLONG 3:21 3:251
CLOST 3:21 3:305
CLRCHN 3:171 3:169,3:169,3:171
CLRST 3:11
CLST 1:5 1:5,1:6,1:6,1:12
CMAP 1:72 2:38,2:51
CMARK 1:97 1:97,1:106,1:107,1:112,1:112,3:352,3:352,3:355,3:356
CMICHK 3:114 3:105,3:107,3:113,3:114
CMMBEG 1:97 3:352
CMBBIT 1:97 3:352,3:352
CMODEM 3:105 3:104,3:104,3:105
CMSBIT 3:46
CNOP 3:21 3:90,3:90,3:243,3:244,3:252
CNTLC 3:64 3:64,3:64

| | | |
|--------|-----------|---|
| CNTLL | 3:81 | 3:81,3:81 |
| CNTL0 | 3:65 | 3:65,3:65 |
| CNTRS | 3:27 | 3:51,3:52,3:52,3:53,3:164,3:233,3:238,3:278,3:282, 3:283,3:284,3:286,3:295,3:295,3:304,3:304,3:304,3:375, 3:375,3:386,3:386,3:386 |
| COD | - | 3:7,3:7,3:7,3:7 |
| CODE | - | 1:13,1:14,1:14,1:69,2:13,2:34,3:7,3:7,3:7,3:8 |
| CODEPA | 1:5 | 1:13,2:34,3:7,3:7,3:7,3:7,3:7 |
| CODL0D | 3:6 | 3:7,3:7 |
| CODTAB | 3:316 | |
| COMAR | 1:70 | 1:12,1:76,1:77,1:77 |
| COMBIT | 3:18 | 3:144,3:157,3:221 |
| COMMA | 2:104 | 2:101,2:104 |
| COMMON | - | 2:7 |
| COMPAR | 3:179 | 3:176,3:179 |
| COMPAT | 3:18 | |
| COMPTR | 1:28 | 1:47,1:47,1:59,1:59 |
| COMPTV | 3:5 | |
| COMREL | 1:28 | 1:48,1:60,1:61,1:61,1:118,2:51 |
| COMSTS | 1:16 | 1:12 |
| COMTST | 1:28 | 1:59,1:59,1:59,1:59 |
| CON | 3:94 | 3:29,3:426 |
| CONC1 | 3:411 | 3:411 |
| CONC2 | 3:411 | 3:411 |
| CONCER | 3:29 | 3:316 |
| CONCHK | 3:316 | 3:316 |
| CONCKS | 3:40,3:40 | |
| CONCLK | 3:410 | 3:377,3:411,3:421 |
| CONCOM | 1:27 | 1:27,1:75 |
| CONFSP | 3:29 | 3:94,3:94 |
| CONLEN | 3:40 | 3:40,3:316 |
| CONLOK | - | 3:29,3:94,3:94,3:99,3:417 |
| CONLOO | 3:93 | 3:94 |
| CONPID | 3:24 | 3:94,3:99,3:118,3:426 |
| CONSOL | 1:22 | 1:45,1:52,1:105,2:128 |
| CONSTA | - | 3:94,3:94 |
| CONSUB | 3:104 | 3:98,3:98,3:104 |
| CONTAB | 1:27 | 1:27,1:27,1:49,1:63,1:63,1:63 |
| CONTLO | 1:12 | 2:139,3:446 |
| COPYRU | 3:209 | 3:199,3:201,3:204,3:209 |
| CORETU | - | 2:92,3:420 |
| CORIOB | 3:33 | 3:159,3:159,3:159 |
| COUBUS | 1:70 | 1:79,1:79 |
| COUCON | 1:70 | 1:27 |
| COUFIX | 1:70 | 1:27 |
| COUNQ3 | 3:382 | 3:382 |
| COUNT | - | 2:4,2:4 |
| COUNTD | 3:359 | 3:359 |
| COUNTQ | 3:382 | 3:382,3:382,3:382,3:382 |
| COUPLR | 1:17 | 1:78 |
| COUTAB | 1:70 | 1:70,1:79,1:79,1:79,1:91,1:93,1:95 |
| COUTBL | 1:70 | 1:70,1:79 |
| CPAGE | - | 2:35 |
| CPCORE | 1:96 | 1:96,1:107,1:112 |
| CPDH | 1:97 | 3:351,3:355,3:356 |
| CPDI | - | 3:351 |
| CPDL | 1:97 | 1:107,3:351,3:356 |
| CPFLGS | 3:46 | 3:369,3:369 |

CPKADD 1:97
CPKSIZ 1:97
CPKSSF 1:97
CPLINE 1:97 3:352
CPMASK 1:96 1:106
CPSATD 1:97 1:104,3:351,3:352,3:355
CPSATS 1:97 1:104,3:355
CPSIZE 1:97
CR 2:107 2:100,2:106,2:108
CRASH - 1:37
CREG 3:21
CRESET 3:21 3:90
CRFNM 3:21 3:291,3:291,3:359
CRIGHT 2:76
CRLFTX 2:108 2:107,3:81
CRPBIT 1:97 3:352
CRPCNT 1:97,3:351 3:352,3:354
CSETPU 1:96 1:96,1:101,1:106,1:112,3:355
CSHORT 3:21 3:256
CSLEE2 3:94 3:93,3:94,3:94
CSLEEP 3:94 3:93,3:94,3:95,3:96,3:96,3:97,3:97,3:98,3:99,3:377
CSLWS 3:21 3:251
CSRFLG 1:97
CTABLE 3:70 3:69,3:70
CTOHOT 3:117 3:119,3:119,3:119
CTYPE 1:97 1:97
CUMBIT 2:79
CUNPAC 3:56 3:56,3:57
CUP 2:76
CURADD 2:80 2:104,2:106,2:110,2:115
CURREN - 2:7,2:7
CWPCIN 1:12 1:11,1:63
CWRGFT 3:21 3:243
CYCLE - 3:27,3:118,3:118,3:415
D2FB 2:78 2:82,2:83,2:119,2:119,2:119,3:349
D2FGET 3:349 3:348,3:348,3:349
D2FL - 2:78,2:82,2:83,2:119,2:119,2:119,3:349,3:349,3:349,
3:417
D2FPOK 2:78 2:119,3:349
DATA 3:17 1:97,3:89,3:107,3:109,3:123,3:175,3:257,3:263,3:263,
3:263,3:274,3:278,3:279,3:279,3:279,3:283,3:285,3:287,
3:299,3:300,3:307,3:307,3:307,3:319,3:322,3:322
DAVRGE 3:178 3:177,3:178
DD - 2:76
DDACCU 2:80 2:98,2:104,2:104,2:104,2:105,2:105,2:107,3:66,3:66,
3:69,3:69,3:71,3:72
DDIENT 2:94 2:94
DDIVAR 2:80
DOPOKE 2:78 2:107,2:110,2:117,3:63
DDRESE 2:93 2:93,2:94
DDSTAC - 2:94
DDT - 2:34,2:34
DDTATT 1:33 1:15,1:33
DDTBFI 2:80 2:107,2:109
DDTBIT 3:46
DDTBUF 2:80 2:107,2:112,2:116
DDTCOD 2:34 2:82,2:91,3:7,3:7,3:46,3:63,3:63,3:64,3:64,3:65,3:65,
3:69,3:69,3:69,3:71,3:71,3:71,3:72,3:72,3:72,3:72,

3:72,3:74,3:74,3:74,3:81,3:81,3:81,3:81,3:228,3:305,
 3:315,3:317,3:317,3:320,3:320,3:382,3:393,3:397,3:445
DDTDIS 2:100 2:98,2:103,3:64,3:65,3:69,3:69,3:71,3:71,3:72,3:72,
 3:72,3:81,3:81
DDTGET 2:119 2:98,2:120,3:67
DDTINI 2:92 2:92
DDTIOB 3:33
DDTLED 3:62 3:65,3:65,3:347,3:348
DDTLOD 2:34 2:34
DDTLOK - 2:78,2:94,2:94,3:417
DDTMPL 3:347 3:347,3:347
DDTOUT 2:119 2:102,2:119,2:121,2:121,3:65,3:67,3:81,3:81
DDTPID 3:24 3:63,3:63,3:349,3:349,3:426
DDTPOL 2:94 2:81,3:63
DDTPRO 2:80 2:93,2:110,2:115
DDTSP 2:76 2:94,2:94
DDTSTE 2:96 2:96,2:120
DDTTAB 3:74 3:74
DDTVAR 2:34 2:76,2:80,3:7,3:76,3:445
DDTVST 2:34,2:34 2:34,2:34,2:34
DDTWAK 3:63 3:63
DDVARE 2:80 2:93
DDVARS 2:80 2:93
DEADSC 3:13 3:90,3:241,3:245,3:274,3:294,3:311,3:398
DEBUG 3:1 3:5
DEBUGM 1:24 1:45,1:48,2:60,2:70,2:82,2:107,2:119,2:126,3:346,3:349,
 3:349
DECIMA 2:104 2:101,2:104
DECNUM 2:80 2:104,2:104,2:105,2:105,2:105
DED41 3:315 3:315,3:315
DED44 3:315 3:315,3:315,3:315,3:315,3:315
DED46 3:315 3:315
DED47 3:315 3:315,3:315
DEDH 3:398 3:380,3:398
DEDREA 3:315 3:315
DEDSO 3:393 3:395
DEDS1 3:395 3:393
DEDS10 3:395 3:395,3:395
DEDS12 3:395 3:393
DEDS13 3:395 3:395
DEDS15 3:395 3:395
DEDS16 3:394 3:393,3:393,3:395
DEDS2 3:395 3:393,3:394,3:395,3:395,3:395,3:395
DEDS5 3:395 3:395
DEDS6 3:395 3:395
DEDS7 3:395 3:393,3:393,3:393
DEDS9 3:395 3:393,3:395
DEDTRN 3:393 3:395
DEDTYP 3:19 3:291,3:291,3:291,3:291,3:291,3:296,3:303
DEFATN - 1:10,1:10,1:15
DEFBEG - 1:10,1:13
DEFBUS 1:3 1:13
DEFENT 1:10 2:35
DEFILL - 1:10,1:10
DEFINT 1:8 1:8,1:8,1:8,1:8,1:13
DEFIO 1:7 1:13
DEFLIN 1:8 1:13,1:15
DEFLOO - 1:10,1:10,2:125

DEFMO 2:126 2:126,2:126
DEFM1 1:6 1:6,1:6,1:6,1:6
DEFMAC 1:10 1:10,1:10,1:10,1:10,1:10
DEFMEM 1:4 1:13,1:14
DEFPAG - 1:14,1:14,1:14,1:14,1:14,1:14,1:14,1:25,1:69,2:34,
2:34,2:34,3:7,3:7,3:7,3:7,3:8,3:8,3:8,3:8,3:8,3:31,
3:31
DEFPCN - 1:10,1:10,3:8
DEFREL 1:7 1:13,3:8
DEFRLD 1:9 1:13,1:15
DELAVE 3:12 3:138,3:139,3:177,3:179,3:179
DELBAS 3:12 3:138,3:139,3:179,3:179,3:219
DELHI 3:12 3:138,3:175,3:177
DELLOW 3:12 3:138,3:175,3:177
DELMAX 3:44 3:44,3:177,3:177
DELSHF 3:174 3:175
DEMAND 3:11 3:146
DEMPAS 3:19 3:71,3:146,3:146,3:159
DEMREL 3:18 3:146,3:146,3:159
DEPOSI 2:107 2:107
DEQUE 3:57 3:57,3:148,3:150,3:156,3:232,3:270,3:322,3:355,3:376
DERRPR 2:109 2:107,2:107,2:109
DESTH 3:16 3:16,3:244
DESTI 3:16 3:16,3:40,3:46,3:46,3:244,3:416
DESTIH 3:16
DESVPA 1:5 3:8,3:8
DEVINC 1:17 1:72,1:102,1:102,1:116
DEVINU 3:108 3:105,3:106,3:108
DEVTAB 3:104 3:104,3:104
DEVTYP 1:17 1:102
DGSB 3:360 3:85,3:361,3:376,3:376,3:402,3:402,3:415,3:416
DHALT 1:29 1:39,2:126
DHPASS - 1:37,1:39,2:126
DIAGRP 3:376 3:361,3:376
DIGIT 2:105 2:102,2:102,2:102,2:102,2:102,2:102,2:102,2:102,
2:102,2:105
DINIT 1:29 1:42,3:72,3:79,3:81
DISCAR 3:359 3:378
DISLED 3:359 3:359,3:359
DISMIS 2:22 2:22
DISPLO 2:98 2:95
DIVIDE 3:60 3:60,3:178
DLINF 3:30 3:44,3:44,3:44,3:44,3:44,3:44,3:44,3:44,3:138,3:138,
3:139,3:177,3:187,3:188,3:190,3:195,3:195,3:197,3:205,
3:219,3:219,3:226
DLOCK 3:12 3:105,3:138,3:138,3:139,3:139,3:175,3:175,3:177,3:177,
3:179,3:179
DLST 1:5 1:5,1:6
DMAP 2:80 2:93,2:110,2:115
DO1..6 2:136 2:136,3:46
DO25..6 2:136 2:136,3:119
DOAK 3:167 3:158,3:160,3:170
DOBUFFS 3:413 3:412,3:413
DOCLOC 2:136 2:136,2:136,2:136
DODEDL 3:132 3:131,3:134
DOINIT 2:133 2:133,3:423
DON - 2:6,2:7
DOROUT 3:203 3:202,3:203

| | | |
|---------|-----------|--|
| DOSPF | 3:187 | 3:186,3:187 |
| DOT | 2:104 | 1:11,1:11,2:101 |
| DOZE | 3:326 | 3:357 |
| DOZEPIC | 3:378 | 3:326,3:378 |
| DOZESP | 3:23 | 3:326,3:326 |
| DOZEW | 3:326 | 3:377 |
| DPCNT | 3:12 | 3:138,3:175,3:177 |
| DS | - | 3:76 |
| DSCPCT | 3:18 | 3:146,3:148,3:149,3:151,3:152,3:171,3:232 |
| DSFREQ | 3:174 | 3:176 |
| DSLEEP | 2:94 | 2:94,2:117,2:118,2:119,2:120,2:120 |
| DSPEOL | 3:80 | 3:79,3:79,3:80 |
| DSPFLA | 2:78 | 3:79,3:81,3:81,3:417 |
| DSPLOC | 3:78 | 3:78,3:79 |
| DSPLOK | - | 3:76,3:77,3:77,3:77,3:417 |
| DSPLYL | 3:79 | 3:77 |
| DSPOUT | 3:80 | 3:79,3:79,3:80 |
| DSPPID | 3:24 | 3:80,3:117,3:426 |
| DSPPOK | 2:78 | 2:89,3:80 |
| DSPPOL | 3:77 | 3:77 |
| DSPSLE | 3:77 | 3:77,3:79,3:80 |
| DSPSP | 3:76 | 3:77,3:77 |
| DSPSWI | 3:81 | 3:81,3:81 |
| DSPTXT | 3:81 | 3:81 |
| DSSTAC | - | 3:77 |
| DSSYSS | 3:76 | 3:77,3:77 |
| DSTAND | 1:50 | 1:13,1:33 |
| DSTH | 1:97,3:17 | 1:106,1:107,3:232,3:259,3:263,3:278,3:278, 3:351,3:355,3:356 |
| DSTHST | 3:19 | |
| DSTL | 3:16 | 3:64,3:65,3:90,3:269,3:324,3:343,3:351,3:356,3:363, 3:402,3:402,3:417,3:417,3:417,3:438,3:438,3:438,3:438, 3:438 |
| DSTOLI | 3:74 | 3:228,3:315,3:382,3:395,3:397 |
| DSYSSP | 2:76 | 2:94,2:94 |
| DTFLAG | 3:182 | 3:198,3:204 |
| DUMMY | 1:14 | 1:11,1:12,1:12,1:12,1:12,1:17,1:29,1:96,2:7,2:30,2:30,2:35, 2:76,2:78,3:9,3:27,3:173 |
| DXTCHK | 2:82 | 2:81,2:83,2:84 |
| DXTFLA | 2:78 | 2:82,2:83,3:63 |
| DYBLKE | 3:35 | 3:110 |
| DYINIT | 3:46 | 3:35,3:417 |
| DYNXT | 3:29 | 3:110,3:110,3:417 |
| DZ | - | 3:23 |
| DZSTAC | - | 3:326 |
| E000 | - | 1:13 |
| E1 | - | 1:10,1:10 |
| E100 | - | 1:13 |
| E2 | - | 1:10,1:10 |
| E200 | - | 1:13 |
| E3 | - | 1:10,1:10 |
| E4 | - | 1:10,1:10 |
| ECKQ | 3:26 | 3:85,3:85 |
| EDDTTA | 3:74 | 3:74 |
| EESBIT | 3:46 | |
| EFHCQ | 3:29 | 3:159,3:159 |
| EHPQ | 3:14 | 3:109,3:290,3:290,3:290,3:323 |
| EHQ | 3:14 | 3:109,3:270,3:270,3:290,3:290,3:290,3:290,3:323 |

EMIQ 3:12 3:105,3:155,3:155
EMSSTK 3:32 3:298,3:298,3:299,3:315,3:381,3:421
EMTPID 1:18,3:24
EMTY 2:128 3:426
END 1:23 2:7,2:107,2:111,3:53,3:169,3:390
ENDBIT 3:18 3:146,3:151,3:152,3:158,3:158
ENDI 1:96,3:23 1:100,1:100,1:104,3:59,3:154,3:241,3:246,3:256,
3:256,3:258,3:328,3:331,3:332,3:363
ENDINI 2:18
ENDO 1:96,3:23 1:103,3:153,3:153,3:270,3:322,3:323,3:335,
3:337,3:338
ENTER - 2:6,2:7
ENTLIS 1:10,1:10 1:11
EPRIQ 3:11 3:100,3:236,3:236,3:236
EQUAL - 1:34,1:52,1:55,1:60,1:60,1:61,1:74,1:75,1:79,1:79,
1:79,1:102,1:102,1:102,1:102,1:103,2:36,2:36,2:38,
2:45,2:56,2:64,2:65,2:82,2:83,2:87,2:92,3:80,3:345,
3:346,3:357,3:381,3:381,3:389,3:390,3:402,3:419
EREGQ 3:11 3:236,3:236
ERQ 3:26 3:119,3:119,3:235,3:235
ERRPRI 2:112 2:109,2:112,2:113
ERUPQ 3:181 3:212,3:212,3:216,3:218,3:229
ESENTQ 3:11 3:105,3:141,3:141,3:148,3:169
ETAB 2:113 2:113,2:113
ETQ 3:26 3:55,3:55,3:119,3:119
EXAMIN 2:110 2:111
EXIT 2:19,2:31 2:19,2:27,2:32
EXPBIT 3:46
F - 3:65,3:65,3:81
F.ILLE 2:13
F.LOCK 2:13
F.QUEU 2:13
F.RQUE 2:13
F.RUNN 2:13
F.WAIT 2:13
FOOO - 1:13
F1 - 1:50,3:389
F100 - 1:13
F2 - 1:95
F200 - 1:13
F2DB 2:78 2:82,2:82,2:120,2:120,3:349,3:349
F2DL - 2:78,2:82,2:82,2:120,2:120,2:120,3:349,3:349,3:349,
3:417
F2DPOK 2:78 2:120,3:349
F2DPUT 3:349 3:347,3:347,3:347,3:349
F2TPOK 2:78 2:89,3:345
F2TPUT 3:345 3:342,3:342,3:342,3:344,3:344,3:345,3:345
FADTEX 2:97 2:97
FAK - 3:7,3:7,3:8
FAKCOD 3:8 3:35,3:46,3:326,3:326,3:334,3:341,3:347,3:350,3:354,
3:355,3:359,3:361,3:363,3:381,3:382,3:392,3:395,3:397,
3:408,3:429,3:445
FAKCON 3:377 3:377,3:380
FAKEH3 3:35 3:89,3:270,3:270,3:270,3:270
FAKESI 3:23 3:59,3:328,3:328,3:330,3:331,3:363,3:363
FAKESO 3:23 3:323,3:335,3:335,3:335
FAKINI 3:419 3:423
FAKIPO 3:24 3:377,3:377

FAKLOD 3:6 3:7,3:7
FAKOPO 3:24 3:377,3:377
FAKTAB 3:380
FAKVAR 3:8 3:341,3:347,3:350,3:355,3:359,3:360,3:363,3:429,3:445
FAKVST 3:6 3:7,3:7,3:8,3:8,3:414,3:414
FALSE - 2:17,2:17,2:29,2:29
FASFAK 3:319 3:317,3:319
FASPOK 3:117 3:117,3:118
FBAD 3:279 3:273
FBLOCK 3:19 3:272,3:312
FCB - 2:13,2:13
FCBCHA 2:13
FCBCOD 2:13
FCBREL 2:13
FCBSP 2:13
FCBSTA 2:13
FCBSTK 2:13
FCBTIM 2:13
FCLIP 3:387 3:387,3:387
FDEAD 3:277 3:273
FDH 3:348 3:378
FDOZE 3:326 3:326,3:326,3:331,3:331,3:346,3:349,3:355,3:362
FDOZEW 3:326 3:326
FGET 3:52 3:53
FGETB 3:279 3:273
FGETR 3:287 3:273
FGVB 3:282 3:273
FH2JAM 3:355 3:378
FH2LED 3:350 3:350,3:351,3:351,3:351
FH2SUC 3:350 3:378
FHCJL 3:357 3:355,3:358
FHCJR 3:357 3:357
FHCQLD 3:355 3:356,3:356,3:356,3:356,3:356
FHCSBL 3:354 3:352,3:354
FHCSL 3:353 3:352,3:353
FHCSR 3:353 3:353
FHCSUC 3:351 3:350,3:352
FHD - 3:347 3:378
FHSIOI 3:59 3:41,3:41,3:41,3:59
FHT 3:341 3:378
FHTHLD 3:341 3:341,3:341,3:342,3:344
FHTPLD 3:344 3:341,3:342,3:344
FHTWRD 3:341 3:344,3:344,3:345
FIELD 2:32
FILLIN 3:12 3:105,3:131,3:154,3:154,3:154,3:155,3:155,3:155
FINCM 3:282 3:273
FINCQ 3:274 3:273
FINCR 3:277 3:273
FINDAC 3:169 3:169,3:169
FINDBA 3:208 3:198,3:204,3:206,3:208
FINDDT 2:81 2:81,2:81,3:63,3:63,3:77
FINDE1 2:131 2:131
FINDE2 2:131 2:131
FINDEV 2:131 3:114
FINDEX 2:131 2:131
FINDNA 3:208 3:192,3:198,3:199,3:199,3:203,3:204,3:205,3:206,3:208
FIOEND 3:23
FIXER 3:387 3:381,3:386,3:386

FIXJIF 2:66 2:50,2:51,2:65,2:66
FIXNAE 3:295 3:297
FIXNRE 3:295 3:286
FIXTAB 1:27 1:63,1:63,1:63,1:63
FJAM 3:328 3:328,3:328,3:330,3:343,3:343,3:348,3:356,3:376
FJAM1B 3:332 3:332,3:333
FJAMBU 3:333 3:333,3:333
FJAMEN 3:330 3:330,3:330,3:343,3:348,3:356,3:363,3:370,3:372,3:373,
3:373,3:376
FJAMLE 3:329 3:329,3:329,3:343,3:343,3:348,3:356,3:363,3:369,3:371,
3:373,3:373,3:376
FLAGOP 3:341 3:341,3:342
FLDIOR 3:181 3:184,3:213,3:213,3:215,3:215
FLIPPE 3:11 3:153,3:153
FLPOOL 3:392 3:390,3:392,3:422
FLUSH 3:50 3:50,3:87,3:141,3:170,3:218,3:229,3:241,3:255
FLUSHB 3:50 3:51,3:100,3:147,3:154,3:157,3:167,3:168,3:233,3:235,
3:279,3:322,3:333,3:352,3:356,3:357,3:376
FLUSHD 3:37 3:51,3:53,3:100,3:100,3:131,3:131,3:388,3:390,3:390,
3:391,3:392
FMESS 3:284 3:273
FNDCLK 1:64 1:35,1:53,1:64
FNDENT 3:194 3:189,3:189,3:197
FNDHAC 3:265 3:247,3:263
FOO - 2:78,2:85,2:89,2:89,2:93,2:93,2:104,2:105,2:105,2:105,
2:105,2:105,2:105,2:120,3:11,3:11,3:12,3:12,3:12,3:12,
3:13,3:14,3:14,3:20,3:23,3:23,3:33,3:33,3:33,3:33,
3:33,3:33,3:33,3:33,3:63,3:63,3:66,3:66,3:66,3:66,
3:72,3:72,3:79,3:81,3:89,3:89,3:90,3:90,3:90,3:90,
3:100,3:100,3:109,3:112,3:113,3:129,3:129,3:131,3:133,
3:133,3:133,3:134,3:135,3:137,3:138,3:138,3:138,3:138,
3:139,3:139,3:141,3:141,3:142,3:146,3:147,3:152,3:153,
3:162,3:169,3:177,3:177,3:177,3:237,3:241,3:241,3:245,
3:245,3:247,3:247,3:249,3:250,3:251,3:252,3:258,3:259,
3:271,3:281,3:281,3:285,3:287,3:301,3:309,3:311,3:313,
3:314,3:319,3:322,3:349,3:369,3:369,3:370,3:370,3:371,
3:371,3:371,3:371,3:372,3:372,3:375,3:401,3:402,3:404,
3:438
FORBAK 3:13,3:14 3:318,3:425
FORDIS 3:273 3:273
FORIMP 3:16 3:244,3:244
FORK 2:23
FORMER 2:114 2:113
FORUS 3:272 3:233
FORUS4 3:273 3:273,3:273,3:273
FOUTOF 3:277 3:273
FQOK 3:279 3:274,3:276,3:277,3:279,3:284,3:286,3:287,3:290,3:295,
3:295,3:296,3:296
FQOKGO 3:284 3:282,3:282,3:284
FQOKJ 3:286 3:285,3:288,3:288,3:288,3:288,3:289
FQOKJ2 3:287 3:287,3:287,3:287
FQOKJU 3:275 3:275,3:275,3:277
FQOKU 3:279 3:275,3:276,3:279,3:280
FREE - 3:27,3:51,3:51,3:51,3:53,3:53,3:53,3:390,3:391,3:392,
3:392,3:417
FREEND - 3:27,3:51,3:51,3:53,3:53,3:53,3:53,3:390,3:391,3:392,
3:392,3:417
FREGET 3:52 3:53,3:158,3:219,3:226,3:257,3:300,3:351,3:357

FREQ 3:283 3:273
 FRESET 3:287 3:273
 FRESRP 3:287 3:273
 FRESRQ 3:277 3:273
 FRET 3:255 3:255
 FRFAL 3:275 3:273
 FRFNM 3:277 3:273
 FRMIMP 3:16 3:269,3:294
 FRPCOD 3:291 3:296
 FRRQ1 3:277 3:277
 FSIOUT 3:323 3:41,3:41,3:41,3:323
 FSUCBU 3:338 3:338,3:338
 FSUCK 3:335 3:335,3:335,3:342,3:342,3:347,3:350,3:351,3:359
 FSUCKL 3:336 3:336,3:336,3:341,3:347,3:350,3:359
 FTH 3:343 3:378
 FTRAPV 1:15
 FTRG1 3:299 3:299
 FTRG2 3:299 3:299,3:299,3:299
 FTRNGT 3:299 3:275,3:277,3:277
 FUSE 3:19 3:19,3:19,3:265,3:272,3:272,3:278,3:279,3:312
 FVSTAR 2:87 2:85
 FWAIT 3:334 3:334,3:334,3:335,3:336,3:337,3:338,3:345,3:349,3:351,
 3:354
 FWAITW 3:334 3:334
 G - 1:107
 GABTYP 3:19 3:263,3:275
 GBRTYP 3:19 3:291
 GDRAW3 3:283 3:282
 GENM 3:363 3:361,3:363
 GENMCN 3:363 3:363
 GENMTI 3:363 3:363
 GET - 2:7
 GETFRE 3:300 3:263,3:307
 GETHAN 3:19 3:91,3:264,3:287
 GETMAX 3:19 3:242,3:242,3:244,3:296,3:394
 GETPAG - 3:125,3:125,3:125
 GETPRI 3:19 3:64,3:242,3:242,3:244,3:269,3:296,3:343,3:415,3:425,
 3:425
 GETUSE 3:19
 GLOBAL - 3:101,3:119,3:136,3:326,3:328,3:329,3:330,3:333,3:334,
 3:335,3:336,3:338,3:353,3:357,3:443
 GO - 2:7,2:7
 GOFH 3:378 3:377,3:378
 GOSJ6 2:128 2:126,2:126
 GOSTAT 3:424 3:380,3:422,3:424
 GOSTT 3:425 3:422,3:422
 GOTIHY 3:12 3:162,3:370
 GRRPKT 3:101 3:95,3:132
 GUDGV2 3:282 3:282
 GUDRAL 3:282 3:282,3:283
 GUDRAW 3:282 3:272
 GUDRQ2 3:283 3:283,3:284,3:284
 GUDRQ3 3:283 3:284
 GUDRQ5 3:284 3:283,3:299
 GUDRQ6 3:285 3:284,3:284
 GUDRQ8 3:284 3:283
 GUDTR8 3:285 3:284
 GVBTYP 3:19 3:314

GVHARE 3:443 3:316
GVTSKC 3:255 3:307
GVTSKM 3:255 3:263
H - 3:69
H2PBLK 3:25 3:69, 3:69, 3:69, 3:96, 3:96, 3:96, 3:96, 3:96, 3:99, 3:102, 3:106,
3:107, 3:123, 3:294, 3:311, 3:317, 3:329, 3:332, 3:338, 3:371,
3:375, 3:378, 3:383, 3:394, 3:395, 3:395, 3:396, 3:398, 3:399,
3:404, 3:407, 3:414, 3:414, 3:429
H2PEND 3:25
H2TX 3:249 3:250
HAC 3:18 3:278, 3:287, 3:293, 3:294
HACCOM 3:40 3:247, 3:263, 3:294
HACEND 3:40
HACMEM 3:40 3:40, 3:247, 3:265, 3:294, 3:316
HACSPC 3:40 3:40, 3:265, 3:294
HALTUS 1:50 1:33, 1:35
HBPID 3:23 3:118, 3:323, 3:337, 3:345, 3:349, 3:353, 3:377
HBPIDO 3:24 3:377, 3:377
HBUSY 3:23 3:59, 3:88, 3:253, 3:254, 3:323, 3:326, 3:331, 3:335, 3:336,
3:337, 3:338
HCTRL 3:14 3:371, 3:371
HDCBIT 2:78 3:316, 3:316
HDISP 3:25 3:78, 3:399
HDRST 3:131 3:132
HEAD 3:181 3:200, 3:204, 3:204, 3:206, 3:206, 3:207
HELST 3:11 3:144
HENDF 3:23 3:90, 3:90, 3:90, 3:241, 3:248, 3:250, 3:256, 3:260, 3:269,
3:322, 3:322
HENDP - 3:246
HERALD 2:95 2:94
HERRF 3:23 3:248, 3:250, 3:256
HEXLET 2:105 2:102, 2:102, 2:102, 2:102, 2:102, 2:105
HEXOUT 2:121 2:121, 2:122, 3:79, 3:79, 3:344
HFTABL 3:317 3:317, 3:317
HI 3:253 3:47, 3:107, 3:112
HI2 3:245 3:245, 3:245
HI2H 3:252 3:252, 3:252
HI2TSK 3:255 3:248, 3:251, 3:255
HI2TSV 3:255 3:247, 3:249, 3:250
HIBADC 3:252 3:245, 3:245, 3:245, 3:245
HIBF 3:13 3:59, 3:241, 3:241, 3:257, 3:258, 3:329, 3:332, 3:332
HIBITS 3:13, 3:15 3:237, 3:237, 3:242, 3:255, 3:255, 3:258, 3:260,
3:318
HIBLK 3:252 3:247, 3:261, 3:262
HICODE 3:16 3:90, 3:90, 3:91, 3:244, 3:244, 3:252, 3:256, 3:256, 3:356,
3:415, 3:415, 3:415, 3:415, 3:415, 3:415, 3:415, 3:415
HICPT1 3:248 3:248
HICUML 3:242
HICUMM 3:250
HICUMP 3:249
HICUMT 3:249
HIDE 3:245 3:245
HIDEAD 3:252 3:246, 3:265
HIDIE2 3:252 3:261
HIDIS3 3:241 3:241, 3:242
HIDIS4 3:241 3:242
HIDISC 3:241 3:245, 3:245, 3:247, 3:252
HIDISJ 3:252 3:251, 3:251, 3:252, 3:252

HIDOWN 3:241 3:245
 HIENDI 3:13 3:247,3:258,3:260
 HIERC 3:267 3:251,3:252
 HIERR 3:252 3:243,3:256,3:256
 HIGO 3:242 3:107
 HIGTGO 3:246
 HIGTX 3:248
 HIHAND 3:13 3:242,3:244,3:263,3:264
 HIHD 3:13 3:89,3:89,3:89,3:90,3:112,3:123,3:242,3:245,3:274,
 3:294,3:311,3:319,3:375,3:383,3:396,3:398,3:398,3:399,
 3:404,3:429
 HIHOST 3:13 3:90,3:107,3:246,3:261,3:263,3:264,3:265
 HIIDLE 3:242 3:112
 HIIFEI 3:252 3:245
 HIIMPD 3:252 3:246
 HILD12 3:256 3:242,3:243,3:244,3:246,3:256
 HILED2 3:244 3:242
 HILED4 3:245 3:242
 HILED5 3:243 3:244
 HILED8 3:244 3:244,3:244
 HILED9 3:244 3:244,3:244
 HILEDE 3:256 3:256
 HILEDI 3:256 3:242,3:244
 HILEDM 3:256 3:243,3:246
 HILO 3:13,3:15 3:107,3:112,3:253,3:253,3:300,3:425
 HILOOP 3:13 3:70,3:123
 HILS1 3:243 3:242
 HILS2 3:243 3:243
 HIMAXP 3:13 3:31,3:252,3:252
 HIMESS 3:250 3:250
 HIMG8M 3:249 3:248
 HINBUF 3:257 3:247,3:248,3:250,3:257
 HINBWT 3:258 3:241,3:247,3:248,3:250
 HINOP 3:252 3:243,3:245
 HINOP1 3:252 3:252
 HINOP2 3:252 3:252
 HINOP3 3:252 3:243
 HINOPT 3:252 3:243,3:244
 HINPID 3:13,3:15 3:102,3:115,3:237,3:253,3:275,3:287,3:318,
 3:425
 HIOldb 3:13 3:259,3:261,3:262,3:263,3:263,3:264,3:267
 HIPAD 3:31 3:246
 HIPBAD 3:251 3:260
 HIPBLJ 3:248 3:248
 HIPBLK 3:251 3:248,3:249,3:249,3:250,3:250
 HIPERR 3:251 3:248,3:250
 HIPK1 3:260 3:260
 HIPK1A 3:249 3:248
 HIPK1C 3:249 3:249,3:250
 HIPK2 3:260 3:260,3:260
 HIPK3 3:260 3:260
 HIPKER 3:260 3:260
 HIPKT 3:260 3:249,3:250
 HIPKTE 3:260 3:248,3:251
 HIPKTH 3:13 3:242,3:244,3:245,3:247,3:248,3:248,3:248,3:249,3:249,
 3:249,3:249,3:250,3:250,3:250,3:251,3:259
 HIPKTR 3:260 3:247
 HIPLC7 3:249 3:248

HIPLNG 3:251 3:250
HIPLTO 3:250 3:248,3:249
HIPOK 3:253 3:253,3:258,3:306,3:308,3:309,3:309
HIPSLO 3:251 3:248,3:250
HIPTIP 3:13
HIRDS 3:247 3:247,3:247,3:247
HIREG 3:246 3:245
HIREG3 3:247 3:246
HIREG4 3:246 3:246
HIREG5 3:246 3:246
HIREG6 3:248 3:247
HIREQ3 3:249 3:249,3:249
HIRFSH 3:249 3:247
HIRMDS 3:247 3:247,3:247,3:247,3:247
HIRSET 3:13 3:242,3:258,3:318
HISAV7 3:13,3:15 3:248,3:248,3:249,3:254,3:254,3:255,3:261,
3:265,3:307,3:307
HISET 3:259 3:247,3:248,3:251,3:259,3:260
HISP 3:13 3:88,3:241,3:247,3:248,3:249,3:249,3:250,3:258,3:259,
3:259
HISUB2 3:251 3:251
HISUB3 3:251 3:251
HISUBC 3:251 3:251,3:251,3:251
HITABL 3:41 3:41,3:59,3:256
HITC 3:318 3:317,3:318,3:320
HITRAN 3:13 3:237,3:242,3:242,3:248,3:248,3:251,3:252,3:253,3:257,
3:260,3:395
HITT 3:13,3:15 3:112,3:253,3:255,3:255,3:258,3:258,3:266,
3:267,3:300,3:302,3:309,3:318,3:318,3:318
HITTCO 3:258 3:242,3:247,3:258
HIURES 3:251 3:249,3:249,3:249,3:249,3:251
HIVHA 3:434 3:246,3:434
HIWAIT 3:242 3:241,3:249
HIWAL1 3:249 3:249
HIWFE 3:254 3:241,3:254,3:256
HIWM 3:253 3:242,3:242,3:246,3:249,3:249,3:253,3:254,3:255,3:257,
3:261,3:266,3:300,3:301,3:304,3:306,3:309,3:313
HIWORD 3:11 3:153,3:371,3:372
HLCODE - 3:7,3:7,3:45,3:88,3:241,3:264,3:267,3:271,3:287
HLOOPI 3:23 3:70
HMODID 1:96,3:9 1:102,3:111
HNEXIS 3:13 3:294
HNINIT 3:13 3:89,3:112,3:123,3:293,3:398,3:429
HOLDWD 3:373 3:361
HOLEN 3:14 3:35,3:35,3:35,3:35,3:35,3:107
HOLHN 3:13 3:107,3:107,3:263,3:264,3:265,3:281
HOME 3:76
HOMODE 3:13 3:89,3:90,3:91,3:107,3:242,3:246,3:252,3:269,3:294,
3:434,3:438,3:438
HOSCHK 3:102 3:47,3:47,3:102
HOSFAK 3:14 3:47,3:88,3:107,3:266
HOSREA 3:14 3:47,3:89,3:107,3:109,3:123,3:243
HOSTID 3:9 3:70,3:112,3:377
HOSTIP 3:14 3:47,3:88
HOSTNO 3:293 3:293
HOSTN1 3:294 3:293
HOSTN2 3:294 3:294
HOSTN3 3:294 3:294,3:294,3:294

HOSTN4 3:293 3:293
HOSTN5 3:293 3:293,3:294
HOSTN6 3:294 3:293,3:294
HOSTN7 3:294 3:294
HOSTN9 3:294 3:294
HOSTNA 3:294 3:294
HOSTNB 3:294 3:294
HOSTNC 3:294 3:294
HOSTND 3:294 3:293,3:294
HOSTNG 3:19 3:19,3:283,3:289,3:294,3:296
HOSTNM 3:293 3:273
HOSTNO 3:293 3:279,3:282
HOSTS 3:25 3:99,3:123,3:317,3:371,3:371,3:372,3:374,3:375,3:383,
3:395,3:395,3:396,3:398,3:404,3:415,3:429
HOSTUP 3:13 3:90,3:245,3:294,3:383,3:396,3:398,3:404
HOSTY8 3:14
HOSTYA 3:14
HOSTYP 3:13,3:15 3:59,3:89,3:89,3:96,3:96,3:107,3:109,3:123,
3:242,3:243,3:255,3:256,3:266,3:270,3:317,3:318,3:322,
3:383,3:425
HOSVDH 3:14 3:47,3:107,3:383
HOTABL 3:41 3:41,3:270,3:322
HOTEST 3:112 3:96,3:112
HOTHRU 3:268 3:249,3:268,3:324
HOTLIM 1:24 1:68,2:36,2:68,3:49
HOTPDL 3:13 3:87,3:102,3:115,3:290,3:317,3:324
HPOKE 3:123 3:118
HQUIT 3:23 3:88,3:251,3:256
HRDOFF 1:96,3:23 1:103,1:105,3:58,3:59,3:86,3:86,3:156,3:175,
3:221,3:260,3:328
HRDOWN 3:13 3:89,3:242,3:398
HREADY 3:23 3:89,3:109,3:123,3:398
HRESET 3:23 3:89,3:107,3:109,3:112,3:319
HRTABL 3:318 3:318,3:318
HSIOIN 3:59 3:241,3:257
HSIOUT 3:323 3:41,3:323
HSMVER 3:5 3:5,3:46
HSTL 3:16 3:64,3:65,3:90,3:91,3:269,3:269,3:343,3:351,3:356,
3:363,3:402,3:402,3:415,3:416,3:416,3:438,3:438,3:438,
3:438
HSTQUT 3:13 3:88
HTARDY 3:13 3:89
HTEMP 3:13,3:15 3:255,3:255,3:256,3:256
HTEMP7 3:13,3:15 3:255,3:255,3:256,3:256,3:257,3:257,3:258,
3:258,3:260,3:260,3:260,3:266,3:266,3:300,3:300
HTEST - 3:316
HTPMFL 3:14
HTPMFN 3:14 3:324
HTPMTL 3:14
HTPMTN 3:14 3:14,3:249,3:268,3:268,3:268,3:371
HTPPFL 3:14 3:268
HTPPFN 3:14 3:268
HTPPPTL 3:14
HTPPTN 3:14 3:268,3:268
HTPWFI 3:14 3:268
HTPWFI 3:14 3:268
HTSBIT 2:79 3:411,3:411
HUSEND 3:25 3:414

HVPIDO 3:24
HWSB 3:360 3:361,3:415,3:416,3:416,3:417,3:417
I - 1:23,3:72
I2M 3:141 3:47,3:105,3:111,3:143
I2MDEM 3:146 3:142,3:146
I2MDUN 3:152 3:151
I2MFLD 3:145 3:142,3:145
I2MHIH 3:144 3:142,3:144
I2MLOC 3:11 3:105,3:141,3:142,3:143,3:143
I2MLST 3:142 3:142,3:142
I2MNUL 3:151 3:143,3:151
I2MPKC 3:147 3:142,3:147
I2MPOK 3:11 3:55,3:142,3:142
I2MQXM 3:153 3:145,3:147
I2MRXM 3:148 3:142,3:149
I2MSFC 3:150 3:143,3:150
I2MSND 3:152 3:143,3:149,3:152
I2MSNL 3:151 3:151,3:151
I2MXMI 3:153 3:144,3:146,3:152,3:153
IBT1 3:425 3:425
IBT2 3:425 3:425
IBT3 3:425 3:422,3:427
IBUFL 1:20 1:20,1:40,2:63,2:63
ICVDH 3:47 3:105
ID - 1:23
IDDTOU 2:119 2:96,2:96,2:106,3:64,3:65,3:66,3:72
IDLEC 2:124 2:125,3:401
IDLEDO 3:401 3:401
IDLES 3:29 3:371,3:401
IDSPOU 3:80 3:79,3:79,3:79,3:80,3:80
IDSTAC - 1:26
IFAKEH 3:378 3:378,3:378
IGDFLA 3:429 3:396,3:396,3:429
IGDOWN 3:429 3:396,3:396,3:396,3:429
IH 3:88 3:47,3:107,3:112
IHO 3:90 3:324
IH1 3:90 3:90
IH4 3:90 3:90,3:90
IH5 3:91 3:90
IH6 3:91 3:90
IH7 3:91 3:91,3:91
IHBUFB 3:14 3:322,3:324
IHBUFF 3:14 3:89,3:322,3:322,3:338
IHCODE 3:16 3:269,3:396
IHCTL 3:90 3:90,3:90
IHCUML 3:269
IHCUMP 3:322
IHDB 3:88 3:90,3:90,3:91,3:322
IHDB2 3:88 3:88
IHDBA 3:88 3:89,3:89
IHDEO 3:89 3:89
IHDEAD 3:89 3:112
IHDMPB 3:270 3:89,3:270,3:270
IHDONE 3:324 3:322
IHDUMP 3:270 3:89,3:89,3:270
IHER2 3:323 3:322
IHER3 3:323 3:322
IHERDU 3:18 3:144

| | | |
|---------|------------|---|
| IHERR | 3:323 | 3:91 |
| IHERRC | 3:323 | 3:323 |
| IHEX | 3:324 | 3:324, 3:324 |
| IHEX2 | 3:324 | 3:323 |
| IHGGOIN | 3:14 | 3:88, 3:89, 3:109, 3:317 |
| IHGTGO | 3:91 | |
| IHIDLE | 3:90 | 3:109 |
| IHLEDR | 3:13 | 3:89, 3:90, 3:90, 3:90, 3:90, 3:90, 3:90, 3:90, 3:91, 3:91, 3:107, 3:109, 3:123, 3:269, 3:269, 3:269, 3:269, 3:269, 3:269, 3:269, 3:270, 3:270, 3:270, 3:270, 3:271, 3:271, 3:324, 3:324, 3:438, 3:438, 3:438, 3:438, 3:438, 3:438, 3:438, |
| IHLO | 3:14 | 3:88, 3:88, 3:109, 3:112 |
| IHLLOC | 3:14 | 3:35, 3:35, 3:35, 3:35, 3:88, 3:88, 3:109, 3:112, 3:112, 3:317, 3:317, 3:407, 3:407 |
| IHLS | 3:270 | 3:90, 3:90, 3:270 |
| IHLMSN | 3:269 | 3:90, 3:91, 3:270 |
| IHLSTP | 3:14 | 3:322, 3:322 |
| IHN1 | 3:90 | 3:90 |
| IHN2 | 3:90 | 3:90 |
| IHNEXT | 3:90 | 3:90 |
| IHOTLM | 1:68 | 2:36 |
| IHPACK | 3:322 | 3:91 |
| IHPKTH | 3:14 | 3:322, 3:324, 3:324, 3:324 |
| IHQ05 | 3:88 | 3:88 |
| IHRO | 3:89 | 3:88 |
| IHR1 | 3:89 | 3:89 |
| IHR2 | 3:324 | 3:324, 3:324 |
| IHR5 | 3:89 | 3:88 |
| IHR7 | 3:89 | 3:89 |
| IHRRAW | 3:91 | 3:91 |
| IHSSEQH | 3:14 | 3:322, 3:324 |
| IHSINI | 3:109 | 3:107, 3:109, 3:407 |
| IHTC | 3:317 | 3:317, 3:426 |
| IHTPID | 3:24 | 3:117, 3:426 |
| IHTT | 3:14 | 3:90, 3:90, 3:91, 3:112, 3:317, 3:317, 3:398 |
| IHVHA | 3:438 | 3:269, 3:438 |
| IHVHA1 | 3:438 | 3:438 |
| IHWQ | 3:14 | 3:88, 3:91, 3:91, 3:91, 3:109, 3:322, 3:323, 3:407 |
| IHYCUM | 3:144 | |
| IHYTIK | 3:12 | 3:131, 3:370 |
| II0BL3 | 3:378 | 3:378, 3:378 |
| ILIS | - | 1:3, 1:3, 1:3 |
| ILLBUF | 1:41, 1:41 | 1:41, 1:41, 1:44 |
| ILLC | 1:37 | 1:37 |
| ILLCNT | 1:41 | 1:40, 1:42, 2:63 |
| ILLCOP | 1:43 | 1:41, 1:43, 1:44 |
| ILLEND | 1:41, 1:41 | 1:41, 1:41 |
| ILLOPA | 1:24 | 1:11, 1:37 |
| ILLOPO | 1:24 | 1:41 |
| ILLPO | 1:37 | 1:53 |
| ILLP1 | 1:37 | 1:53 |
| ILOCKT | 1:73 | 1:73, 1:75 |
| ILOPTX | 2:97 | 2:97 |
| IMCFLD | 3:145 | |
| IMCNUL | 3:151 | |
| IMCUMP | 3:152 | |
| IMEMT | 2:37 | 2:37, 2:50 |

IMP.PK - 1:15
IMPDL2 3:396 3:396
IMPDL3 3:396 3:396
IMPDL4 3:396 3:396,3:396
IMPDLP 3:396 3:396,3:396
IMPDWN 3:396 3:380
IMPDX 3:396 3:396
IMPLNK 3:64
IMPOFF 3:360 3:362,3:424
IMPVER - 3:5,3:5
IN - 2:6,2:7
INBAS2 2:129 2:135,3:115,3:115
INBASE 2:129 3:377,3:377
INCH 3:17 3:157,3:232,3:236,3:255,3:260,3:276,3:278,3:355
INCM1 3:282 3:274,3:282,3:282,3:282
INCTRO 3:282 3:274,3:282
INCTR1 3:282 3:274
INCTR2 3:282 3:282
INDIRE - 2:68,2:68
INDVAR 1:6 1:11
INFREE 3:12 3:137,3:167,3:167,3:168,3:168,3:168,3:168,3:234,3:235
INHRST 3:11 3:95,3:132,3:132,3:132,3:133,3:134,3:139,3:147,3:158,
3:162,3:163,3:222
INI - 1:5,1:7,1:7,1:7,1:7
INIAALL 3:9 3:267
INICON 1:72 1:27,2:66
INIFIX 1:72 1:27,2:66
INIFLG 3:181 3:138,3:164,3:219
INIRAT 1:16 1:77
INITFO 2:18
INITLI 2:134 2:6,2:134
INITPI 2:135 2:6,2:135
INMTYP 3:19 3:251,3:270,3:324
INNER 1:10 1:10
INQTYT 3:19 3:295,3:305,3:305,3:305
INRTYP 3:19 3:291,3:303
INTABL 1:8 1:53
INTER 1:53 1:52
INTIME 1:28 1:51,2:37,2:49,2:50,2:50,2:50,2:50,2:52,2:52,2:52,
2:52,2:54,2:55,2:81,2:81,2:81,2:81,2:92,2:92,3:129,
3:129,3:129,3:418,3:419,3:423,3:423
INTLOO 3:70 3:70
INTPIT 3:46 3:384,3:384,3:385
INTPSH 3:46
INTPTI 3:46 3:384,3:384,3:385
INVMAP 3:437 3:437,3:438,3:439,3:444
IOBASE 1:69 1:11,1:69,1:102,2:56,2:131,3:98
IOBL3 3:33 3:378
IOBL4 3:33 3:378
IOBL5 3:33 3:378
IOBL6 3:33 3:378
IOBLEN 3:23 3:33,3:33,3:33,3:33
IOBLOC 3:11,3:13 3:70,3:88,3:105,3:107,3:108,3:108,3:111,3:112,
3:113,3:113,3:113,3:122,3:123,3:124,3:137,3:141,3:153,
3:154,3:241,3:249,3:253,3:257,3:260,3:270,3:317,3:398,
3:399,3:400,3:407
IOBTAB 1:7 1:11,1:69
IOCON 1:72 1:27,2:58

| | | |
|--------|-------|---|
| IOCTAB | 1:70 | 1:70, 1:79, 1:79 |
| IOCTBL | 1:70 | 1:79 |
| IOFIX | 1:72 | 1:27, 2:58, 2:58, 2:58 |
| IOKILL | 1:69 | 2:56, 3:72 |
| IOMASK | 1:17 | 1:82, 1:102, 2:56 |
| IRET | 1:19 | 1:37, 1:37, 1:37, 1:37 |
| IRRTIM | 3:43 | 3:43, 3:136 |
| IS | - | 2:6, 2:7 |
| ISTACK | - | 1:33, 1:35, 1:37 |
| ISTR | 3:64 | 3:64 |
| IT | 3:17 | 2:7, 3:55, 3:155, 3:175, 3:260 |
| ITABC | 3:415 | 3:416, 3:420, 3:420 |
| ITABCL | 3:416 | 3:420 |
| ITABI | 3:417 | 3:417, 3:423, 3:423 |
| ITABIL | 3:417 | 3:423 |
| ITABL | 3:417 | 3:417, 3:420 |
| ITABLE | 2:7 | 2:7, 3:423 |
| ITABLL | 3:417 | 3:420 |
| ITABQ | 3:417 | 3:417, 3:421 |
| ITABQL | 3:417 | 3:421 |
| ITABS | 3:414 | 3:414, 3:420, 3:420, 3:420 |
| ITABSL | 3:414 | 3:420, 3:420, 3:420, 3:420 |
| ITEXTO | 2:121 | 2:94, 2:96, 2:105, 2:107, 2:112, 2:113, 3:65, 3:70, 3:71, 3:72, 3:72, 3:81, 3:81 |
| IX | 1:19 | 1:33, 1:33, 1:34, 1:34, 1:35, 1:35, 1:36, 1:36, 1:37, 1:37, 1:37, 1:38, 1:40, 1:40 |
| JAM | 3:328 | |
| JAMBUF | 3:333 | |
| JAMCHK | 3:331 | 3:328 |
| JAMEND | 3:330 | |
| JAMHOL | 3:331 | 3:329, 3:332, 3:363 |
| JAMLEA | 3:329 | |
| JAMPKT | 3:331 | 3:328, 3:329, 3:330, 3:331, 3:332 |
| JAMPOK | 3:331 | 3:331 |
| JBTC | 3:320 | 3:320 |
| JDSPLY | 3:77 | 3:77, 3:426 |
| JED | - | 3:316 |
| JIHTC | 3:317 | 3:317 |
| JJDDT | 3:63 | 3:426 |
| JJTTY | 3:63 | 3:426 |
| JLOOP | 1:12 | 1:11, 1:45 |
| JPOLL | 1:24 | 1:45, 2:81 |
| JTIME | 1:22 | 1:35, 1:36, 2:62 |
| JUMP | - | 2:7, 2:7 |
| JUNK | 3:27 | 3:50, 3:105, 3:154, 3:155, 3:158, 3:241, 3:241, 3:332, 3:412 |
| K | 3:12 | 3:72, 3:132, 3:133, 3:134, 3:135, 3:136 |
| KB128 | 3:173 | |
| KB256 | 3:173 | |
| KB32 | 3:173 | |
| KB390 | 3:173 | |
| KB48 | 3:173 | |
| KB64 | 3:173 | 3:42, 3:42, 3:42, 3:42, 3:42, 3:42, 3:42 |
| KERLIM | 1:24 | 1:68 |
| KILLIN | 3:139 | 3:105, 3:134, 3:135, 3:139, 3:149, 3:158, 3:161, 3:163, 3:163, 3:169, 3:382, 3:430 |
| KNMISS | 3:135 | 3:133, 3:133, 3:135 |
| KNVALU | 3:43 | 3:43, 3:136 |
| KPOINT | 3:12 | 3:133, 3:135, 3:135 |

L - 1:23, 3:152
L\$1.6 1:24 1:11, 2:136
L\$25.6 1:24 1:11, 2:136
L%VARS - 3:35
L1 - 1:8, 1:8, 1:8, 1:8, 1:8, 1:8
L2 - 1:8, 1:8, 1:8, 1:8, 1:8, 1:8
L3 - 1:8, 1:8, 1:8, 1:8, 1:8, 1:8
L4 - 1:8, 1:8, 1:8, 1:8, 1:8, 1:8
LASTNU 2:80 2:121
LATER 3:11 3:100, 3:141, 3:153, 3:169
LBASE 2:124 3:97, 3:414
LBEG - 1:23
LBIG 3:1 3:5, 3:6, 3:7, 3:119, 3:141, 3:156, 3:261, 3:300, 3:305, 3:314,
3:315
LC - 1:23
LCCBAS - 3:97
LCILBU - 1:29, 1:29, 1:41, 1:43, 1:44, 1:44, 3:79, 3:373, 3:373,
3:373, 3:373
LCKTAB 1:68 1:91
LCLOCK 1:22 1:30, 1:30, 1:35, 1:36, 1:36, 1:45, 1:46, 1:46, 1:47, 1:64,
1:64, 1:116, 2:66, 2:128, 3:326, 3:331, 3:334, 3:337
LCLSTK 1:14 1:23
LCODE 1:14 1:11, 1:12, 1:24, 1:30, 1:99, 2:16, 2:28, 2:64, 2:81, 2:81,
2:90, 2:121, 2:125, 3:7, 3:8, 3:41, 3:43, 3:47, 3:47, 3:49,
3:50, 3:63, 3:63, 3:77, 3:77, 3:83, 3:86, 3:87, 3:93, 3:127,
3:139, 3:141, 3:142, 3:148, 3:151, 3:167, 3:175, 3:185, 3:212,
3:216, 3:217, 3:232, 3:300, 3:301, 3:314, 3:315, 3:317, 3:317,
3:320, 3:320, 3:326, 3:334, 3:391, 3:434, 3:435, 3:437, 3:438,
3:439, 3:445
LCOMAR 1:12 1:76, 1:77
LCOML 1:27 1:75
LCSTAC - 1:26
LDELT 3:12 3:133
LDEVTA - 3:104
LDRRUT 3:18
LDSPLO - 3:79
LEAVE - 3:52, 3:169, 3:390
LEDGO 3:271 3:271
LEDG1 3:271 3:271
LEDG3 3:271 3:271
LEDG4 3:271 3:271
LEDG5 3:271 3:271, 3:271
LEDG6 3:271 3:271
LEDG7 3:271 3:271
LEDG8 3:271 3:271
LEDGET 3:271 3:89, 3:90
LEDPO 3:87 3:252, 3:396
LEDP1 3:87 3:87
LEDP2 3:87 3:296, 3:394
LEDPC 3:296 3:275, 3:277
LEDPC1 3:296 3:296
LEDPC2 3:296 3:296, 3:296
LEDPF 3:87 3:395
LEFT - 2:7
LEN - 1:23, 1:23
LENL 3:16 3:91, 3:91, 3:324, 3:329, 3:336, 3:347
LEVEL1 1:33 1:13, 1:15
LEVEL2 1:34, 1:34 1:13

LF 2:106 2:100,2:106
 LFASPO - 3:118
 LFLAG 3:30 3:30,3:30,3:192,3:192,3:192,3:197,3:197,3:197,3:197,
 3:198,3:198,3:199,3:199,3:201,3:202,3:204,3:204,3:204,
 3:206,3:206,3:207,3:208
 LFLG 3:30 3:191,3:192,3:197,3:197,3:198,3:198,3:198,3:198,3:198,3:201,
 3:204,3:206,3:206,3:207,3:210,3:210,3:210,3:210,3:210
 LFLGB 3:30 3:192,3:199,3:204,3:207,3:208
 LFLGF 3:30 3:199,3:201,3:204
 LFLGU 3:30 3:192,3:192,3:197,3:198,3:198,3:202,3:206,3:206
 LGCUM 3:271
 LHACCS 3:40 3:265,3:294
 LI2MLS - 3:142
 LIKE - 2:32
 LIM - 1:5,1:6
 LIMEMT 2:37 2:50
 LIMIT - 1:13,1:14,1:14,1:14,1:14,1:14,1:14,1:14,1:25,1:69,2:34,
 2:34,2:34,2:34,3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:7,
 3:7,3:8,3:8,3:8,3:8,3:31,3:31
 LIMITAB 1:68 1:6,1:11,1:118,2:50
 LINEI1 3:137 3:132,3:137
 LINEI2 3:136 3:105
 LINEPR 3:42,3:42 3:42,3:42,3:136
 LINEUP 3:138 3:133,3:138,3:163
 LIS - 1:7,1:7
 LIS1 - 1:7,1:7
 LIST - 1:8,1:8
 LITES 3:399 3:380,3:401
 LK - 1:23
 LKERCK 1:24 1:12,1:52,1:73,1:77
 LKEREN 1:12 1:24
 LKERP 1:25
 LKPATC 1:25 3:445
 LKPLEN 1:16 1:25,1:25
 LKSTAC - 1:51
 LLOCKT 1:73 1:75
 LMAP 1:20 1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:6,1:11,1:51,1:88,
 1:89,1:89,2:38,2:39,2:41,2:64,2:133,2:134,3:93,3:127
 LMAPCT 3:29,3:414 3:29,3:414,3:418,3:423
 LMAPTB 1:6 1:11,1:20
 LMDCON 1:27 1:28,1:57,1:57
 LMIQ 3:12 3:105,3:155,3:155,3:156,3:156,3:156
 LMISS 3:12 3:131,3:135
 LMMBAS 1:69 2:43
 LNEI 3:11 3:162,3:162,3:162,3:209,3:219,3:370
 LNGALL 3:45 3:267
 LNGIMP 3:45 3:267
 LOCALC 1:24 1:12,2:36,2:36,2:68
 LOCBEG - 1:10,1:24
 LOCBLK 3:19
 LOCBLT 2:68 2:60,2:61,2:68
 LOCCON 1:72 1:27
 LOCCST 1:14 1:14,1:14,1:14
 LOCDEF - 1:28,1:28,1:28,1:28,1:70,1:70,1:70,1:71,1:71,1:72,
 1:72,1:72,1:72,1:98,2:78,2:78,2:78,2:78,3:25,3:26,
 3:27,3:27,3:27,3:27,3:27,3:27,3:27,3:28,3:28,3:28,
 3:28,3:29,3:31,3:31,3:32,3:33,3:33,3:33,3:33,3:33,
 3:33,3:33,3:33,3:34,3:34,3:34,3:35,3:35,3:35,

3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:76,
3:182,3:182,3:433
LOCEND 1:14 1:24,1:52,1:68,2:36
LOCFAD 1:18 1:37,2:96,2:97,2:120
LOCFIX 1:72 1:27,2:60,2:60
LOCFLG 3:93 3:99,3:99
LOCILL 1:20 1:40,1:40,2:63,2:63,2:63
LOCILQ 1:18 1:37,2:96,2:97,2:120
LOCIPT 1:20 1:40,1:40,2:63
LOCKDE 2:9
LOCKFD 3:23 3:33,3:33,3:33,3:33,3:85,3:85,3:159,3:159,3:326,3:326,
3:378
LOCKFW 3:23 3:33,3:33,3:33,3:33,3:334,3:334,3:378
LOCKHI 3:13 3:35,3:35,3:35,3:35,3:107,3:112,3:112,3:237,3:237,
3:253,3:253,3:300,3:318,3:318,3:425,3:427
LOCKIH 3:14 3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:35,3:87,3:87,
3:88,3:88,3:89,3:90,3:90,3:90,3:90,3:91,3:109,3:112,
3:112,3:270,3:270,3:290,3:290,3:322,3:323,3:324,3:395,
3:395
LOCKM 3:11 3:105,3:121,3:121,3:131,3:131,3:141,3:141,3:141,3:143,
3:143,3:143,3:144,3:144,3:144,3:148,3:148,3:148,3:148,
3:148,3:151,3:152,3:156,3:156,3:157,3:157,3:159,3:164,
3:164,3:167,3:168,3:168,3:169,3:170,3:234,3:234,3:235,
3:235,3:236,3:236,3:354,3:354,3:382,3:382,3:430,3:430
LOCKRO - 3:27,3:212,3:213,3:215,3:215,3:216,3:217,3:217,
3:218,3:218,3:224,3:225,3:225,3:229,3:229,3:417
LOCOPP 2:80 2:107,2:108,2:111,3:66,3:66
LOCQUT 1:18 1:32,2:96,2:97,2:120
LOCSSST 1:14 1:14,1:14
LOCSTS 1:27 1:48
LOCTVS 3:6 3:8
LOCVST 1:14 1:14,3:8
LOCZEL 1:22 1:51
LOCZER 1:22 1:22,1:51
LOD - 1:5,1:6,1:6,1:12
LODINT 1:8 1:19
LOOP1 2:128 2:126,2:127
LOOP2 2:128 2:128
LOOP3 2:128 2:128,3:46,3:410
LOOP4 2:128
LOOP5 2:126 2:126
LOOPE 2:127 2:126,2:126,2:126,2:127,2:127,2:127
LOOPM 2:126 2:128,3:84,3:253
LOOPMV 2:126 2:19,2:20,2:31,2:32,2:125,3:63,3:63,3:94,3:128,3:164,
3:185,3:326,3:334
LOVRTA - 3:65
LOWMSK 1:96 1:103,1:103,1:103,1:103
LOWORD 3:11 3:153,3:153,3:371,3:372
LPCKSU 3:86 3:86,3:263,3:278,3:351
LPSBTA - 2:85
LRBLK 3:20 3:32,3:298,3:298,3:299,3:315,3:381,3:381,3:421
LREASF 3:315,3:315 3:315
LST 3:30 1:5,1:5,3:30,3:200,3:201,3:201,3:201,3:201,3:202,3:202,
3:204,3:204,3:206,3:206,3:207,3:207,3:207,3:207,3:207,
3:207,3:207,3:207,3:210
LSTACK - 1:45,2:62,2:126,2:126,2:128,3:50,3:50,3:54,3:56,3:89,
3:239,3:241,3:241,3:246,3:247,3:248,3:249,3:251,3:255,
3:256,3:257,3:260,3:260,3:263,3:269,3:269,3:270,3:277,

3:278,3:279,3:290,3:293,3:293,3:300,3:324,3:382,3:394,
 3:395
 LSTKLN 1:14 1:14,1:23,2:62,2:62,2:126,2:126,2:126,2:126
 LSTLIM - 3:445,3:445,3:445
 LSTLST 3:30 3:201,3:201,3:201,3:202,3:204,3:206,3:206,3:207,3:207,
 3:207,3:207,3:207,3:207
 LSTOL 3:30 3:204,3:207,3:207
 LSTOST 3:30 3:200,3:201,3:202
 LSTPKT 3:19 3:249,3:260,3:286,3:322
 LSYSFC - 2:15
 LTB 3:30 3:30,3:187,3:188,3:189,3:189,3:192,3:194,3:195,3:195,
 3:195,3:195,3:195,3:195,3:197,3:197,3:197,3:197,3:198,3:198,
 3:201,3:201,3:202,3:204,3:205,3:205,3:206,3:208,3:211,
 3:226,3:226
 LTBDST 3:30 3:187,3:189,3:189,3:195,3:195,3:201,3:205,3:226
 LTBNAY 3:30 3:188,3:192,3:194,3:195,3:195,3:195,3:198,3:198,3:201,
 3:202,3:204,3:205,3:206,3:208,3:226,3:226
 LTBSUM 3:182 3:189,3:195,3:195,3:197,3:210,3:211
 LTIME 1:22 1:36,1:46,1:64,2:66,2:128
 LTQ - 3:26,3:55,3:55,3:119,3:119,3:232,3:232,3:232,3:235,
 3:235,3:417
 LTRNPU 3:314,3:314 3:314
 LTRTN 3:314 3:314
 LTSTS - 3:369
 LTVARS 3:8 3:93,3:445
 LUSE 3:32 3:263,3:263,3:280,3:281,3:298,3:312,3:315,3:324,3:393
 LUSEO 3:32 3:32,3:263,3:281
 LVARS 1:14 1:11,1:19,3:231,3:326,3:445
 LVHALI - 3:441
 LVHINV - 3:437,3:440,3:440,3:441,3:442,3:442,3:442,3:442,3:443
 M - 3:69
 M% - 2:7
 M%LOCA 2:6
 MO - 1:11,1:14,1:14,1:28,1:38,1:39,1:45,1:49,1:49,1:57,
 1:57,1:57,1:91,1:91,1:92,2:64,2:65,2:65,2:84,2:94,
 2:131,2:133,2:133,3:49,3:74,3:84,3:94,3:128,3:141,
 3:156,3:185,3:185,3:185,3:300,3:314,3:314,3:315,3:315,
 3:317,3:317,3:320,3:320,3:435,3:435,3:437,3:437
 M1 - 1:6,1:14,1:29,1:34,1:39,1:40,1:47,1:57,1:58,1:58,1:60,
 1:60,1:60,1:67,1:68,1:68,1:68,1:88,1:88,2:9,2:34,2:37,
 2:37,2:41,2:44,2:44,2:45,2:45,2:45,2:49,2:50,2:50,
 2:50,2:50,2:50,2:51,2:51,2:51,2:52,2:52,2:53,2:54,
 2:55,2:84,2:84,2:94,2:94,2:126,2:133,3:7,3:7,3:7,3:7,
 3:8,3:8,3:49,3:49,3:77,3:84,3:94,3:94,3:94,3:113,3:128,
 3:128,3:153,3:153,3:185,3:185,3:185,3:185,3:226,
 3:226,3:226,3:226,3:226,3:237,3:237,3:237,3:317,
 3:319,3:320,3:326,3:328,3:329,3:329,3:330,3:331,3:331,
 3:331,3:331,3:334,3:335,3:336,3:336,3:336,3:337,3:337,
 3:337,3:414,3:414,3:422
 M2 - 1:14,1:47,1:47,1:47,1:48,1:48,1:60,1:60,1:67,1:74,
 1:80,1:81,1:81,1:81,1:81,1:91,1:114,1:118,2:34,2:45,
 2:49,2:52,2:109,2:115,2:116,2:117,2:117,2:117,2:117,
 2:117,2:118,2:118,2:118,2:126,2:129,2:129,2:133,3:35,
 3:35,3:49,3:49,3:50,3:52,3:52,3:54,3:55,3:56,3:56,
 3:56,3:56,3:56,3:56,3:57,3:57,3:59,3:65,3:65,3:65,
 3:84,3:85,3:93,3:94,3:117,3:117,3:117,3:117,3:117,
 3:117,3:119,3:120,3:121,3:122,3:127,3:128,3:148,3:153,
 3:153,3:153,3:155,3:156,3:158,3:161,3:171,3:185,3:185

3:187,3:187,3:187,3:187,3:187,3:188,3:188,3:188,
3:189,3:189,3:212,3:213,3:215,3:215,3:217,3:217,3:221,
3:224,3:225,3:225,3:229,3:234,3:235,3:235,3:235,3:236,
3:237,3:238,3:238,3:238,3:238,3:239,3:242,3:243,3:244,
3:246,3:247,3:257,3:259,3:259,3:259,3:259,3:260,3:270,
3:322,3:322,3:326,3:328,3:329,3:330,3:332,3:332,3:333,
3:334,3:335,3:336,3:338,3:338,3:338,3:351,3:352,3:353,
3:354,3:357,3:369,3:369,3:369,3:369,3:369,3:369,3:389,
3:410,3:414,3:418,3:419,3:419,3:419,3:419,3:423,3:423,
3:423,3:439
M2I 3:154 3:47,3:105,3:111,3:157
M2IDSP 3:157 3:157,3:157
M2IHIIH 3:161 3:160,3:163
M2IHW 3:154
M2ILOC 3:12 3:105,3:154,3:154,3:154,3:155
M2INUL 3:160 3:160,3:160
M2INXT 3:158 3:154,3:155,3:158
M2IRCK 3:166 3:164,3:166
M2IREG 3:158 3:157,3:157,3:158
M2IRLD 3:159 3:157,3:159
M2IRUP 3:164 3:160,3:165
M2IRUT 3:160 3:157,3:157,3:160
M2NGHB 3:25 3:161,3:352,3:355
M2PBLK 3:25 3:69,3:95,3:95,3:99,3:103,3:105,3:105,3:118,3:121,
3:122,3:162,3:177,3:178,3:179,3:179,3:209,3:214,3:219,
3:222,3:233,3:352,3:370,3:371,3:371,3:382,3:383,3:400,
3:414,3:430
M2PEND 3:25 3:69
M3 - 1:49,1:49,1:51,1:90,1:91,1:91,1:101,1:101,1:101,1:106,1:106,
1:106,1:107,1:110,1:110,1:111,1:115,1:115,1:116,2:9,
2:34,2:50,2:84,2:94,2:107,2:109,2:109,2:109,2:109,
2:110,2:112,2:112,2:115,2:115,2:115,2:115,2:115,
2:115,2:115,2:116,2:116,2:117,2:117,2:117,2:117,2:117,
2:117,2:118,2:118,2:119,2:120,2:133,3:37,3:63,3:84,
3:94,3:94,3:113,3:128,3:128,3:161,3:184,3:184,3:185,
3:185,3:190,3:259,3:259,3:259,3:319,3:328,3:329,3:329,
3:329,3:330,3:331,3:331,3:331,3:331,3:334,3:335,3:336,
3:337,3:337,3:337,3:353,3:353,3:353,3:355,3:357,3:357,
3:357,3:414,3:419
MAGMOD 1:96,3:23 1:102,3:111
MAP - 1:6,1:14,1:69,2:6,2:16,2:28,2:34,3:7,3:7,3:7,3:8
MAPB1 - 3:414
MAPB2 - 3:414
MAPB3 - 3:414
MAPB4 - 3:414
MAPB5 - 3:414
MAPB6 - 3:414
MAPBIT 2:80 2:110,2:110,2:115
MAPCOD - 2:126,2:126,3:7,3:7,3:314,3:315,3:317,3:320
MAPCOM 1:20 1:47,1:47,1:49,1:51,1:55,1:55,1:59,1:59,3:49
MAPCT 3:414 3:414,3:418,3:423
MAPDDT - 2:81,2:116,2:126,3:7,3:7,3:300,3:314,3:315,3:317,3:320
MAPFAK - 3:65,3:85,3:326,3:334,3:423
MAPMSK 1:17 1:58,2:127,3:336,3:338,3:388
MAPPKG 3:7 3:7,3:49,3:141,3:156
MAPREL - 1:45,1:48,1:49,1:49,1:75,1:103,1:105,2:45,2:64,2:65,
2:65,2:117,2:117,2:118,2:118,2:131,3:84,3:117,3:353,
3:357,3:369,3:410,3:419,3:422

MAPRUT 3:7,3:7 3:185
 MAPV2 - 2:126,2:127,2:128,2:129,2:133,3:50,3:50,3:52,3:54,
 3:54,3:55,3:56,3:57,3:65,3:84,3:85,3:91,3:93,3:117,
 3:119,3:121,3:127,3:148,3:153,3:155,3:158,3:171,3:187,
 3:187,3:189,3:213,3:215,3:217,3:221,3:225,3:225,3:226,
 3:235,3:237,3:238,3:247,3:256,3:257,3:259,3:260,3:260,
 3:270,3:278,3:279,3:285,3:288,3:290,3:290,3:297,3:299,
 3:322,3:332,3:338,3:338,3:352,3:369,3:388,3:401,3:410,
 3:412,3:414,3:420
 MAPVAR - 1:34,1:39,1:49,2:63,2:81,2:109,2:112,2:117,2:118,2:126,
 2:126,2:133,3:84,3:153,3:226,3:237,3:259,3:270,3:329,
 3:336,3:336,3:338,3:353,3:353,3:357,3:357,3:414,3:420,
 3:422
 MAPVDH 3:7
 MAPVHA 3:7 3:369,3:435,3:437
 MASTER 3:11 3:11,3:11,3:132,3:132,3:132,3:133,3:139,3:152,3:158,
 3:163,3:163,3:163,3:163
 MAXBUF 3:27 3:385,3:390,3:391,3:412,3:422
 MAXCHN 3:11 3:101,3:136,3:150,3:152,3:152,3:167,3:167,3:382
 MAXLED 3:21 3:91,3:242,3:244
 MAXNF 3:27 3:375,3:375,3:384,3:386,3:412
 MAXSTR 1:7 1:11,1:24
 MAXVHA 3:433 3:433
 MBLKS 2:124 2:129,2:130,3:88,3:97,3:141,3:154,3:253,3:300,3:326,
 3:334,3:425
 MBUSY 1:96,3:23 1:103,1:104,3:141,3:143,3:154
 MD - 1:23
 MDISP 3:25 3:78,3:400
 MDSTAC - 1:26
 MEDMIN 3:9 3:132
 MEMCON 1:72 1:27,2:60
 MEMDIS 1:77 1:75,1:77
 MEMFIX 1:72 1:27
 MEMKIL 1:25 1:47,1:54,1:77
 MEMRAT 1:16 1:54
 MEMSEG 1:28 1:55,1:55,1:67,1:74,3:49
 MEMTOT 1:28 1:55,2:38
 MEMTST 1:67 1:60,1:67,2:38,2:47,2:53
 MENDF 1:96,3:23 1:103,3:153,3:153,3:154
 MERRF 3:23
 MESEEN 3:19 3:19,3:274,3:291,3:295,3:296
 MESGET 3:261 3:248,3:249
 MESNO 3:19 3:19,3:302
 MESNUM 3:19 3:299,3:302,3:314
 MESNXT 3:19 3:273,3:273,3:274,3:283
 MESOUT 3:19 3:19,3:273,3:274,3:291,3:291,3:295
 MESSO 3:32 3:32,3:265,3:288,3:314
 MESSID 1:97,3:19 3:259,3:402
 MESSNO 3:32 3:263,3:265
 MESST - 3:27,3:406,3:406
 MESSTK 3:32 3:298,3:298,3:299,3:315,3:381,3:381,3:381,3:381,3:381,
 3:381,3:381,3:421
 MESSTO 3:406 3:380
 MESSTS 3:406 3:406,3:406,3:406
 MESTST 3:291 3:273
 MESTYP 3:21 3:243
 MG00 3:261 3:261,3:263
 MG01 3:261

MGO2 3:261 3:261
MGO3 3:261 3:261
MGO4 3:261 3:261
MGO5 3:261 3:261
MGO6 3:261 3:261
MGO7 3:263 3:261,3:261
MGO8 3:261 3:261
MGO9 3:261 3:264
MG10 3:264 3:264
MG11 3:264 3:264,3:265,3:265
MG12 3:265 3:264
MG13 3:265 3:265
MG15 3:262 3:263
MG21 3:263 3:264
MG22 3:265 3:264
MG23 3:263 3:263
MGNL 3:360 3:363,3:416
MGSB 3:360 3:361,3:363,3:363,3:402,3:402,3:415,3:416,3:416,3:417
MGTEST 3:264 3:261,3:261
MHPCCHK 3:102 3:102,3:102,3:103
MICUME 3:158
MICUMI 3:157
MIDH 3:17 1:97,3:91,3:247,3:259,3:263,3:263,3:272,3:273,3:275,
3:278,3:279,3:280
MIDL 3:16 3:90,3:90,3:90,3:91,3:269,3:271,3:271,3:351,3:356,
3:402,3:402,3:402,3:416
MINE 3:25 3:64,3:65,3:65,3:90,3:132,3:138,3:144,3:151,3:162,
3:163,3:164,3:183,3:210,3:219,3:220,3:246,3:259,3:263,
3:265,3:267,3:294,3:307,3:343,3:351,3:410,3:417,3:417,
3:417,3:417,3:422,3:423,3:424
MINF 3:27 3:52,3:238,3:304,3:375,3:384
MINPID 3:11,3:13,3:23 3:102,3:108,3:115,3:124
MINPRO 2:66 1:11,1:11,2:66
MINTIM 3:28 3:403,3:403
MISW 3:1 3:29,3:93,3:95,3:99,3:105,3:107,3:113,3:114
MITHRU 3:12 3:234
MLIMX 1:6 1:6
MLIS - 1:3,1:3,1:3
MLNGTH 3:363 3:363
MLOOP 3:11 3:70,3:124,3:124,3:137,3:154,3:400
MLOOPE 3:23 3:70,3:124
MLOOPI 3:23 3:70,3:124
MLTPKT 3:19 3:248,3:248,3:250,3:283,3:284,3:302,3:303,3:305,3:314
MM - 1:23
MMAXST 1:24 1:11,1:36
MMLIMS 1:6 1:11,1:68
MMSTAC - 1:26
MNBUF2 3:37 3:37,3:37,3:37,3:37,3:388,3:412,3:412,3:412,3:412,
3:412
MNCODE 1:6 1:11,1:11
MNOBUF 3:27 3:158
MNOVAR 1:6 1:11
MNVARs 1:6 1:6,1:11
MODCHK 3:103 3:47,3:47,3:103
MODEM 3:11 3:100,3:105,3:111,3:136,3:138,3:141,3:144,3:145,3:161,
3:164,3:224,3:355
MODEMS 3:25 3:99,3:118,3:121,3:214,3:219,3:222,3:370,3:371,3:371,
3:371,3:371,3:372,3:374,3:382,3:383,3:400,3:430

MODID 1:96,3:9 1:102,1:102,3:111,3:111
 MODLEN 3:12 3:35,3:105
 MODLOO 3:70 3:70
 MOTPID 3:11,3:13,3:23 3:55,3:102,3:108,3:115,3:168,3:236
 MP - 2:4,2:4,2:4,2:4,2:4
 MPOKE 3:124 3:118,3:124
 MQCNT 3:382 3:382
 MQCNT1 3:382 3:382
 MQCNT2 3:382 3:382
 MQCNT3 3:382 3:382
 MQCNT4 3:382 3:382
 MQCNT5 3:382 3:382
 MQCNT6 3:382 3:382
 MQCNT7 3:382 3:382
 MQCNT8 3:382 3:382,3:382
 MQUIT 3:23 3:141,3:154
 MRESET 3:23 3:137,3:141,3:154
 MRTIME 3:11 3:121,3:136,3:148
 MSGBIT 2:78 3:361
 MSPARE 3:12
 MSTO 3:32 3:32,3:32,3:265,3:306,3:306,3:406
 MSTOO 3:32 3:306,3:406
 MSTR 3:65 3:65,3:65
 MSTRIP 1:7 1:13
 MTEST 3:111 3:95,3:111
 MTEST1 3:264 3:264,3:264,3:264
 MTEST2 3:264 3:264
 MYIMP 3:25 3:242,3:246,3:293,3:398,3:398,3:415,3:429
 MYNET 3:9 3:416,3:416
 MYPROC 1:19 1:79,1:80,1:89,1:91
 MYSEGS 1:22 1:54,1:55,2:41
 N 3:12 2:27,2:27,2:27,3:65,3:71,3:72,3:81,3:133,3:135,3:136
 N.HALT 3:28 3:71
 N.RELO 3:28 3:71,3:159
 N.REST 3:28 3:71
 N10XH 3:9 3:46
 N10XI 3:9 3:46
 NAL 3:27 3:286,3:295,3:295,3:304,3:304,3:375,3:381,3:381,3:385
 NAL2 3:295 3:295,3:295
 NAME 2:9,2:16,2:20,2:30,2:32 1:5,1:5,1:5,1:5,1:5,1:5,1:5,1:5,1:5,
 1:5,1:5,1:5,1:5,1:5,1:5,1:5,1:5,1:5,1:12,1:12,1:23,1:23,
 1:23,1:23,2:6,2:6,2:6,2:6,2:6,2:6,2:9,2:9,2:15,2:15,
 2:15,2:16,2:16,2:16,2:16,2:19,2:19,2:20,2:22,2:22,
 2:23,2:23,2:23,2:23,2:23,2:27,2:27,2:27,2:27,2:27,
 2:27,2:28,2:28,2:28,2:30,2:30,2:31,2:32
 NBAK 3:27 3:278,3:300,3:383,3:384
 NCC 3:9 3:40,3:40,3:46,3:46,3:46,3:46,3:416,3:416
 NCCHST 3:46 3:402
 NCCIMP 3:46 3:65,3:402,3:402
 NCCLNK 3:46 3:402
 NCMOVE 3:238 3:164,3:233,3:239
 NCODEM 1:69 1:11
 NCODEP 1:69 1:11,1:88,1:88,1:89,1:89,2:38,2:41,2:41,2:44,
 2:45,2:45,2:47,2:47,2:65
 NCPLOD 3:6
 NDVARS 1:69 1:11,2:41
 NEIGH 3:11 3:132,3:138,3:139,3:161,3:161,3:161
 NEQUAL - 2:52

NETH 1:97,3:17 3:91,3:124,3:124,3:144,3:144,3:146,3:148,3:151,
3:152,3:152,3:157,3:160,3:161,3:164,3:164,3:166,3:167,
3:167,3:167,3:171,3:182,3:187,3:212,3:221,3:232,3:259,
3:290,3:351
NETL 3:16 3:90,3:91,3:356,3:415,3:415,3:415,3:415,3:415,
3:415,3:415
NEW - 2:7,2:7
NEWARG 3:66 3:66,3:72
NEWCOM 1:22 1:54,1:59,1:59
NEWCTS 3:27 3:383,3:384,3:385,3:385,3:385,3:385,3:385,3:386
NEWLOO 3:167 3:167
NEWMES 3:285 3:285
NEWPKC 1:1 1:109,1:112,1:117
NF - 3:27,3:50,3:51,3:51,3:52,3:52,3:53,3:53,3:54,3:54,
3:238,3:238,3:239,3:286,3:286,3:286,3:286,3:295,3:295,
3:295,3:304,3:304,3:304,3:375,3:381,3:381,3:386,
3:386,3:386,3:386,3:392,3:392,3:417
NFH 1:96,3:9 1:101,3:9,3:9,3:69,3:96,3:96,3:96,3:96,
3:99,3:99,3:106,3:107,3:112,3:118,3:244,3:265,3:269,
3:269,3:293,3:293,3:294,3:294,3:294,3:328,3:374,3:377,
3:434,3:438,3:439
NFH2 3:9 3:69,3:123,3:123,3:326,3:334,3:371,3:371,3:371,
3:372,3:374,3:375,3:377,3:383,3:383,3:398,3:398,3:399,
3:408,3:415
NHI 3:27 3:257,3:357,3:385
NHOBLK 3:35
NIBUF 1:20 1:20,1:40,2:63
NICEST 3:429 3:415
NILEND 3:373,3:373 3:373
NILOPS 3:373,3:373 3:373,3:373,3:373,3:373,3:402
NIMP 3:9 3:12,3:30,3:30,3:30,3:30,3:30,3:134,3:182,3:189,3:189,
3:191,3:200,3:210,3:210,3:211,3:211,3:211,3:223,3:225,
3:228,3:228,3:246,3:374,3:404,3:410,3:433
NLINE 3:9 3:30,3:30,3:30,3:195
NLLOCST 1:26 1:63,1:63
NMD 3:9 3:25,3:25,3:95,3:99,3:103,3:105,3:162,3:176,3:179,
3:179,3:181,3:209,3:400
NMDBLK 3:35
NMDLIM 3:1,3:5 3:5,3:12,3:12,3:12,3:43,3:43,3:43,3:43,3:137,
3:382
NMI 3:27 3:158,3:385
NMSEG 1:4 1:16,1:22,1:25,1:28,1:47,1:47,1:48,1:54,1:55,1:60,
1:61,1:61,1:67,2:39,2:40,2:43,3:49,3:49
NOBLK 3:18 3:278,3:287,3:293,3:294
NOIMPO 3:9 3:192,3:192,3:192,3:192,3:200,3:210,3:210,3:210,
3:210,3:211,3:224,3:227,3:228,3:228
NONOPS 3:88 3:89,3:242
NOPID 1:18,3:24 1:24,3:426
NOPIDS 2:125 2:125,3:426
NOPTVM 1:69 1:11
NOSAVE - 1:32,1:33,1:36,1:44,1:44,1:51,2:81,2:125,2:125,3:114,
3:239,3:253,3:253,3:254,3:257,3:269,3:270,3:299,3:319,
3:319,3:319
NOT - 3:94
NOTEKT 2:114 2:96,2:97,2:113,2:113
NOTLA 3:286 3:286,3:286
NOTLB 3:286 3:286
NOTME 3:65

NOTRAP - 1:57, 1:57, 1:58, 1:59, 1:59, 1:59, 3:420
 NOTRAW 3:272 3:272
 NOVARP 1:69 1:11, 2:39
 NOW - 2:7
 NPROC 1:16 1:70
 NQUIT - 1:40, 1:49, 1:56, 1:58, 1:58, 1:59, 2:45
 NRBLK 3:32 3:32
 NRE 3:27 3:282, 3:283, 3:284, 3:286, 3:286, 3:295, 3:295, 3:304, 3:304,
 3:304, 3:375, 3:385, 3:385
 NRE2 3:295 3:295
 NRE3 3:295 3:295
 NREQUP 1:69 2:40, 2:41
 NRH 3:9 3:9, 3:96, 3:99, 3:293
 NRUT 3:27 3:164, 3:219, 3:226, 3:385
 NS - 3:429
 NSCHK 3:431 3:380
 NSEGS 1:69 1:69, 1:72, 2:56
 NSF 3:27 3:233, 3:351, 3:375, 3:385
 NSFTOF 3:28 3:122, 3:164, 3:429, 3:430
 NSITRY 1:100 1:100
 NSLEEP 3:431 3:429, 3:429, 3:430, 3:430, 3:430
 NSPARM 1:69 1:11
 NSPARP 1:69 1:11, 2:41, 2:47, 2:50
 NSPC 3:28 3:122, 3:122, 3:164, 3:210, 3:430
 NSRTF 3:28 3:71, 3:122, 3:122, 3:159, 3:164, 3:429, 3:430
 NSRUTD 3:28 3:219, 3:430, 3:430
 NSSPSV 3:429 3:415, 3:431, 3:431
 NSSTAC - 3:415, 3:415
 NSTLIN 3:28 3:122, 3:159
 NSTPFL 3:71 3:71, 3:71
 NSTRIP 3:326 3:326, 3:334
 NSUER 1:20 1:20, 1:40, 1:43
 NTB 3:30 3:30, 3:187, 3:187, 3:192, 3:192, 3:194, 3:194, 3:194, 3:195, 3:195,
 3:195, 3:195, 3:197, 3:197, 3:201, 3:201, 3:202, 3:202, 3:204,
 3:204, 3:208, 3:208, 3:208, 3:208, 3:210, 3:225, 3:225, 3:226,
 3:226
 NTBEND 3:30 3:195, 3:195, 3:210, 3:211
 NTBIDX 3:30 3:187, 3:192, 3:194, 3:195, 3:195, 3:195, 3:197, 3:201, 3:202,
 3:204, 3:208, 3:208, 3:211, 3:225, 3:226
 NTENEX 3:9 3:9, 3:9
 NTIPIT 3:27 3:385
 NTIPTI 3:27 3:385
 NULLHD 3:11 3:144, 3:144, 3:144, 3:144, 3:144, 3:144, 3:146, 3:146, 3:146,
 3:146, 3:146, 3:151, 3:153, 3:153
 NUMCTS 3:27 3:27, 3:27, 3:27, 3:386, 3:386
 NUMHST 3:27 3:27, 3:46
 NUMOUT 2:121 2:96, 2:96, 2:106, 2:112, 2:113, 2:121, 3:66
 NUP 3:12 3:133, 3:133, 3:133, 3:136
 NVARSM 1:69 1:11
 NVARSP 1:69 1:11
 NVDHI 3:27 3:385
 NVHAI 3:433 3:433
 NXISTX 2:114 2:113
 NXMESO 3:288 3:283
 NXMES1 3:288 3:290
 NXMES3 3:288 3:288, 3:288
 NXMES4 3:288 3:288, 3:288, 3:288
 NMESS5 3:288 3:289

NXMES6 3:288 3:288
NXTBF 3:12 3:105,3:131,3:154,3:155,3:158
NXTLED 3:14 3:87,3:87,3:271,3:271,3:271,3:395
NZ - 3:40,3:43,3:43,3:45,3:45,3:79,3:125,3:217
O - 3:65
O.VD 3:47 3:47,3:47
OCMOVE 3:239 3:239,3:278,3:282,3:283
OCTBIT 3:18 3:244,3:269,3:341,3:342
ODELT 3:12 3:12,3:132,3:133,3:133,3:133,3:133,3:135,3:135
ODEVEN 3:18 3:146,3:148,3:150,3:151,3:152,3:167,3:167,3:167
OF - 2:7
OFF - 2:7
OFFDIS 1:24 1:63
OFLSHB 3:50 3:263,3:324
OFLUSH 3:50 3:249,3:250,3:262,3:297,3:307
OLD - 2:7
OLDIMP 3:423 3:423,3:423,3:423
OLDM3 3:285 3:285
OLDM4 3:286 3:285
OLDM5 3:286 3:285
OLDM6 3:286 3:286
OLDMES 3:285 3:285
OLDMSQ 3:286 3:286
OLDP 1:19 2:128,2:128
OLDS 3:360 3:362,3:362
OLST 1:5 1:5,1:6
ONCE - 2:7
ONE - 2:7,2:7
OPCLEA 3:72 3:72,3:73
OPGET 3:67 3:66,3:67,3:72,3:72,3:72,3:72
OPHGO 3:28 3:89,3:90,3:138,3:139,3:242,3:245,3:398,3:402
OPHOST 3:69 3:69,3:69
OPKILL 3:72 3:72,3:72
OPMODE 3:69 3:69
OPNICE 3:71 3:71
OPPANI 3:71 3:71
OPRELO 3:72 3:72,3:72
OPTV - 1:13,1:14
OPTVPA 1:5 3:8,3:8,3:8,3:8
ORG - 1:14,1:14,1:14,1:14,1:25,1:69,2:34,2:34,2:34,3:7,3:7,
3:7,3:7,3:7,3:8,3:8,3:8,3:8,3:31,3:31
OTSLEE 3:128 3:396
OUNPCK 3:56 3:91,3:276
OUR - 2:7,2:7,2:7
OUTER 1:10 1:10
OUTTYP 3:19 3:291
OVRBIT 2:79 2:98,2:107,3:65
OVRRID 2:100 2:98,3:72,3:72
OVRTAB 3:46 3:46,3:65,3:65
OWHEOB 3:54 3:283
P 1:6 1:6,1:6,1:6,1:6,3:71
P\$ - 2:27,2:27
P\$1.6 1:13 1:11
P\$25.6 1:13 1:11
P\$LOOP - 1:10,1:11
P\$PCNT - 1:10,1:11
P% - 2:7,2:7
P.HALT 3:28 3:71

P.RELO 3:28 3:71,3:159,3:164
 P.REST 3:28 3:71
 PACKM - 1:89,1:89,1:91,1:100,1:100.2:54.2:115,2:131,3:56,3:59,
 3:153,3:153,3:256,3:256,3:260,3:270,3:322,3:323,3:363,
 3:390
 PAGE - 1:11,1:11,1:12,1:12,1:12,1:12,1:12,1:12,1:17,1:19,
 1:23,1:24,1:28,1:29,1:29,1:30,1:68,1:70,1:73,1:96,
 1:98,1:99,1:101,2:7,2:7,2:7,2:7,2:7,2:7,2:7,2:7,2:7,
 2:7,2:7,2:9,2:9,2:16,2:16,2:28,2:28,2:35,2:35,2:35
 2:64,2:66,2:70,2:70,2:76,2:76,2:78,2:78,2:78,2:80,
 2:80,2:81,2:82,2:90,2:91,2:121,2:124,2:124,2:125,3:9,
 3:25,3:27,3:27,3:30,3:31,3:31,3:37,3:39,3:40,3:41,
 3:42,3:43,3:44,3:45,3:46,3:46,3:47,3:47,3:47,3:49,
 3:49,3:50,3:62,3:63,3:63,3:63,3:63,3:64,3:65,3:69,
 3:71,3:72,3:72,3:74,3:76,3:76,3:77,3:77,3:81,3:81,
 3:83,3:85,3:86,3:87,3:88,3:93,3:93,3:95,3:100,3:102,
 3:117,3:127,3:131,3:139,3:141,3:142,3:142,3:144,3:146,
 3:147,3:148,3:151,3:159,3:167,3:173,3:173,3:175,3:176,
 3:181,3:181,3:181,3:182,3:183,3:185,3:186,3:212,3:216,
 3:217,3:217,3:219,3:227,3:228,3:228,3:231,3:232,3:241,
 3:262,3:264,3:265,3:267,3:268,3:270,3:271,3:287,3:287,
 3:300,3:301,3:305,3:305,3:314,3:315,3:315,3:315,3:316,
 3:317,3:317,3:317,3:317,3:320,3:320,3:320,3:320,3:321,
 3:326,3:326,3:326,3:326,3:334,3:334,3:341,3:341,3:347,
 3:347,3:350,3:350,3:353,3:354,3:355,3:355,3:357,3:359,
 3:359,3:360,3:361,3:363,3:363,3:381,3:382,3:382,3:391,
 3:392,3:393,3:395,3:397,3:397,3:408,3:429,3:429,3:433,
 3:433,3:434,3:435,3:437,3:438,3:439,3:440,3:444
 PAGEBC 1:28 1:68,2:34,2:37,2:37,3:413
 PAKIOR 3:46,3:46 3:46,3:46
 PAKTYP 1:97,3:18 1:104,3:157,3:272
 PATCHI - 2:7
 PATINI 3:423
 PATTN - 1:10,1:11
 PC - 2:94
 PCCALL 3:350
 PCHALT 1:18 1:18,1:32,1:37,1:37,1:39,1:50,1:50,1:71,1:85,1:86,
 1:89,1:93,1:94,2:126,2:126
 PCHELP 3:29 3:72,3:355
 PCKSUB 3:86 3:145,3:148,3:157,3:186,3:260
 PCORE 3:18
 PCRUN 1:18 1:18,1:50,1:51,1:71,1:86,1:86,1:89,1:94,1:94
 PCSTEP 1:18 1:85
 PDBBLK 2:14 2:14
 PDBCHA 2:26
 PDBLIM 2:14
 PDBOFF 2:14
 PDBPID 2:26
 PDBPKO 2:14
 PDBRAT 2:14
 PDDTTA 3:74,3:74 3:74,3:74
 PDINF 3:30 3:190,3:190,3:192,3:198,3:200,3:200,3:201,3:201,3:205,
 3:205,3:210
 PDIST 3:30 3:30,3:190,3:192,3:198,3:198,3:200,3:200,3:200,3:201,
 3:201,3:201,3:203,3:203,3:205,3:205,3:205,3:205,3:206,
 3:206,3:210,3:210
 PDST 3:30 3:190,3:192,3:198,3:198,3:198,3:200,3:200,3:201,3:201,3:203,
 3:203,3:205,3:205,3:205,3:206,3:206

PFBNUM 3:328 3:326,3:328,3:329,3:332,3:334,3:338
PFLAGS 3:18 3:91,3:160,3:242
PFX 2:80 2:98,2:104,2:105,2:106,2:109,2:110,3:64,3:66
PG - 1:6,1:6,1:11,1:11,1:11,2:7,2:7,2:7,2:7,2:7
PG\$ - 1:5,1:5
PGINIT 1:28 1:11,2:37,2:64
PHDEDL 3:131 3:118,3:131
PHDOWN 3:11
PHFLAG 3:11 3:95,3:131,3:131,3:133,3:134,3:135,3:138,3:139,3:143,
3:144,3:144,3:147,3:148,3:152,3:158,3:161,3:162,3:162,
3:162,3:163,3:163,3:177,3:222,3:233,3:370,3:382,3:383,
3:400
PHUP 3:11 3:11,3:11,3:133,3:134,3:138,3:143,3:148,3:158,3:162,
3:163,3:163,3:163,3:177,3:233,3:370,3:370,3:382,3:383,
3:400
PHYSIC - 1:13,1:14,1:14,1:69,2:34,2:34,3:7,3:7,3:7,3:7,
3:7,3:7,3:7,3:8
PID 1:19 1:74,1:90,2:21,2:33,2:88,2:89,2:119,2:120,2:136,3:55,
3:55,3:59,3:63,3:80,3:83,3:87,3:94,3:99,3:118,3:118,
3:118,3:118,3:118,3:119,3:122,3:124,3:128,3:138,3:139,
3:159,3:168,3:185,3:210,3:214,3:232,3:236,3:237,3:253,
3:275,3:287,3:290,3:292,3:317,3:318,3:319,3:321,3:321,
3:323,3:324,3:324,3:331,3:331,3:337,3:337,3:345,3:349,
3:349,3:363,3:428,3:430
PID3 1:24 1:74
PIDGET 1:21 1:21,1:74,1:74,2:56,2:128,3:117
PIDRCL 1:17 1:74,1:74,2:56,2:56,3:117
PIDSTO 1:17 1:74
PIDTOT 1:21 1:74,2:56
PILEND 1:29 1:41,3:373
PILLOP - 1:10,1:11
PILLOV 1:44,1:44 1:24,1:44
PILNUM 1:29 1:29,1:44
PILOPS 1:29 1:41,3:373
PILOVP 1:29 1:44,1:44,3:72,3:373
PKCACT 1:98 1:101,1:101,1:101,1:106,1:107,1:107,1:109,1:112,1:116,
1:117,3:84,3:357,3:357,3:419
PKCADD 1:98 1:105,1:107,1:108,1:108,1:110,1:112,1:117,3:419
PKCBFA 1:98 1:108,1:110,1:114
PKCBLT 1:114 1:109,1:112
PKCCLA 1:98 1:98,1:116
PKCCLL 1:98 1:116
PKCCLR 1:115 1:108,1:111,1:115
PKCDON 1:98 1:108,1:110,1:114
PKCETY 1:98 1:109,1:111,1:111,1:112
PKCEXT 1:98 1:101,1:115,1:117,3:419
PKCFHA 1:98 1:106
PKCFHO 1:98 1:106
PKCFID 1:98 1:106,1:116
PKCFIM 1:98 1:106
PKCFLN 1:98 1:106,1:106,1:107
PKCHOL 3:147 3:147,3:147,3:159
PKCIC 1:106 1:104,1:104,3:353
PKCIID 1:101 1:115
PKCIST 1:98 1:104,1:106,1:108,1:109
PKCITB 1:101 1:101,1:116,1:116
PKCITL 1:101 1:116
PKCLEN 1:98 1:108,1:108,1:108,1:110,1:112,1:117

PKCLHA 1:98 1:101,1:106,1:106,1:107
 PKCLHO 1:98 1:101,1:106,1:107
 PKCLID 1:98 1:106,1:107,1:110,1:115
 PKCLIM 1:98 1:106,1:106,1:107,1:107
 PKCLLN 1:98 1:106,1:106
 PKCLMK 1:97 1:108,1:114
 PKCLOK - 1:98,1:101,1:101,1:116,1:117,3:353,3:353,3:353,3:357,
 3:357,3:357
 PKCLTO 1:96 1:107,1:108
 PKCMAX 1:98 1:112,1:112
 PKCMYI 1:98 1:104,1:116
 PKCNIM 1:98 1:104
 PKCOC 1:110 1:103,3:357
 PKCORE - 1:15
 PKCOST 1:98 1:110,1:110,1:112
 PKCOTM 1:98 1:112,1:112
 PKCRAT 1:96 1:101,1:117
 PKCRCR 1:98 1:101,1:101,1:104,1:108,1:115,1:116,1:116,3:422
 PKCSRF 1:98 1:111,1:112,1:112,1:112,1:117
 PKCSSF 1:98 1:101,1:107,1:108,1:109,1:110,1:111,1:117
 PKCST 1:98 1:101,1:101,1:101,1:104,1:106,1:107,1:107,1:107,1:108,
 1:108,1:109,1:109,1:110,1:110,1:110,1:110,1:111,1:112,
 1:112,1:112,1:114,1:115,1:115,1:116,1:117,3:84,3:357,
 3:357,3:419,3:419
 PKCSTO 1:96 1:110,1:113
 PKCTIM 1:98 1:101,1:115
 PKCTMK 1:97 1:107,1:109,1:114
 PKCTMX 1:96 1:104,1:108
 PKCTRY 1:96,1:96 1:116
 PKCTYH 1:98 1:101
 PKCTYP 1:98 1:107,1:112,1:117,3:419
 PKCUPT 1:115 1:101,1:107,1:108,1:110,1:113,1:117
 PKG - 3:7,3:7
 PKGCOD 3:7 3:7,3:7,3:42,3:49,3:85,3:117,3:125,3:131,3:142,3:144,
 3:146,3:147,3:159,3:227,3:445
 PKGLOD 3:6 3:7
 PKGTAB 3:125
 PKGVAR 3:7,3:7,3:8 3:7,3:7,3:445
 PKGVST 3:6 3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:8
 PKTCOD 3:19 3:273,3:275,3:276,3:278,3:295,3:296,3:324
 PKTH 1:97,3:17 1:97,1:106,1:106,1:106,1:107,1:107,3:19,3:146,
 3:247,3:259,3:260,3:263,3:270,3:272,3:276,3:276,3:278,
 3:278,3:285,3:322
 PKTNUM 3:19 3:249,3:250,3:250,3:250,3:250,3:250,3:263,3:285
 PKTS8 3:45 3:298,3:302,3:303
 PLRCOM 1:98 1:109,1:109,1:112,1:112,1:117
 PLRPKC 1:98 1:109,1:112,1:117,1:117
 POINT 3:37 3:56,3:58,3:59,3:238,3:239,3:322,3:388,3:388,3:388,
 3:388,3:389,3:389,3:389,3:412,3:412,3:412,3:412,3:412,
 3:412,3:412,3:413
 POKE 2:21,2:33 2:21,2:33,3:63,3:128
 POKEM 3:55 3:121,3:132,3:133,3:163,3:214,3:354
 POLBLT 1:49 1:45,1:46,3:84
 POLLER 2:81 2:81,2:81
 POLRLD 1:9,1:101 1:89,1:101
 POLTIM 2:78 2:82,3:63
 POSITI - 2:7,2:7
 POST 2:19

PPFLAG 3:182 3:190,3:198,3:200
PREFIX 2:80 2:98,2:104,2:105,2:107,2:110,2:110,3:66
PRIBIT 3:18 3:236,3:263,3:278,3:290,3:302,3:302,3:303,3:305,3:305,
3:305,3:308,3:309,3:314
PRILED 3:21 3:91,3:244,3:244,3:269,3:296,3:394,3:394
PRINTC 3:345 3:341,3:342,3:344,3:345
PROAMP 1:68 1:78
PROC - 2:19
PROCBT 1:19 1:42,1:45,1:49,1:49,1:52,1:58,1:63,1:63,1:75,1:78,
1:92,2:58,2:58,2:60,2:70,2:96,2:107,2:119,2:126,2:128
PROCD 1:22 1:78,1:79,1:82,1:82,1:84,1:84
PROCES 2:15
PROCEX 1:70 1:79,1:79,1:92,2:61,2:115,3:369
PROCNO 1:19 1:50,1:52,1:79,1:84,1:84,1:92,1:95,2:62
PROHLT 1:68 1:78
PROHNG 1:68 2:59
PROIOR 1:70 1:78,2:61
PROKIL 1:68 1:82,2:61
PROPDL 3:43 3:44,3:138,3:178
PROTCN 3:43 3:43,3:136
PRRATE 1:16 1:90,2:60
PRTIME 1:72 1:90,2:60,2:61
PSBACT 2:76 2:87,2:89
PSBBRK 2:76 2:87
PSBCTL 2:76 2:87,2:87,2:89,2:89
PSBDAT 2:76 2:88,2:89
PSBECH 2:76
PSBFRE 2:76 2:87,2:88,2:89
PSBINT 2:76
PSBONE 2:76 2:86
PSBOUT 2:76 2:87,2:89,2:89
PSBOVR 2:76
PSBSTA 2:76 2:87,2:87,2:89
PSBTAB 2:86 2:85,2:86
PSBTWO 2:76 2:86
PSDATA 1:98 1:101,1:111
PSE 1:1,3:1 1:15,1:29,1:41,1:44,1:96,2:34,3:5,3:40,3:40,3:40,3:40,
3:40,3:40,3:42,3:42,3:42,3:43,3:43,3:43,3:43,3:45,
3:45,3:45,3:46,3:72,3:78,3:79,3:79,3:313,3:351,3:373,
3:373,3:399,3:415,3:415
PSEMIC 3:347 3:343,3:343
PSETUL 1:98 1:106,1:106,1:111,1:111
PSETUP 1:98 1:98,1:106,1:111,1:112
PSTOPS 3:71 3:71
PSTPFL 3:71 3:71,3:71
PTIME 3:326 3:326,3:331,3:334,3:337
PTPRDR 1:18 1:34
PTR 1:1,1:18 1:18,1:29,1:34,2:13,2:13,2:13,2:26
PTRAD 1:34 1:34
PTRB 1:29 1:34
PTRBC 1:29
PTRBL 1:29 1:29,1:34,1:34
PTRCS 1:29
PTRDN 1:29
PTRFB 1:29
PTRFLG 1:29
PTRIP 1:29 1:34,1:34
PTROP 1:29 1:34

PTRSR 1:29
 PTRST 1:29
 PUT - 2:7
 PUTARG 3:66 3:66,3:70,3:72
 Q1 1:30 1:30
 Q50 1:30 1:51,1:53
 Q70 1:30 1:53
 QERTYP 3:18 3:263,3:278,3:281,3:293,3:294,3:294,3:305,3:305,3:305,
 3:308,3:309
 QQHAD 1:19 1:32,2:62
 QQHCT 1:19 1:32,2:62,2:62
 QQHPC 1:19 1:32,2:62
 QSUBR 1:32 1:31,1:31,1:32,1:32
 QT 3:17 3:236,3:283,3:285
 QTOADR 3:46 3:410
 QUIT - 1:34,1:39,1:39,1:41,1:42,1:42,1:42,1:44,1:47,1:47,
 1:47,1:47,1:49,1:50,1:50,1:50,1:50,1:51,1:52,1:55,
 1:56,1:57,1:57,1:58,1:59,1:60,1:60,1:63,1:63,1:64,
 1:64,1:65,1:65,1:66,1:74,1:74,1:76,1:77,1:77,1:80,
 1:81,1:81,1:81,1:82,1:82,1:84,1:84,1:88,1:102,1:103,
 2:52,2:52,2:54,2:54,2:55,2:55,2:56,2:85,3:49,3:49,
 3:117,3:117
 QUITAD 1:19 1:30,1:30,1:32,2:52
 QUITFL 1:22 1:30,1:31,1:32
 QUITPC 1:19 1:30,1:30,1:30,1:31,1:31,1:31
 QUITRT 1:19 1:30,1:31,2:62,2:62
 QUITST 1:19 1:30,1:31
 QUITTM 1:19 1:30,1:30,1:31,1:31
 QUITV 1:19 1:30,1:84
 QUITXT 2:114 2:97,2:113
 QUNPAC 3:56 3:53
 QUTPAT - 1:85,1:88,1:88,1:88,1:90,1:90,1:90,1:90,1:93,1:93,
 1:93,1:94,1:94,2:85,2:89,2:126,2:126,2:126,2:127,2:128,
 2:128,3:401,3:401
 QX 1:19 1:30,1:30,1:31,1:31,1:31,1:31,1:31
 R - 1:102,3:72
 RO - 1:42,1:50,1:55,1:56,1:85,1:89,1:94,1:95,1:101,1:107,
 1:108,1:109,1:110,1:112,1:117,2:84,2:128,3:49,3:49,
 3:71,3:71,3:77,3:81,3:85,3:90,3:94,3:100,3:242,3:245,
 3:261,3:262,3:263,3:264,3:264,3:265,3:265,3:266,3:266,
 3:267,3:275,3:276,3:278,3:278,3:279,3:280,3:281,3:286,
 3:286,3:288,3:289,3:296,3:297,3:298,3:298,3:301,3:303,
 3:303,3:304,3:306,3:308,3:309,3:312,3:313,3:315,3:324,
 3:324,3:373,3:382,3:391,3:392,3:394,3:395,3:396,3:396,
 3:398,3:421,3:421,3:421,3:423
 R1 - 1:9,1:30,1:30,1:30,1:30,1:30,1:30,1:31,1:32,1:32,
 1:32,1:32,1:33,1:33,1:33,1:33,1:33,1:34,1:34,1:34,
 1:34,1:35,1:35,1:35,1:36,1:36,1:36,1:36,1:36,1:36,
 1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,1:37,
 1:37,1:38,1:38,1:38,1:38,1:38,1:38,1:39,1:39,1:39,
 1:39,1:39,1:40,1:40,1:40,1:40,1:40,1:40,1:40,1:40,
 1:40,1:41,1:41,1:43,1:43,1:43,1:44,1:45,1:45,1:45,
 1:46,1:46,1:46,1:46,1:47,1:47,1:47,1:47,1:47,1:48,
 1:49,1:49,1:50,1:50,1:50,1:50,1:50,1:50,1:50,1:50,
 1:50,1:50,1:51,1:51,1:51,1:51,1:51,1:51,1:51,1:51,
 1:51,1:51,1:52,1:52,1:52,1:52,1:52,1:52,1:52,1:52,
 1:52,1:52,1:53,1:54,1:54,1:54,1:54,1:55,1:55,1:55,
 1:55,1:55,1:56,1:56,1:56,1:56,1:56,1:57,1:58,1:58,

1:58,1:59,1:59,1:59,1:59,1:59,1:59,1:60,1:60,
1:60,1:60,1:60,1:60,1:61,1:61,1:61,1:61,1:61,
1:64,1:64,1:65,1:65,1:66,1:67,1:67,1:67,1:67,1:73,
1:74,1:74,1:74,1:74,1:74,1:74,1:75,1:75,1:75,
1:75,1:76,1:76,1:77,1:77,1:77,1:77,1:77,1:77,
1:77,1:77,1:78,1:78,1:78,1:79,1:79,1:79,1:79,
1:79,1:79,1:79,1:79,1:79,1:79,1:80,1:81,1:82,1:82,
1:82,1:82,1:82,1:82,1:84,1:84,1:85,1:85,1:85,1:85,
1:85,1:85,1:85,1:86,1:86,1:86,1:86,1:86,1:86,
1:86,1:86,1:86,1:86,1:86,1:86,1:86,1:86,1:87,
1:87,1:87,1:87,1:87,1:88,1:88,1:88,1:89,1:89,
1:89,1:89,1:89,1:89,1:90,1:90,1:90,1:90,1:91,1:91,
1:91,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,
1:92,1:92,1:92,1:93,1:93,1:93,1:93,1:93,1:93,1:93,
1:93,1:93,1:93,1:93,1:93,1:93,1:93,1:93,1:93,1:93,
1:94,1:94,1:94,1:94,1:94,1:94,1:94,1:94,1:94,1:94,
1:94,1:94,1:94,1:94,1:94,1:94,1:95,1:95,1:95,
1:100,1:100,1:100,1:100,1:100,1:100,1:100,1:101,1:101,
1:101,1:101,1:101,1:101,1:102,1:102,1:102,1:102,1:102,
1:102,1:103,1:103,1:103,1:103,1:103,1:103,1:103,1:103,
1:104,1:104,1:104,1:105,1:105,1:105,1:105,1:106,1:106,
1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:106,
1:107,1:107,1:107,1:107,1:108,1:108,1:108,1:108,1:108,
1:108,1:108,1:108,1:109,1:110,1:110,1:110,1:111,1:111,
1:112,1:112,1:112,1:112,1:114,1:114,1:114,1:115,1:115,
1:115,1:115,1:115,1:115,1:115,1:116,1:116,1:116,1:117,
1:117,1:117,1:117,1:117,1:117,1:118,1:118,1:118,1:118,
2:23,2:36,2:38,2:38,2:38,2:38,2:38,2:39,2:39,2:39,
2:39,2:40,2:40,2:40,2:41,2:41,2:41,2:41,2:41,2:41,
2:43,2:43,2:43,2:43,2:43,2:44,2:45,2:45,2:45,2:46,
2:47,2:47,2:47,2:47,2:48,2:48,2:49,2:49,2:49,2:49,
2:49,2:49,2:49,2:50,2:50,2:50,2:50,2:51,2:51,2:51,
2:51,2:52,2:52,2:52,2:52,2:52,2:53,2:53,2:53,2:53,
2:54,2:54,2:55,2:55,2:56,2:56,2:56,2:56,2:56,2:56,
2:56,2:56,2:56,2:61,2:61,2:61,2:61,2:62,2:62,
2:62,2:62,2:62,2:62,2:62,2:63,2:64,2:64,2:64,2:64,
2:64,2:65,2:65,2:66,2:66,2:66,2:68,2:68,2:68,2:70,
2:70,2:70,2:81,2:81,2:81,2:81,2:81,2:82,2:82,2:82,
2:83,2:84,2:85,2:87,2:88,2:88,2:88,2:89,2:89,
2:89,2:89,2:90,2:90,2:90,2:91,2:93,2:93,2:93,2:93,
2:93,2:93,2:94,2:96,2:96,2:96,2:96,2:96,2:96,2:96,
2:96,2:96,2:96,2:96,2:98,2:98,2:98,2:98,2:98,2:105,
2:105,2:105,2:105,2:105,2:105,2:106,2:106,2:106,2:107,
2:109,2:109,2:110,2:110,2:110,2:112,2:112,2:112,2:112,
2:112,2:112,2:112,2:112,2:113,2:113,2:113,2:113,2:117,
2:117,2:117,2:117,2:117,2:117,2:118,2:118,2:119,2:119,2:119,
2:119,2:119,2:120,2:120,2:120,2:121,2:121,2:121,2:121,
2:121,2:121,2:121,2:121,2:121,2:121,2:122,2:122,2:122,
2:122,2:122,2:125,2:125,2:125,2:125,2:126,2:126,2:128,
2:128,2:128,2:129,2:129,2:129,2:129,2:130,2:130,2:131,
2:133,2:133,2:135,2:136,2:136,2:136,2:136,2:136,2:136,
2:136,2:136,2:136,3:49,3:49,3:49,3:50,3:50,3:50,
3:50,3:50,3:51,3:51,3:51,3:51,3:51,3:51,3:51,3:51,
3:51,3:52,3:52,3:52,3:52,3:52,3:52,3:53,3:54,3:54,
3:54,3:54,3:54,3:55,3:55,3:55,3:56,3:56,3:56,3:56,
3:56,3:56,3:56,3:56,3:56,3:56,3:56,3:56,3:56,3:56,
3:57,3:57,3:57,3:57,3:57,3:57,3:57,3:57,3:57,3:57,
3:57,3:58,3:58,3:59,3:59,3:59,3:59,3:59,3:59,

3:221,3:221,3:221,3:221,3:221,3:221,3:221,3:221,3:221,
3:222,3:223,3:223,3:223,3:223,3:223,3:223,3:223,3:223,
3:223,3:223,3:223,3:223,3:224,3:224,3:224,3:224,
3:225,3:225,3:225,3:225,3:225,3:226,3:226,3:226,3:226,
3:227,3:227,3:227,3:227,3:227,3:227,3:227,3:227,3:228,
3:228,3:228,3:228,3:228,3:229,3:229,3:229,3:229,
3:229,3:232,3:232,3:232,3:232,3:232,3:233,3:235,3:236,
3:236,3:236,3:236,3:237,3:237,3:237,3:237,3:237,
3:238,3:238,3:238,3:238,3:238,3:239,3:239,3:239,
3:239,3:239,3:241,3:241,3:241,3:242,3:242,3:242,3:242,
3:242,3:242,3:242,3:242,3:243,3:244,3:244,3:244,3:244,
3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,
3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,
3:244,3:244,3:244,3:244,3:245,3:245,3:245,3:245,
3:245,3:245,3:246,3:246,3:246,3:246,3:246,3:246,3:246,
3:246,3:246,3:247,3:247,3:247,3:247,3:248,3:248,
3:248,3:248,3:248,3:248,3:248,3:248,3:249,3:249,
3:249,3:249,3:249,3:249,3:249,3:250,3:250,3:250,
3:250,3:250,3:250,3:251,3:251,3:251,3:252,3:252,
3:252,3:252,3:252,3:252,3:252,3:252,3:252,3:252,
3:253,3:253,3:253,3:256,3:256,3:256,3:256,3:256,
3:256,3:256,3:256,3:258,3:258,3:258,3:258,3:259,
3:259,3:259,3:259,3:259,3:259,3:259,3:259,3:259,
3:259,3:259,3:260,3:260,3:260,3:260,3:260,3:260,
3:260,3:260,3:260,3:260,3:260,3:263,3:263,3:263,
3:263,3:263,3:265,3:265,3:265,3:265,3:265,3:265,
3:265,3:265,3:265,3:265,3:265,3:265,3:265,3:265,
3:265,3:266,3:266,3:266,3:266,3:266,3:266,3:267,
3:267,3:267,3:267,3:267,3:267,3:267,3:267,3:267,
3:267,3:267,3:267,3:268,3:268,3:268,3:268,3:268,
3:268,3:268,3:268,3:269,3:269,3:269,3:269,3:269,
3:269,3:269,3:269,3:270,3:270,3:270,3:270,3:270,
3:270,3:271,3:271,3:271,3:271,3:271,3:271,3:271,
3:271,3:271,3:271,3:271,3:271,3:272,3:272,3:272,
3:272,3:272,3:273,3:273,3:274,3:274,3:274,3:275,
3:275,3:275,3:275,3:276,3:276,3:276,3:276,3:276,
3:277,3:277,3:277,3:277,3:278,3:278,3:278,3:278,
3:278,3:278,3:278,3:278,3:278,3:279,3:279,
3:280,3:280,3:281,3:282,3:282,3:282,3:283,3:283,
3:283,3:284,3:284,3:284,3:285,3:285,3:285,3:285,
3:285,3:285,3:285,3:285,3:285,3:286,3:286,3:286,
3:286,3:286,3:286,3:286,3:286,3:286,3:286,3:286,
3:286,3:286,3:286,3:287,3:287,3:287,3:287,3:287,
3:287,3:287,3:287,3:288,3:288,3:288,3:288,3:288,
3:288,3:289,3:289,3:289,3:289,3:290,3:290,3:290,
3:290,3:290,3:290,3:290,3:292,3:292,3:292,3:292,
3:293,3:293,3:293,3:294,3:294,3:294,3:295,3:295,
3:295,3:295,3:295,3:295,3:296,3:296,3:296,3:296,
3:296,3:296,3:296,3:296,3:296,3:296,3:296,3:296,
3:296,3:296,3:296,3:296,3:296,3:296,3:296,3:297,
3:297,3:297,3:297,3:298,3:298,3:298,3:298,3:298,
3:298,3:298,3:299,3:299,3:300,3:300,3:300,3:301,3:301,
3:301,3:302,3:302,3:302,3:302,3:303,3:303,3:303,
3:304,3:304,3:304,3:304,3:304,3:305,3:305,3:305,
3:305,3:305,3:305,3:305,3:305,3:305,3:305,3:306,
3:306,3:306,3:306,3:306,3:306,3:309,3:309,3:311,3:315,
3:315,3:315,3:315,3:317,3:317,3:319,3:319,3:319,3:319.

3:319,3:319,3:319,3:320,3:320,3:321,3:321,3:322,3:322,
3:322,3:322,3:322,3:322,3:322,3:322,3:322,3:324,3:324,
3:324,3:324,3:324,3:324,3:324,3:324,3:324,3:324,3:324,
3:326,3:326,3:328,3:328,3:328,3:328,3:328,3:328,3:329,
3:329,3:329,3:329,3:329,3:329,3:330,3:330,3:330,3:330,
3:331,3:331,3:332,3:332,3:332,3:332,3:332,3:332,3:333,
3:334,3:334,3:335,3:335,3:336,3:336,3:336,3:336,3:336,
3:336,3:336,3:337,3:338,3:338,3:338,3:338,3:342,3:342,
3:342,3:343,3:343,3:343,3:343,3:343,3:344,3:344,3:344,
3:344,3:345,3:345,3:345,3:345,3:345,3:346,3:346,3:347,
3:347,3:347,3:347,3:347,3:347,3:348,3:348,3:348,3:349,
3:349,3:349,3:349,3:350,3:350,3:351,3:351,3:351,3:351,
3:351,3:351,3:351,3:351,3:351,3:351,3:351,3:351,3:351,
3:351,3:351,3:351,3:351,3:351,3:352,3:352,3:352,
3:352,3:352,3:352,3:352,3:352,3:352,3:352,3:352,3:352,
3:353,3:353,3:354,3:354,3:355,3:355,3:355,3:355,3:355,
3:355,3:355,3:355,3:355,3:355,3:355,3:356,3:356,3:356,
3:356,3:356,3:357,3:357,3:357,3:357,3:357,3:357,3:362,
3:362,3:362,3:362,3:362,3:362,3:362,3:362,3:362,3:366,
3:366,3:367,3:368,3:368,3:369,3:369,3:369,3:369,3:369,
3:369,3:369,3:369,3:370,3:370,3:370,3:370,3:370,3:370,
3:370,3:370,3:370,3:370,3:371,3:371,3:371,3:371,3:371,
3:371,3:371,3:371,3:371,3:371,3:371,3:371,3:371,3:371,
3:372,3:372,3:372,3:372,3:372,3:373,3:373,3:373,3:373,
3:373,3:374,3:374,3:374,3:374,3:374,3:374,3:374,3:374,
3:374,3:374,3:375,3:375,3:375,3:375,3:375,3:375,3:375,
3:375,3:375,3:375,3:375,3:375,3:375,3:375,3:375,3:375,
3:375,3:375,3:376,3:376,3:376,3:376,3:376,3:376,3:376,
3:376,3:376,3:376,3:376,3:376,3:376,3:376,3:377,3:377,
3:377,3:377,3:377,3:377,3:377,3:377,3:377,3:377,3:377,
3:377,3:377,3:377,3:378,3:378,3:378,3:378,3:378,3:381,
3:381,3:381,3:381,3:381,3:381,3:381,3:381,3:381,3:381,
3:381,3:381,3:382,3:382,3:382,3:382,3:383,3:383,3:383,
3:383,3:384,3:384,3:384,3:384,3:384,3:385,3:385,3:385,
3:385,3:385,3:385,3:386,3:386,3:386,3:386,3:387,3:387,
3:387,3:387,3:389,3:389,3:390,3:391,3:391,3:391,3:392,
3:392,3:392,3:392,3:392,3:392,3:392,3:393,3:393,3:393,
3:393,3:394,3:394,3:394,3:394,3:394,3:394,3:395,3:396,
3:396,3:396,3:396,3:396,3:396,3:396,3:396,3:396,3:396,
3:396,3:397,3:397,3:397,3:397,3:398,3:398,3:398,3:398,
3:399,3:399,3:399,3:399,3:400,3:400,3:400,3:400,3:400,
3:401,3:401,3:401,3:401,3:402,3:402,3:402,3:402,3:402,
3:402,3:402,3:402,3:402,3:402,3:402,3:402,3:402,3:402,
3:402,3:402,3:402,3:403,3:403,3:403,3:405,3:407,3:407,
3:407,3:408,3:408,3:408,3:410,3:410,3:410,3:410,3:411,
3:412,3:412,3:412,3:412,3:412,3:412,3:412,3:412,3:412,
3:412,3:412,3:412,3:412,3:412,3:413,3:413,3:418,3:418,
3:418,3:418,3:419,3:419,3:420,3:420,3:420,3:420,3:420,
3:420,3:420,3:420,3:420,3:420,3:421,3:421,3:421,3:421,
3:421,3:421,3:421,3:421,3:422,3:422,3:422,3:422,3:422,
3:422,3:422,3:423,3:423,3:423,3:423,3:423,3:423,3:423,
3:423,3:423,3:423,3:423,3:423,3:423,3:423,3:423,3:424,
3:424,3:424,3:427,3:428,3:428,3:428,3:428,3:428,3:428,
3:428,3:429,3:429,3:429,3:429,3:429,3:430,3:430,3:430,
3:430,3:431,3:431,3:431,3:431,3:434,3:434,3:434,3:434,
3:434,3:434,3:434,3:434,3:434,3:434,3:435,3:435,3:435,
3:435,3:435,3:435,3:435,3:437,3:437,3:437,3:437,3:437,
3:437,3:437,3:438,3:438,3:438,3:438,3:438,3:438,3:438

3:438,3:438,3:438,3:438,3:439,3:439,3:439,3:439,3:439,
3:439,3:440,3:440,3:440,3:440,3:440,3:440,3:440,
3:441,3:441,3:441,3:441,3:441,3:441,3:442,3:442,3:442,
3:442,3:442,3:442,3:442,3:442,3:442,3:442,3:443,
3:443,3:443,3:443,3:444,3:444,3:444,3:444,3:444,
3:444,3:444

R2 -
1:9,1:30,1:30,1:31,1:31,1:31,1:31,1:31,1:32,1:32,
1:32,1:32,1:32,1:33,1:33,1:33,1:34,1:35,1:36,1:36,
1:36,1:36,1:36,1:36,1:36,1:37,1:37,1:37,1:37,1:37,
1:37,1:38,1:38,1:38,1:39,1:39,1:39,1:40,1:40,1:41,
1:41,1:43,1:47,1:47,1:50,1:51,1:51,1:51,1:51,1:51,
1:51,1:52,1:52,1:52,1:52,1:52,1:52,1:54,1:54,
1:54,1:54,1:54,1:55,1:55,1:55,1:55,1:55,1:56,1:56,
1:56,1:57,1:57,1:57,1:58,1:58,1:58,1:59,1:59,1:59,
1:65,1:65,1:66,1:66,1:66,1:67,1:67,1:67,1:67,
1:74,1:74,1:74,1:74,1:74,1:74,1:74,1:74,1:75,
1:75,1:75,1:75,1:76,1:76,1:76,1:76,1:76,1:77,1:77,
1:77,1:77,1:77,1:77,1:77,1:77,1:78,1:78,1:78,1:78,
1:79,1:79,1:79,1:79,1:79,1:80,1:80,1:81,1:81,1:81,
1:82,1:82,1:85,1:85,1:85,1:85,1:85,1:85,1:85,
1:85,1:85,1:87,1:87,1:88,1:88,1:88,1:88,1:88,
1:88,1:88,1:89,1:89,1:89,1:89,1:89,1:89,1:90,
1:90,1:90,1:90,1:90,1:90,1:91,1:91,1:91,1:91,1:91,
1:91,1:91,1:91,1:91,1:91,1:92,1:92,1:92,1:92,1:92,
1:92,1:93,1:93,1:93,1:93,1:93,1:93,1:93,1:93,1:93,
1:93,1:93,1:93,1:94,1:94,1:94,1:94,1:94,1:94,1:94,
1:94,1:94,1:94,1:94,1:94,1:94,1:95,1:95,1:95,
1:95,1:95,1:95,1:95,1:95,1:95,1:100,1:100,1:100,1:101,
1:101,1:101,1:101,1:101,1:101,1:102,1:102,1:103,1:103,
1:104,1:104,1:106,1:109,1:110,1:110,1:110,1:112,1:112,
1:112,1:112,1:112,1:112,1:112,1:112,1:112,1:114,
1:114,1:116,1:116,1:116,1:116,1:116,1:117,1:117,1:117,
1:117,1:117,2:23,2:36,2:36,2:38,2:39,2:39,2:39,2:41,
2:43,2:43,2:43,2:43,2:45,2:45,2:45,2:45,2:46,
2:47,2:47,2:47,2:47,2:47,2:47,2:47,2:47,2:48,
2:49,2:49,2:49,2:49,2:50,2:50,2:50,2:50,2:50,2:50,
2:52,2:52,2:52,2:52,2:52,2:53,2:54,2:54,2:54,2:54,
2:54,2:54,2:54,2:55,2:55,2:55,2:56,2:56,2:56,2:58,
2:58,2:58,2:58,2:60,2:60,2:60,2:60,2:62,2:62,2:62,
2:63,2:63,2:63,2:64,2:64,2:64,2:64,2:65,2:65,
2:68,2:68,2:68,2:68,2:70,2:70,2:70,2:70,2:82,2:90,
2:90,2:90,2:90,2:90,2:90,2:90,2:90,2:90,2:90,2:90,
2:91,2:91,2:91,2:91,2:91,2:91,2:93,2:93,2:93,
2:93,2:93,2:93,2:96,2:96,2:98,2:98,2:98,2:104,
2:104,2:105,2:105,2:105,2:105,2:105,2:105,2:105,
2:105,2:105,2:105,2:105,2:112,2:112,2:113,2:115,2:115,
2:115,2:115,2:115,2:115,2:115,2:115,2:115,2:115,
2:115,2:121,2:121,2:121,2:121,2:121,2:121,2:122,
2:126,2:126,2:126,2:126,2:126,2:126,2:127,2:128,
2:128,2:128,2:128,2:128,2:128,2:128,2:128,2:128,
2:129,2:129,2:130,2:131,2:131,2:131,2:131,2:131,
2:131,2:131,2:131,2:131,2:134,2:134,2:134,2:135,
3:49,3:50,3:50,3:50,3:50,3:50,3:50,3:50,3:51,
3:51,3:51,3:51,3:51,3:51,3:52,3:52,3:52,3:52,3:52,
3:52,3:53,3:53,3:53,3:53,3:53,3:53,3:53,3:53,3:53,
3:53,3:54,3:54,3:54,3:55,3:55,3:55,3:55,3:55,3:56,
3:56,3:56,3:56,3:56,3:56,3:56,3:56,3:57,3:57,
3:57,3:57,3:57,3:57,3:58,3:58,3:58,3:59,3:59,3:60,

3:60,3:60,3:60,3:60,3:69,3:69,3:69,3:70,3:70,3:70,
3:79,3:79,3:79,3:79,3:79,3:79,3:83,3:83,3:83,
3:83,3:83,3:83,3:85,3:85,3:85,3:87,3:87,3:87,3:87,
3:87,3:88,3:89,3:91,3:91,3:91,3:91,3:91,3:91,3:91,
3:93,3:93,3:95,3:95,3:95,3:97,3:97,3:98,3:98,3:98,
3:98,3:98,3:98,3:100,3:100,3:100,3:100,3:100,
3:100,3:100,3:100,3:100,3:102,3:102,3:102,3:103,3:103,
3:103,3:103,3:103,3:103,3:104,3:104,3:104,3:105,3:105,
3:105,3:105,3:105,3:105,3:105,3:106,3:106,3:106,3:106,
3:106,3:106,3:107,3:107,3:107,3:107,3:107,3:107,3:107,
3:111,3:112,3:112,3:112,3:112,3:115,3:115,3:115,3:117,
3:117,3:117,3:117,3:118,3:118,3:118,3:118,3:118,3:118,
3:118,3:119,3:119,3:121,3:121,3:122,3:122,3:122,3:122,
3:127,3:127,3:131,3:131,3:131,3:131,3:131,3:131,3:131,
3:131,3:131,3:131,3:132,3:133,3:136,3:136,3:136,3:136,
3:136,3:136,3:136,3:136,3:136,3:136,3:137,3:137,
3:137,3:137,3:137,3:141,3:141,3:141,3:141,3:141,3:141,
3:141,3:141,3:142,3:142,3:142,3:142,3:142,3:142,3:143,
3:143,3:144,3:144,3:144,3:144,3:144,3:145,3:145,
3:147,3:147,3:148,3:148,3:148,3:148,3:150,3:150,3:151,
3:151,3:151,3:151,3:151,3:152,3:152,3:152,3:152,3:152,
3:152,3:152,3:152,3:153,3:153,3:153,3:153,3:153,3:153,
3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:153,
3:153,3:153,3:154,3:154,3:154,3:154,3:154,3:154,3:154,
3:155,3:155,3:155,3:155,3:155,3:158,3:158,3:158,3:158,
3:159,3:159,3:159,3:161,3:161,3:162,3:162,3:162,3:162,
3:162,3:162,3:162,3:163,3:163,3:164,3:164,3:164,3:164,
3:164,3:164,3:164,3:166,3:166,3:166,3:166,3:166,3:168,
3:169,3:169,3:169,3:169,3:169,3:169,3:169,3:169,3:169,
3:169,3:169,3:169,3:169,3:170,3:170,3:171,3:175,3:175,
3:175,3:175,3:175,3:175,3:177,3:177,3:178,3:187,3:187,3:187,
3:188,3:188,3:188,3:188,3:188,3:189,3:189,3:189,3:189,3:189,
3:189,3:189,3:189,3:189,3:191,3:192,3:192,3:192,3:192,
3:192,3:192,3:192,3:192,3:192,3:194,3:194,3:194,3:195,
3:195,3:195,3:195,3:195,3:195,3:195,3:195,3:195,3:197,
3:197,3:197,3:197,3:198,3:198,3:199,3:199,3:199,3:200,
3:200,3:201,3:201,3:201,3:201,3:201,3:202,3:202,3:202,
3:202,3:203,3:203,3:204,3:204,3:204,3:204,3:204,3:204,
3:204,3:204,3:204,3:205,3:205,3:205,3:206,3:206,3:206,
3:206,3:207,3:207,3:207,3:207,3:207,3:208,3:208,3:208,
3:208,3:208,3:209,3:209,3:209,3:209,3:209,3:209,3:209,
3:209,3:210,3:210,3:210,3:210,3:210,3:212,3:212,3:212,
3:212,3:212,3:213,3:214,3:214,3:215,3:215,3:215,3:215,
3:216,3:216,3:217,3:217,3:217,3:218,3:218,3:218,3:218,
3:218,3:218,3:219,3:219,3:219,3:220,3:220,3:220,3:220,
3:222,3:222,3:223,3:223,3:223,3:223,3:223,3:223,3:223,
3:223,3:223,3:223,3:223,3:224,3:224,3:224,3:225,3:225,3:225,
3:225,3:226,3:226,3:226,3:226,3:226,3:226,3:226,3:226,
3:226,3:226,3:226,3:227,3:227,3:227,3:227,3:227,3:227,
3:227,3:227,3:227,3:227,3:228,3:228,3:229,3:229,3:229,
3:229,3:235,3:235,3:236,3:236,3:236,3:238,3:238,3:238,
3:238,3:239,3:239,3:239,3:241,3:241,3:241,3:242,3:243,
3:243,3:243,3:243,3:244,3:244,3:244,3:244,3:244,3:246,
3:246,3:246,3:246,3:246,3:247,3:248,3:248,3:249,3:249,
3:249,3:249,3:249,3:250,3:251,3:251,3:251,3:251,3:251,
3:252,3:252,3:252,3:252,3:253,3:253,3:255,3:255,3:255,
3:255,3:256,3:256,3:256,3:256,3:256,3:256,3:256,3:257,
3:258,3:258,3:259,3:259,3:259,3:261,3:261,3:261,3:265,

3:265,3:265,3:267,3:267,3:267,3:268,3:268,3:268,3:269,
3:269,3:269,3:269,3:269,3:269,3:269,3:269,3:269,3:269,
3:269,3:269,3:269,3:269,3:269,3:270,3:270,3:270,3:270,
3:271,3:271,3:271,3:272,3:272,3:272,3:272,3:272,3:273,
3:273,3:273,3:273,3:273,3:274,3:274,3:274,3:274,3:274,
3:274,3:274,3:274,3:275,3:275,3:275,3:276,3:276,3:278,
3:278,3:278,3:278,3:278,3:278,3:278,3:278,3:278,3:278,
3:278,3:278,3:279,3:279,3:282,3:283,3:283,3:283,3:283,
3:283,3:283,3:283,3:284,3:284,3:284,3:285,3:285,3:285,
3:285,3:285,3:285,3:286,3:286,3:286,3:286,3:286,3:286,
3:286,3:288,3:288,3:288,3:288,3:288,3:288,3:288,3:288,
3:289,3:289,3:289,3:289,3:289,3:289,3:289,3:289,3:290,
3:290,3:290,3:290,3:294,3:295,3:295,3:295,3:295,3:296,
3:296,3:296,3:296,3:296,3:297,3:297,3:297,3:297,3:297,
3:297,3:297,3:297,3:297,3:297,3:298,3:298,3:298,3:298,
3:298,3:298,3:298,3:298,3:298,3:298,3:298,3:298,3:298,
3:298,3:298,3:298,3:299,3:299,3:299,3:299,3:299,3:299,
3:299,3:299,3:301,3:301,3:303,3:303,3:304,3:304,3:304,
3:304,3:304,3:304,3:304,3:304,3:304,3:304,3:304,3:304,
3:304,3:304,3:305,3:308,3:308,3:308,3:308,3:308,3:308,
3:308,3:308,3:308,3:308,3:308,3:308,3:309,3:309,
3:309,3:309,3:309,3:309,3:309,3:309,3:309,3:309,3:310,
3:310,3:310,3:310,3:310,3:310,3:311,3:311,3:311,3:311,
3:311,3:311,3:311,3:311,3:311,3:311,3:311,3:312,
3:312,3:312,3:312,3:312,3:312,3:312,3:312,3:312,3:312,
3:312,3:312,3:312,3:312,3:312,3:312,3:312,3:312,3:313,
3:313,3:313,3:314,3:314,3:314,3:314,3:314,3:314,3:314,
3:314,3:314,3:314,3:315,3:315,3:315,3:315,3:315,3:315,
3:315,3:315,3:315,3:317,3:319,3:319,3:319,3:319,3:319,
3:319,3:319,3:320,3:321,3:321,3:321,3:322,3:322,3:322,
3:322,3:323,3:323,3:329,3:329,3:332,3:332,3:332,3:332,
3:332,3:332,3:333,3:336,3:336,3:336,3:338,3:338,3:338,
3:341,3:343,3:343,3:344,3:344,3:344,3:345,3:347,3:348,
3:350,3:352,3:353,3:354,3:354,3:356,3:356,3:356,3:359,
3:359,3:359,3:359,3:362,3:362,3:362,3:362,3:362,3:362,
3:363,3:363,3:363,3:363,3:363,3:363,3:363,3:363,3:363,
3:363,3:363,3:363,3:363,3:363,3:363,3:363,3:363,3:363,
3:363,3:363,3:363,3:363,3:364,3:365,3:365,3:365,3:366,
3:366,3:366,3:366,3:368,3:368,3:368,3:369,3:369,3:369,
3:369,3:370,3:370,3:370,3:370,3:370,3:370,3:370,3:371,
3:371,3:371,3:371,3:371,3:371,3:373,3:373,3:376,3:376,
3:376,3:376,3:376,3:376,3:376,3:376,3:376,3:376,3:377,
3:381,3:381,3:381,3:382,3:382,3:382,3:383,3:383,3:384,
3:384,3:384,3:385,3:386,3:386,3:387,3:387,3:387,3:387,
3:387,3:387,3:387,3:387,3:387,3:388,3:388,3:388,3:388,
3:389,3:389,3:389,3:390,3:390,3:390,3:390,3:390,3:390,
3:391,3:391,3:391,3:391,3:391,3:391,3:392,3:392,3:392,
3:392,3:392,3:392,3:397,3:397,3:397,3:398,3:398,3:398,
3:398,3:398,3:398,3:401,3:401,3:402,3:402,3:402,3:402,
3:402,3:402,3:402,3:402,3:402,3:403,3:403,3:404,3:404,
3:404,3:404,3:404,3:404,3:404,3:404,3:404,3:404,3:404,
3:405,3:405,3:405,3:407,3:407,3:407,3:410,3:410,3:410,
3:410,3:410,3:410,3:411,3:412,3:412,3:412,3:412,3:412,
3:412,3:412,3:412,3:412,3:412,3:412,3:412,3:412,3:412,
3:412,3:412,3:412,3:413,3:413,3:413,3:419,3:420,3:420,
3:420,3:420,3:420,3:420,3:420,3:420,3:420,3:420,3:421,
3:422,3:422,3:422,3:422,3:423,3:423,3:423,3:423,3:423,
3:423,3:423,3:423,3:423,3:424,3:424,3:424,3:429,3:429,

R3

3:431,3:437,3:437,3:437,3:437,3:437,3:437,3:437,3:438,3:438,
 3:438,3:439,3:439,3:439,3:439,3:439,3:439,3:441,3:441,
 3:441,3:441,3:442,3:442,3:442,3:444,3:444,3:444
 - 1:30,1:30,1:31,1:32,1:32,1:32,1:32,1:34,1:34,1:36,
 1:36,1:36,1:37,1:37,1:37,1:37,1:39,1:39,1:39,1:39,
 1:39,1:39,1:39,1:39,1:40,1:40,1:40,1:40,1:41,1:41,
 1:41,1:41,1:41,1:42,1:42,1:42,1:42,1:43,1:43,1:43,
 1:43,1:43,1:43,1:44,1:44,1:44,1:44,1:44,1:46,1:46,
 1:48,1:48,1:49,1:49,1:49,1:49,1:49,1:49,1:49,1:50,
 1:51,1:51,1:52,1:52,1:52,1:54,1:54,1:54,1:54,1:54,
 1:55,1:55,1:55,1:56,1:56,1:56,1:57,1:57,1:57,1:57,
 1:57,1:57,1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,
 1:63,1:63,1:63,1:64,1:64,1:64,1:64,1:64,1:64,1:64,
 1:65,1:65,1:65,1:66,1:66,1:66,1:66,1:73,1:74,1:74,
 1:74,1:74,1:75,1:75,1:77,1:77,1:77,1:77,1:78,1:78,
 1:78,1:78,1:79,1:79,1:79,1:80,1:80,1:80,1:80,1:80,
 1:80,1:80,1:80,1:80,1:81,1:82,1:82,1:82,1:82,1:82,
 1:82,1:82,1:84,1:84,1:85,1:85,1:85,1:85,1:85,1:85,
 1:85,1:85,1:85,1:85,1:85,1:85,1:86,1:86,1:86,1:86,
 1:86,1:86,1:86,1:86,1:86,1:86,1:87,1:87,1:87,1:87,
 1:87,1:87,1:87,1:87,1:88,1:88,1:88,1:88,1:88,1:88,
 1:88,1:88,1:89,1:89,1:89,1:89,1:89,1:89,1:90,1:90,
 1:90,1:90,1:90,1:90,1:90,1:90,1:90,1:90,1:90,1:90,
 1:91,1:91,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,
 1:92,1:92,1:92,1:93,1:93,1:93,1:93,1:93,1:94,1:94,
 1:95,1:95,1:95,1:100,1:100,1:101,1:101,1:103,1:103,
 1:103,1:104,1:104,1:104,1:104,1:105,1:106,1:106,1:106,
 1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:106,1:107,
 1:107,1:107,1:107,1:107,1:107,1:107,1:107,1:107,1:108,
 1:108,1:110,1:110,1:110,1:111,1:111,1:111,1:111,1:111,
 1:111,1:112,1:112,1:112,1:112,1:112,1:112,1:112,1:112,
 1:112,1:112,1:112,1:114,1:114,1:114,1:114,1:115,1:115,1:116,
 1:116,1:116,2:15,2:15,2:18,2:18,2:18,2:18,2:38,2:38,
 2:38,2:39,2:40,2:40,2:40,2:40,2:41,2:43,2:56,
 2:56,2:58,2:58,2:58,2:60,2:60,2:60,2:60,2:61,2:62,
 2:62,2:68,2:68,2:98,2:98,2:104,2:104,2:104,2:104,2:106,
 2:107,2:107,2:110,2:110,2:110,2:110,2:110,2:110,
 2:110,2:110,2:110,2:115,2:115,2:115,2:115,2:126,2:126,
 2:126,2:127,2:129,2:129,2:129,2:130,2:131,2:131,2:131,
 2:131,2:131,2:131,2:131,2:131,2:131,2:134,2:134,2:134,
 2:134,2:135,3:49,3:49,3:49,3:49,3:52,3:52,3:52,3:52,3:52,
 3:53,3:53,3:53,3:53,3:56,3:56,3:57,3:57,3:57,3:58,
 3:58,3:58,3:59,3:59,3:59,3:59,3:59,3:59,3:60,3:60,
 3:60,3:60,3:60,3:60,3:64,3:64,3:66,3:66,3:66,3:66,
 3:66,3:69,3:69,3:69,3:69,3:69,3:69,3:69,3:71,3:72,
 3:72,3:72,3:72,3:72,3:72,3:79,3:79,3:83,3:83,3:83,
 3:83,3:83,3:83,3:83,3:83,3:86,3:86,3:86,3:86,3:86,
 3:86,3:86,3:86,3:86,3:86,3:87,3:87,3:87,3:87,3:87,
 3:87,3:87,3:87,3:88,3:88,3:88,3:88,3:88,3:89,
 3:89,3:89,3:89,3:89,3:89,3:90,3:90,3:90,3:90,3:90,
 3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:91,3:91,
 3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,
 3:91,3:91,3:96,3:96,3:100,3:100,3:100,3:100,3:100,
 3:100,3:100,3:100,3:100,3:100,3:100,3:103,3:103,3:104,
 3:104,3:104,3:104,3:104,3:104,3:105,3:105,3:105,
 3:105,3:105,3:107,3:107,3:107,3:107,3:108,3:108,3:108,
 3:108,3:108,3:108,3:108,3:109,3:109,3:109,3:109,3:109,
 3:109,3:109,3:109,3:112,3:112,3:112,3:112,

3:115,3:115,3:117,3:118,3:118,3:123,3:123,3:123,3:123,3:123,
3:123,3:123,3:123,3:127,3:127,3:127,3:127,3:127,3:131,
3:131,3:131,3:132,3:132,3:132,3:132,3:132,3:132,3:132,3:132,
3:132,3:132,3:133,3:133,3:133,3:133,3:133,3:133,3:133,3:134,
3:134,3:134,3:134,3:135,3:135,3:135,3:135,3:141,3:141,3:141,
3:144,3:144,3:144,3:144,3:144,3:144,3:144,3:144,3:144,3:144,
3:145,3:145,3:145,3:145,3:146,3:146,3:146,3:147,3:148,3:148,
3:148,3:150,3:150,3:150,3:150,3:150,3:150,3:151,3:151,
3:151,3:151,3:151,3:151,3:151,3:151,3:152,3:152,3:152,
3:152,3:152,3:152,3:152,3:154,3:154,3:154,3:154,3:155,3:155,
3:155,3:157,3:158,3:158,3:159,3:159,3:159,3:160,3:161,
3:161,3:161,3:161,3:161,3:161,3:161,3:162,3:162,3:162,
3:163,3:163,3:163,3:163,3:163,3:163,3:163,3:164,3:164,
3:164,3:164,3:164,3:164,3:164,3:164,3:164,3:166,3:166,
3:167,3:167,3:167,3:167,3:167,3:167,3:167,3:167,3:167,
3:168,3:168,3:169,3:169,3:169,3:169,3:169,3:170,3:171,
3:171,3:177,3:178,3:178,3:178,3:186,3:186,3:187,3:187,
3:188,3:188,3:188,3:188,3:188,3:188,3:189,3:189,3:189,
3:189,3:189,3:189,3:190,3:192,3:192,3:192,3:192,3:192,
3:192,3:194,3:195,3:195,3:195,3:195,3:195,3:195,3:195,
3:197,3:197,3:197,3:198,3:198,3:198,3:198,3:198,3:200,
3:200,3:200,3:200,3:200,3:201,3:201,3:203,3:203,3:209,
3:209,3:209,3:209,3:209,3:209,3:209,3:209,3:209,3:209,
3:209,3:209,3:209,3:210,3:210,3:210,3:211,3:211,3:211,
3:211,3:211,3:212,3:212,3:212,3:213,3:214,3:214,3:214,
3:214,3:215,3:215,3:215,3:215,3:215,3:217,3:217,3:217,
3:217,3:217,3:217,3:217,3:217,3:217,3:218,3:218,3:219,
3:219,3:219,3:220,3:220,3:220,3:220,3:220,3:220,3:220,
3:220,3:220,3:220,3:220,3:220,3:223,3:223,3:223,3:223,
3:223,3:223,3:224,3:224,3:224,3:224,3:225,3:225,3:225,
3:225,3:225,3:225,3:226,3:226,3:226,3:226,3:226,3:226,
3:226,3:226,3:226,3:226,3:226,3:226,3:226,3:226,3:226,
3:226,3:227,3:227,3:227,3:227,3:227,3:227,3:227,3:227,
3:228,3:228,3:229,3:229,3:229,3:234,3:234,3:234,3:234,3:234,
3:234,3:234,3:235,3:235,3:235,3:235,3:236,3:236,3:237,3:237,
3:237,3:237,3:238,3:238,3:238,3:238,3:238,3:238,3:239,
3:239,3:239,3:239,3:239,3:239,3:241,3:242,3:242,3:242,
3:242,3:242,3:242,3:242,3:243,3:243,3:244,3:244,3:244,
3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,3:244,
3:244,3:244,3:244,3:246,3:246,3:246,3:246,3:246,3:247,
3:248,3:248,3:248,3:248,3:248,3:248,3:248,3:248,3:249,
3:249,3:249,3:250,3:251,3:251,3:251,3:252,3:252,3:252,
3:252,3:253,3:253,3:253,3:254,3:255,3:255,3:255,3:255,
3:255,3:256,3:256,3:256,3:256,3:257,3:259,3:259,3:259,3:259,
3:259,3:260,3:260,3:261,3:261,3:263,3:263,3:263,3:263,3:263,
3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:264,
3:266,3:266,3:266,3:266,3:266,3:266,3:266,3:267,3:267,
3:268,3:268,3:269,3:269,3:269,3:270,3:270,3:270,3:270,
3:270,3:270,3:270,3:271,3:271,3:271,3:271,3:273,3:273,
3:273,3:273,3:273,3:274,3:274,3:274,3:274,3:274,3:274,
3:274,3:275,3:275,3:275,3:276,3:276,3:276,3:277,3:277,
3:278,3:278,3:278,3:278,3:278,3:278,3:278,3:278,3:278,
3:278,3:278,3:278,3:279,3:279,3:279,3:280,3:280,3:280,
3:280,3:281,3:281,3:281,3:281,3:281,3:281,3:281,3:282,3:282,
3:282,3:282,3:282,3:283,3:283,3:283,3:283,3:283,3:283,
3:283,3:284,3:284,3:284,3:285,3:285,3:285,3:286,3:286,
3:286,3:286,3:286,3:286,3:286,3:287,3:287,3:287,3:287,

3:287,3:288,3:288,3:288,3:288,3:288,3:288,3:288,3:288,
3:288,3:288,3:289,3:289,3:289,3:289,3:290,3:290,3:290,
3:290,3:290,3:290,3:290,3:292,3:292,3:292,3:292,3:292,
3:292,3:293,3:293,3:293,3:293,3:293,3:293,3:293,3:294,
3:294,3:294,3:294,3:294,3:294,3:294,3:294,3:294,3:294,
3:294,3:294,3:294,3:295,3:295,3:295,3:295,3:295,3:295,
3:295,3:295,3:295,3:296,3:296,3:296,3:296,3:296,3:296,
3:296,3:296,3:297,3:297,3:297,3:297,3:298,3:298,3:298,
3:298,3:299,3:299,3:299,3:299,3:299,3:299,3:299,3:299,
3:299,3:299,3:299,3:299,3:299,3:299,3:300,3:300,3:300,
3:300,3:300,3:300,3:301,3:301,3:301,3:301,3:302,3:302,
3:302,3:302,3:302,3:302,3:302,3:302,3:302,3:302,3:302,
3:303,3:303,3:303,3:303,3:303,3:303,3:303,3:303,3:303,
3:303,3:303,3:303,3:303,3:303,3:303,3:303,3:303,3:303,
3:303,3:303,3:303,3:304,3:304,3:304,3:305,3:305,
3:305,3:305,3:305,3:305,3:305,3:305,3:305,3:305,3:305,
3:305,3:306,3:306,3:306,3:306,3:307,3:307,3:307,3:308,
3:308,3:308,3:308,3:308,3:308,3:308,3:308,3:309,3:309,
3:309,3:311,3:313,3:313,3:313,3:313,3:313,3:313,3:314,
3:314,3:314,3:314,3:314,3:314,3:314,3:314,3:314,3:314,
3:314,3:314,3:314,3:315,3:315,3:315,3:315,3:316,3:316,
3:316,3:316,3:316,3:316,3:316,3:318,3:318,3:318,3:318,
3:319,3:319,3:319,3:319,3:319,3:319,3:319,3:319,3:322,
3:322,3:322,3:322,3:322,3:322,3:323,3:323,3:323,3:323,
3:323,3:323,3:323,3:324,3:324,3:324,3:324,3:324,3:324,
3:324,3:329,3:329,3:329,3:332,3:332,3:332,3:332,3:332,
3:332,3:336,3:336,3:336,3:351,3:351,3:351,3:351,3:351,
3:351,3:351,3:351,3:351,3:351,3:351,3:351,3:351,3:351,
3:351,3:351,3:351,3:351,3:351,3:352,3:352,3:352,3:352,
3:352,3:352,3:352,3:352,3:352,3:352,3:353,3:354,3:354,
3:354,3:354,3:354,3:355,3:355,3:355,3:355,3:355,3:355,
3:355,3:355,3:355,3:355,3:356,3:357,3:357,3:362,3:362,
3:362,3:362,3:362,3:362,3:362,3:363,3:363,3:363,3:363,
3:363,3:363,3:363,3:363,3:363,3:365,3:366,3:368,3:370,
3:370,3:370,3:371,3:371,3:371,3:371,3:371,3:371,3:371,
3:371,3:372,3:372,3:373,3:373,3:373,3:373,3:373,3:373,
3:373,3:373,3:375,3:375,3:375,3:375,3:375,3:375,3:375,
3:375,3:376,3:376,3:376,3:381,3:381,3:381,3:381,3:381,
3:381,3:383,3:383,3:383,3:383,3:383,3:384,3:384,3:384,
3:385,3:385,3:385,3:385,3:385,3:385,3:386,3:386,3:386,
3:386,3:386,3:386,3:386,3:387,3:387,3:387,3:387,3:387,
3:387,3:387,3:387,3:387,3:388,3:388,3:388,3:388,3:388,
3:388,3:388,3:388,3:388,3:388,3:388,3:388,3:389,3:389,
3:389,3:389,3:389,3:389,3:389,3:389,3:390,3:391,3:391,
3:391,3:391,3:393,3:393,3:393,3:393,3:393,3:393,3:393,
3:394,3:394,3:394,3:394,3:394,3:394,3:394,3:395,3:395,
3:395,3:395,3:395,3:395,3:395,3:395,3:395,3:396,3:396,
3:396,3:396,3:396,3:396,3:396,3:396,3:396,3:396,3:396,
3:398,3:398,3:399,3:399,3:399,3:399,3:399,3:400,3:400,
3:400,3:400,3:400,3:400,3:400,3:400,3:401,3:401,3:401,
3:401,3:401,3:401,3:401,3:402,3:402,3:402,3:402,3:402,
3:402,3:404,3:404,3:404,3:404,3:404,3:404,3:404,3:404,
3:404,3:404,3:404,3:404,3:404,3:404,3:404,3:404,3:405,
3:405,3:405,3:405,3:405,3:410,3:410,3:410,3:410,3:410,
3:410,3:410,3:410,3:411,3:411,3:411,3:411,3:411,3:411,
3:411,3:411,3:412,3:412,3:412,3:412,3:413,3:413,3:413,
3:413,3:413,3:420,3:420,3:421,3:421,3:421,3:421,3:421,
3:421,3:421,3:421,3:421,3:421,3:422,3:422,

R4 -

1:13,1:13,1:13,1:30,1:30,1:31,1:31,1:31,1:31,1:31,1:31,
1:31,1:31,1:32,1:33,1:33,1:33,1:34,1:34,1:34,1:34,
1:35,1:35,1:35,1:36,1:36,1:36,1:36,1:36,1:36,
1:37,1:38,1:38,1:38,1:38,1:38,1:38,1:38,1:38,1:38,
1:38,1:38,1:39,1:39,1:39,1:39,1:39,1:39,1:39,1:39,
1:39,1:39,1:39,1:39,1:39,1:40,1:41,1:41,1:41,1:41,1:41,
1:42,1:44,1:44,1:44,1:44,1:47,1:47,1:47,1:47,1:47,
1:47,1:48,1:48,1:48,1:48,1:48,1:48,1:49,1:49,
1:49,1:49,1:49,1:49,1:49,1:49,1:49,1:49,1:50,1:51,
1:53,1:62,1:62,1:62,1:62,1:63,1:63,1:63,1:63,
1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,
1:63,1:63,1:63,1:63,1:63,1:64,1:64,1:64,1:64,1:64,
1:64,1:64,1:64,1:65,1:65,1:65,1:66,1:66,1:66,
1:73,1:74,1:74,1:75,1:78,1:78,1:78,1:78,1:78,
1:78,1:78,1:78,1:78,1:79,1:79,1:79,1:79,1:79,
1:80,1:80,1:80,1:80,1:81,1:81,1:81,1:81,1:82,1:82,
1:82,1:82,1:82,1:82,1:82,1:82,1:84,1:84,1:84,1:84,
1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,1:84,
1:84,1:85,1:85,1:85,1:87,1:87,1:87,1:88,1:88,1:88,
1:90,1:93,1:94,1:95,1:106,1:106,1:106,1:106,1:106,
1:107,1:107,1:108,1:108,1:109,1:109,1:109,1:109,1:109,
1:109,1:109,1:110,1:112,1:112,1:114,1:114,1:114,2:38,
2:38,2:39,2:40,2:40,2:40,2:40,2:40,2:41,2:43,2:45,
2:52,2:52,2:55,2:56,2:56,2:56,2:58,2:58,2:58,2:58,
2:58,2:58,2:58,2:58,2:58,2:61,2:61,2:61,2:61,2:61,
2:61,2:61,2:61,2:63,2:63,2:70,2:85,2:85,2:85,2:87,
2:87,2:87,2:87,2:87,2:88,2:88,2:89,2:89,2:89,2:89,
2:98,2:107,2:110,2:110,2:112,2:112,2:112,2:121,2:121,
2:121,2:122,2:125,2:126,2:126,2:127,2:127,2:127,2:127,
2:127,2:127,2:128,2:128,2:128,2:128,2:128,2:128,2:128,
2:128,2:128,3:55,3:55,3:55,3:59,3:59,3:64,3:64,3:64,
3:66,3:66,3:66,3:66,3:66,3:66,3:72,3:79,3:79,
3:87,3:87,3:87,3:87,3:87,3:87,3:88,3:88,3:88,3:88,
3:88,3:88,3:88,3:88,3:88,3:88,3:88,3:89,3:89,3:89,
3:89,3:89,3:89,3:89,3:89,3:89,3:89,3:89,3:89,3:89,
3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,
3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,3:90,
3:90,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,
3:91,3:95,3:95,3:96,3:96,3:96,3:97,3:99,3:100,3:100,
3:100,3:100,3:100,3:100,3:101,3:101,3:102,3:102,3:102,
3:102,3:102,3:102,3:102,3:103,3:103,3:103,3:103,3:103,
3:104,3:105,3:105,3:105,3:105,3:105,3:105,3:105,3:105,
3:105,3:105,3:105,3:105,3:105,3:105,3:105,3:105,3:105,
3:105,3:106,3:107,3:107,3:107,3:107,3:107,3:107,3:107,
3:107,3:107,3:107,3:108,3:108,3:108,3:108,3:108,3:108,
3:108,3:109,3:109,3:109,3:109,3:109,3:109,3:109,3:109,
3:109,3:109,3:109,3:110,3:110,3:110,3:110,3:110,3:110,
3:111,3:111,3:111,3:112,3:112,3:112,3:112,3:112,3:112,
3:112,3:112,3:112,3:112,3:112,3:112,3:112,3:113,3:113,
3:113,3:113,3:113,3:113,3:113,3:113,3:113,3:115,3:115,
3:115,3:115,3:117,3:117,3:118,3:120,3:120,3:120,3:120,

3:121,3:121,3:121,3:121,3:121,3:121,3:121,3:122,3:122,
3:122,3:122,3:123,3:123,3:123,3:123,3:123,3:124,3:124,
3:124,3:124,3:124,3:131,3:131,3:131,3:131,3:131,3:131,
3:131,3:131,3:131,3:131,3:131,3:131,3:131,3:131,3:131,
3:132,3:132,3:132,3:132,3:133,3:133,3:133,3:133,3:133,
3:133,3:133,3:133,3:133,3:133,3:133,3:133,3:133,3:133,
3:133,3:134,3:134,3:134,3:134,3:134,3:134,3:134,3:135,
3:135,3:135,3:135,3:135,3:135,3:135,3:135,3:135,3:136,
3:136,3:136,3:136,3:136,3:136,3:136,3:136,3:137,3:137,
3:137,3:137,3:138,3:138,3:138,3:138,3:138,3:138,3:138,
3:138,3:138,3:138,3:138,3:138,3:138,3:138,3:138,3:139,
3:139,3:139,3:139,3:139,3:139,3:139,3:139,3:139,3:139,
3:141,3:141,3:141,3:141,3:141,3:141,3:141,3:141,3:141,
3:141,3:141,3:141,3:141,3:142,3:142,3:142,3:143,3:143,3:143,
3:143,3:143,3:143,3:144,3:144,3:144,3:144,3:144,3:144,
3:144,3:144,3:144,3:144,3:144,3:144,3:145,3:145,
3:146,3:146,3:146,3:146,3:146,3:146,3:147,3:147,3:147,
3:147,3:147,3:147,3:148,3:148,3:148,3:148,3:148,3:148,
3:148,3:148,3:148,3:148,3:148,3:148,3:148,3:150,3:150,
3:150,3:151,3:151,3:151,3:151,3:151,3:152,3:152,3:152,
3:152,3:152,3:152,3:152,3:152,3:152,3:153,3:153,
3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:153,
3:154,3:154,3:154,3:154,3:154,3:154,3:154,3:154,3:154,
3:154,3:155,3:155,3:155,3:155,3:155,3:155,3:155,3:155,
3:156,3:156,3:156,3:156,3:156,3:156,3:157,3:157,3:157,
3:158,3:158,3:158,3:158,3:159,3:159,3:160,3:161,3:161,
3:161,3:161,3:161,3:161,3:161,3:162,3:162,3:162,3:162,
3:163,3:163,3:164,3:164,3:164,3:164,3:167,3:167,3:167,
3:167,3:167,3:167,3:167,3:167,3:167,3:167,3:168,3:168,
3:168,3:168,3:168,3:168,3:169,3:169,3:169,3:169,3:169,
3:169,3:170,3:171,3:171,3:171,3:171,3:175,3:175,3:175,
3:175,3:175,3:175,3:175,3:177,3:177,3:177,3:177,3:177,
3:177,3:177,3:177,3:177,3:178,3:178,3:178,3:178,3:178,
3:178,3:179,3:179,3:179,3:179,3:179,3:179,3:179,3:179,
3:179,3:187,3:187,3:187,3:187,3:187,3:187,3:187,3:187,
3:187,3:188,3:188,3:189,3:189,3:189,3:191,3:192,3:192,
3:192,3:192,3:192,3:192,3:194,3:194,3:194,3:194,3:194,
3:194,3:194,3:194,3:195,3:195,3:195,3:195,3:195,3:195,
3:195,3:195,3:197,3:197,3:197,3:197,3:197,3:197,3:197,
3:197,3:197,3:198,3:198,3:198,3:198,3:199,3:199,3:200,
3:200,3:201,3:201,3:201,3:201,3:201,3:201,3:202,3:202,
3:202,3:202,3:202,3:203,3:204,3:204,3:204,3:204,3:204,
3:204,3:204,3:204,3:205,3:205,3:205,3:205,3:206,3:206,
3:206,3:206,3:206,3:206,3:206,3:207,3:207,3:208,3:208,
3:208,3:208,3:208,3:208,3:208,3:208,3:208,3:208,3:208,
3:208,3:210,3:210,3:211,3:211,3:211,3:211,3:211,3:212,
3:212,3:212,3:214,3:214,3:214,3:214,3:219,3:219,3:219,
3:220,3:220,3:220,3:220,3:221,3:221,3:221,3:221,3:221,
3:221,3:221,3:221,3:222,3:222,3:222,3:222,3:222,3:222,
3:223,3:223,3:223,3:223,3:223,3:223,3:224,3:224,3:224,
3:225,3:225,3:225,3:225,3:225,3:226,3:226,3:227,3:227,
3:227,3:227,3:227,3:227,3:227,3:228,3:228,3:228,3:228,
3:229,3:229,3:232,3:234,3:234,3:234,3:234,3:234,3:234,
3:234,3:234,3:234,3:234,3:235,3:235,3:235,3:235,3:235,
3:236,3:236,3:236,3:236,3:236,3:237,3:237,3:237,3:237,
3:237,3:237,3:237,3:238,3:238,3:238,3:238,3:238,3:238,
3:238,3:239,3:239,3:241,3:241,3:241,3:241,3:241,3:242,
3:242,3:242,3:242,3:242,3:242,3:242,3:243,3:244,

3:331,3:332,3:332,3:333,3:334,3:334,3:334,3:334,3:334,
 3:334,3:334,3:335,3:335,3:335,3:335,3:335,3:335,3:335,
 3:336,3:336,3:336,3:336,3:337,3:337,3:337,3:337,3:337,
 3:337,3:337,3:337,3:338,3:338,3:338,3:338,3:345,3:345,
 3:346,3:346,3:349,3:349,3:349,3:349,3:351,3:353,3:354,
 3:355,3:357,3:357,3:363,3:363,3:363,3:364,3:365,
 3:366,3:367,3:368,3:377,3:377,3:377,3:377,3:377,
 3:377,3:377,3:378,3:378,3:378,3:382,3:382,3:382,
 3:382,3:382,3:382,3:382,3:382,3:382,3:383,3:383,
 3:383,3:383,3:383,3:383,3:383,3:384,3:384,3:384,
 3:384,3:384,3:384,3:385,3:385,3:386,3:386,
 3:386,3:386,3:386,3:390,3:390,3:391,3:391,3:391,3:393,
 3:393,3:393,3:393,3:393,3:393,3:394,3:394,3:394,
 3:394,3:394,3:394,3:396,3:396,3:396,3:396,3:398,3:398,
 3:399,3:399,3:399,3:399,3:399,3:400,3:400,3:400,3:400,
 3:400,3:400,3:402,3:402,3:402,3:402,3:402,3:402,
 3:403,3:403,3:403,3:403,3:403,3:405,3:406,3:406,3:406,
 3:406,3:406,3:406,3:406,3:406,3:406,3:407,3:408,
 3:410,3:410,3:410,3:410,3:410,3:410,3:410,3:410,3:410,
 3:410,3:410,3:410,3:411,3:411,3:419,3:422,3:422,3:422,
 3:424,3:424,3:424,3:425,3:425,3:425,3:425,3:427,
 3:427,3:427,3:431,3:434,3:434,3:439,3:439,3:439,
 3:439,3:439,3:439,3:440,3:442,3:442,3:442,3:442
 1:30,1:30,1:30,1:30,1:30,1:32,1:32,1:32,1:32,
 1:32,1:34,1:34,1:34,1:34,1:34,1:34,1:34,1:47,1:47,
 1:47,1:47,1:47,1:48,1:48,1:48,1:48,1:50,1:50,1:51,
 1:85,1:85,1:85,1:85,1:87,1:88,1:88,1:88,1:88,1:88,
 1:89,1:89,1:89,1:89,1:90,1:90,1:90,1:90,1:90,1:90,
 1:90,1:91,1:91,1:91,1:91,1:91,1:91,1:91,1:91,
 1:92,1:93,1:93,1:93,1:93,1:93,1:94,1:94,1:94,1:95,
 1:95,1:95,1:95,1:95,2:126,2:126,2:126,2:126,2:126,
 2:128,3:87,3:87,3:87,3:87,3:87,3:87,3:87,3:87,3:90,
 3:90,3:90,3:90,3:90,3:90,3:90,3:91,3:91,3:91,3:91,
 3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,
 3:91,3:91,3:91,3:233,3:239,3:239,3:239,3:239,3:247,
 3:247,3:247,3:248,3:251,3:253,3:253,3:254,3:255,3:260,
 3:260,3:260,3:260,3:260,3:260,3:261,3:261,3:262,3:263,
 3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,
 3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,
 3:264,3:264,3:264,3:264,3:264,3:264,3:264,3:264,3:264,
 3:269,3:269,3:269,3:269,3:270,3:271,3:271,3:271,3:271,
 3:271,3:271,3:271,3:271,3:271,3:271,3:271,3:271,3:271,
 3:271,3:271,3:272,3:272,3:272,3:272,3:272,3:273,
 3:273,3:274,3:274,3:274,3:275,3:277,3:278,3:278,3:278,
 3:278,3:278,3:278,3:278,3:278,3:278,3:279,3:279,
 3:279,3:279,3:279,3:280,3:280,3:281,3:281,3:282,3:282,
 3:282,3:282,3:282,3:283,3:283,3:283,3:285,3:285,3:285,
 3:285,3:285,3:285,3:285,3:285,3:285,3:285,3:285,3:286,
 3:286,3:286,3:287,3:287,3:289,3:289,3:289,3:289,3:293,
 3:293,3:293,3:293,3:293,3:294,3:294,3:295,3:295,3:295,
 3:295,3:295,3:299,3:299,3:299,3:300,3:300,3:300,3:300,
 3:300,3:300,3:302,3:302,3:302,3:302,3:302,3:302,3:302,
 3:302,3:302,3:302,3:302,3:302,3:302,3:302,3:302,3:302,
 3:303,3:303,3:303,3:303,3:303,3:303,3:303,3:304,3:304,
 3:304,3:304,3:304,3:304,3:304,3:305,3:307,3:307,3:307,
 3:307,3:307,3:307,3:307,3:308,3:308,3:309,3:309,3:310,
 3:311,3:311,3:311,3:311,3:311,3:311,3:311,3:311,

R7

-
3:311,3:313,3:313,3:313,3:313,3:313,3:315,3:315,3:315,
3:315,3:315,3:315,3:315,3:315,3:324,3:382,3:382,3:382,
3:382,3:382,3:382,3:382,3:382,3:382,3:395,3:395,
3:395,3:395
1:30,1:30,1:30,1:30,1:30,1:30,1:31,1:31,1:31,
1:31,1:31,1:31,1:31,1:31,1:31,1:34,1:34,1:34,
1:34,1:38,1:38,1:39,1:39,1:39,1:40,1:40,1:41,1:41,
1:42,1:42,1:42,1:42,1:42,1:42,1:42,1:42,1:42,
1:43,1:43,1:43,1:43,1:44,1:44,1:44,1:44,1:45,1:45,
1:45,1:45,1:45,1:45,1:45,1:46,1:46,1:46,1:46,
1:46,1:46,1:47,1:47,1:47,1:48,1:49,1:49,1:51,1:54,
1:54,1:54,1:54,1:54,1:54,1:54,1:54,1:55,1:55,
1:55,1:55,1:55,1:55,1:55,1:56,1:56,1:56,1:57,1:57,
1:57,1:57,1:57,1:58,1:58,1:58,1:58,1:58,1:58,
1:59,1:59,1:59,1:59,1:59,1:59,1:60,1:60,1:60,
1:60,1:60,1:61,1:61,1:61,1:61,1:62,1:62,1:62,1:63,
1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,1:63,
1:63,1:64,1:64,1:65,1:65,1:65,1:66,1:66,1:66,1:66,
1:66,1:67,1:67,1:67,1:67,1:67,1:67,1:73,1:73,1:73,
1:73,1:73,1:73,1:73,1:73,1:73,1:74,1:74,1:74,1:74,
1:74,1:74,1:77,1:77,1:77,1:78,1:78,1:78,1:78,1:78,
1:78,1:78,1:78,1:78,1:79,1:79,1:80,1:80,1:80,1:80,
1:81,1:81,1:81,1:81,1:81,1:81,1:82,1:82,1:82,1:82,
1:82,1:82,1:84,1:84,1:84,1:84,1:84,1:84,1:84,
1:85,1:85,1:85,1:85,1:85,1:85,1:85,1:88,1:88,
1:88,1:90,1:90,1:90,1:91,1:91,1:91,1:91,1:92,1:92,
1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,1:92,
1:92,1:92,1:92,1:92,1:93,1:93,1:93,1:93,1:93,1:93,
1:93,1:93,1:94,1:94,1:94,1:94,1:94,1:94,1:94,1:94,
1:94,1:95,1:95,1:95,1:95,1:100,1:100,1:100,1:100,1:102,
1:102,1:102,1:102,1:102,1:102,1:102,1:102,1:102,
1:102,1:103,1:103,1:103,1:103,1:103,1:103,1:103,
1:104,1:104,1:104,1:104,1:104,1:104,1:104,1:104,
1:104,1:104,1:104,1:104,1:104,1:104,1:104,1:104,
1:105,1:105,1:106,1:106,1:106,1:107,1:107,1:107,1:107,
1:107,1:107,1:107,1:107,1:107,1:108,1:108,1:108,
1:108,1:108,1:108,1:108,1:108,1:108,1:109,1:109,
1:109,1:109,1:110,1:110,1:110,1:110,1:110,1:110,1:110,
1:110,1:110,1:110,1:111,1:111,1:111,1:111,1:112,
1:112,1:112,1:112,1:112,1:112,1:112,1:112,1:112,1:112,
1:112,1:112,1:114,1:114,1:114,1:114,1:114,1:114,1:114,
1:114,1:114,1:114,1:114,1:114,1:114,1:114,1:114,1:114,
1:115,1:115,1:115,1:115,1:115,1:115,1:116,1:116,1:116,
1:116,1:116,1:116,1:116,1:116,1:116,1:116,1:116,1:116,
1:116,1:116,1:116,1:116,1:116,1:116,1:117,1:117,1:117,
1:117,1:117,1:117,1:117,1:117,2:16,2:16,2:21,2:28,
2:28,2:33,2:39,2:39,2:39,2:41,2:41,2:41,2:41,
2:41,2:42,2:42,2:42,2:42,2:42,2:42,2:43,2:43,
2:43,2:43,2:43,2:43,2:44,2:44,2:44,2:44,2:45,2:45,
2:45,2:47,2:47,2:49,2:49,2:49,2:50,2:50,2:50,2:50,
2:50,2:50,2:51,2:51,2:51,2:51,2:52,2:52,2:52,2:53,
2:53,2:54,2:54,2:55,2:55,2:55,2:55,2:56,2:56,2:58,
2:58,2:58,2:58,2:58,2:58,2:58,2:59,2:60,2:64,2:64,
2:64,2:64,2:64,2:65,2:65,2:66,2:66,2:66,2:66,
2:66,2:66,2:66,2:68,2:68,2:68,2:68,2:68,2:68,2:68,
2:68,2:68,2:68,2:68,2:68,2:81,2:81,2:82,2:82,2:82,
2:82,2:84,2:84,2:84,2:84,2:85,2:85,2:85,2:87,2:87,
2:87,2:88,2:89,2:89,2:89,2:89,2:89,2:90,2:90,

2:90,2:90,2:90,2:90,2:90,2:91,2:91,2:91,2:91,
2:91,2:91,2:91,2:91,2:93,2:93,2:93,2:93,2:93,
2:93,2:94,2:94,2:94,2:94,2:96,2:96,2:98,2:98,2:98,
2:98,2:98,2:98,2:104,2:105,2:105,2:105,2:105,2:105,
2:105,2:105,2:105,2:106,2:106,2:106,2:106,2:110,2:110,
2:112,2:112,2:112,2:113,2:115,2:115,2:115,2:115,2:115,
2:115,2:115,2:115,2:118,2:120,2:120,2:129,2:129,2:129,
2:129,2:129,2:129,2:131,2:133,2:133,2:133,2:134,2:134,
2:134,2:134,2:136,2:136,3:49,3:49,3:49,3:49,3:49,3:49,
3:49,3:49,3:50,3:50,3:50,3:50,3:50,3:50,3:51,
3:51,3:51,3:51,3:51,3:51,3:51,3:51,3:52,3:52,
3:52,3:52,3:52,3:53,3:53,3:53,3:53,3:53,3:53,3:53,
3:53,3:53,3:53,3:54,3:54,3:54,3:54,3:54,3:54,3:54,
3:55,3:55,3:55,3:56,3:56,3:57,3:57,3:57,3:57,3:59,
3:59,3:59,3:59,3:59,3:59,3:60,3:60,3:63,3:63,3:63,
3:65,3:65,3:65,3:65,3:65,3:65,3:65,3:65,3:66,
3:66,3:66,3:66,3:66,3:66,3:66,3:70,3:70,3:70,3:70,
3:70,3:71,3:71,3:77,3:77,3:77,3:81,3:81,3:81,3:81,
3:83,3:85,3:85,3:85,3:86,3:86,3:86,3:86,3:87,
3:88,3:89,3:89,3:89,3:89,3:89,3:89,3:89,3:89,3:90,
3:90,3:90,3:90,3:90,3:91,3:91,3:91,3:91,3:91,3:91,
3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,3:91,
3:91,3:91,3:91,3:93,3:93,3:93,3:93,3:93,3:93,3:94,
3:94,3:94,3:94,3:95,3:95,3:95,3:96,3:96,3:96,3:96,
3:96,3:96,3:96,3:96,3:96,3:97,3:97,3:97,3:97,3:99,
3:99,3:103,3:103,3:103,3:103,3:103,3:105,3:105,3:105,
3:105,3:105,3:105,3:105,3:105,3:105,3:107,3:107,3:108,
3:108,3:108,3:109,3:109,3:109,3:109,3:109,3:109,3:109,
3:110,3:110,3:110,3:110,3:110,3:111,3:111,3:111,3:111,
3:111,3:111,3:111,3:112,3:112,3:112,3:112,3:112,3:112,
3:112,3:112,3:114,3:114,3:115,3:115,3:118,3:118,3:118,3:118,
3:118,3:118,3:118,3:118,3:118,3:118,3:118,3:119,3:119,
3:119,3:119,3:120,3:120,3:120,3:120,3:120,3:120,3:121,
3:121,3:121,3:121,3:122,3:122,3:122,3:122,3:124,3:124,3:124,
3:124,3:124,3:124,3:124,3:124,3:127,3:127,3:128,3:128,
3:129,3:129,3:129,3:129,3:129,3:129,3:129,3:131,
3:131,3:131,3:131,3:131,3:132,3:132,3:132,3:132,3:132,
3:133,3:133,3:133,3:133,3:133,3:133,3:133,3:133,3:133,
3:133,3:133,3:134,3:134,3:134,3:134,3:135,3:135,3:135,
3:135,3:135,3:135,3:135,3:135,3:136,3:136,3:136,3:136,
3:136,3:136,3:137,3:137,3:137,3:137,3:137,3:137,3:137,
3:138,3:138,3:138,3:138,3:138,3:138,3:139,3:139,3:141,
3:141,3:141,3:141,3:142,3:142,3:144,3:144,3:144,3:144,
3:144,3:144,3:144,3:144,3:144,3:144,3:144,3:144,3:146,
3:146,3:146,3:146,3:146,3:146,3:146,3:147,3:147,3:147,
3:148,3:148,3:148,3:148,3:148,3:148,3:148,3:148,3:148,
3:148,3:148,3:148,3:148,3:148,3:149,3:149,3:150,3:150,
3:150,3:150,3:152,3:152,3:152,3:152,3:152,3:152,3:152,
3:152,3:152,3:152,3:152,3:152,3:153,3:153,3:153,3:153,
3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:153,3:154,
3:154,3:154,3:154,3:154,3:156,3:156,3:156,3:156,3:156,
3:156,3:157,3:158,3:158,3:158,3:158,3:158,3:159,3:159,
3:161,3:161,3:161,3:161,3:161,3:161,3:162,3:162,3:162,
3:162,3:162,3:164,3:164,3:164,3:166,3:166,3:166,3:166,3:166,
3:166,3:166,3:166,3:166,3:167,3:167,3:167,3:167,3:167,
3:167,3:167,3:167,3:167,3:167,3:167,3:167,3:167,3:167,
3:167,3:168,3:168,3:168,3:168,3:168,3:168,3:168,3:168,
3:168,3:168,3:169,3:169,3:171,3:171,3:171,3:171,

3:171,3:171,3:171,3:175,3:175,3:177,3:177,3:179,3:179,
3:179,3:179,3:179,3:179,3:179,3:179,3:179,3:179,
3:184,3:184,3:184,3:184,3:184,3:184,3:184,3:184,
3:187,3:187,3:187,3:187,3:187,3:190,3:190,3:190,3:192,
3:192,3:192,3:192,3:192,3:192,3:195,3:195,3:195,
3:195,3:195,3:195,3:195,3:195,3:195,3:195,3:195,
3:195,3:195,3:197,3:197,3:197,3:197,3:197,3:197,3:197,
3:197,3:197,3:197,3:197,3:197,3:197,3:197,3:197,
3:198,3:198,3:198,3:198,3:198,3:198,3:198,3:198,
3:198,3:198,3:199,3:199,3:199,3:199,3:200,3:200,3:200,
3:200,3:200,3:200,3:200,3:200,3:200,3:200,3:201,
3:201,3:201,3:201,3:201,3:201,3:201,3:201,3:201,
3:201,3:201,3:201,3:201,3:201,3:201,3:201,3:202,
3:202,3:202,3:202,3:203,3:203,3:203,3:204,3:204,3:204,
3:204,3:204,3:204,3:204,3:204,3:204,3:205,3:205,
3:205,3:205,3:205,3:205,3:205,3:205,3:206,3:206,3:206,
3:206,3:206,3:206,3:206,3:206,3:206,3:207,3:207,3:207,
3:207,3:207,3:207,3:207,3:207,3:208,3:208,3:208,3:208,
3:208,3:208,3:208,3:209,3:209,3:209,3:210,3:210,3:210,
3:210,3:210,3:210,3:210,3:211,3:211,3:211,3:211,
3:212,3:212,3:212,3:212,3:212,3:212,3:212,3:212,
3:212,3:213,3:216,3:216,3:217,3:217,3:217,3:217,3:217,
3:217,3:218,3:218,3:218,3:219,3:219,3:219,3:219,3:219,
3:219,3:219,3:219,3:219,3:219,3:220,3:220,3:220,3:220,
3:220,3:220,3:220,3:220,3:220,3:220,3:220,3:221,
3:221,3:221,3:221,3:221,3:221,3:222,3:222,3:223,
3:223,3:225,3:225,3:226,3:226,3:226,3:226,3:226,3:226,
3:226,3:226,3:226,3:226,3:226,3:226,3:229,3:229,
3:232,3:232,3:232,3:233,3:233,3:233,3:234,3:234,3:234,
3:234,3:234,3:234,3:234,3:235,3:235,3:235,3:235,3:236,
3:236,3:237,3:237,3:237,3:237,3:238,3:239,3:239,3:239,
3:239,3:239,3:239,3:241,3:241,3:241,3:242,3:242,3:242,
3:242,3:242,3:242,3:242,3:242,3:242,3:243,3:243,3:243,
3:244,3:244,3:244,3:244,3:246,3:246,3:246,3:247,3:247,
3:247,3:247,3:247,3:247,3:247,3:247,3:247,3:247,3:247,
3:247,3:247,3:247,3:247,3:248,3:248,3:248,3:248,3:248,
3:248,3:248,3:248,3:248,3:248,3:248,3:248,3:248,3:248,
3:248,3:249,3:249,3:249,3:249,3:249,3:249,3:250,3:250,
3:250,3:250,3:250,3:250,3:250,3:250,3:250,3:251,3:251,
3:251,3:251,3:251,3:251,3:251,3:252,3:252,3:252,3:252,
3:252,3:252,3:252,3:253,3:253,3:254,3:254,3:255,3:255,
3:255,3:255,3:255,3:255,3:255,3:255,3:255,3:255,3:256,
3:256,3:256,3:256,3:256,3:256,3:256,3:257,3:258,3:258,
3:258,3:258,3:258,3:258,3:258,3:258,3:259,3:259,3:259,
3:259,3:259,3:259,3:259,3:259,3:259,3:259,3:259,3:259,
3:259,3:259,3:259,3:259,3:259,3:259,3:259,3:259,3:259,
3:259,3:259,3:259,3:260,3:260,3:260,3:260,3:260,3:260,
3:260,3:260,3:260,3:260,3:260,3:261,3:261,3:261,3:261,
3:261,3:261,3:261,3:261,3:261,3:262,3:262,3:263,3:263,
3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,3:263,
3:264,3:264,3:264,3:265,3:265,3:265,3:265,3:265,3:265,
3:265,3:265,3:266,3:266,3:266,3:266,3:266,3:266,3:267,
3:267,3:267,3:269,3:269,3:270,3:270,3:270,3:270,3:270,
3:270,3:270,3:270,3:271,3:271,3:272,3:272,3:272,3:272,
3:272,3:272,3:272,3:272,3:272,3:273,3:273,3:273,3:273,
3:273,3:273,3:273,3:273,3:273,3:274,3:274,3:274,3:274,
3:275,3:275,3:275,3:275,3:275,3:275,3:275,3:275,3:275,
3:276,3:276,3:276,3:276,3:276,3:276,3:276,3:277,3:277,

3:399,3:399,3:400,3:400,3:400,3:402,3:402,3:402,
3:402,3:403,3:403,3:403,3:403,3:403,3:405,3:405,
3:405,3:406,3:406,3:406,3:406,3:406,3:406,3:410,
3:410,3:410,3:410,3:410,3:410,3:410,3:410,3:410,
3:410,3:410,3:410,3:410,3:410,3:410,3:410,3:410,
3:418,3:418,3:418,3:418,3:418,3:419,3:419,3:419,
3:419,3:419,3:421,3:421,3:425,3:425,3:429,3:429,3:429,
3:429,3:429,3:434,3:438,3:438,3:440,3:440,3:440,
3:442,3:442,3:442,3:442,3:443,3:443
RAL 3:20 3:284,3:286,3:297,3:298,3:304,3:381
RALBTS 3:292 3:292,3:292
RALLYP 3:292 3:283,3:283
RALPUT 3:283 3:282,3:286
RALSHF 3:231 3:292,3:295
RAWPKT 3:19 3:46,3:244,3:244,3:247,3:247,3:269,3:269,3:269,3:272
RBFGET 2:90 2:82,2:87,2:91,3:346
RBFLLEN 1:96 1:98,1:98,1:105
RBFLOK 2:78 2:85,2:85,2:90,2:90,2:91,2:91,2:93,2:93,2:93,
3:417,3:417
RBFFPUT 2:90 2:82,2:88,2:88,2:90,3:80,3:345
RBUFE 2:78 2:90,2:90,2:91,2:93,2:93
RBUFF 2:78
RBUFLE 2:78 2:78,2:78
RBUFS 2:78 2:85,2:90,2:91,2:91,2:93,2:93
RC - 1:23
RC8SEC 3:183 3:184
RCDHEL 3:11 3:11,3:131,3:132,3:133,3:134,3:135,3:163
RCKCON 1:71 1:27,1:92,1:92,1:92,1:114,2:61
RCKTAB 1:68 1:88
RCLIP 3:28 3:309,3:309,3:309
RCLOCK 1:36 1:35,1:36,2:62
RCNTRS 3:383 3:380,3:386
RCSTAC - 1:26
RE - 2:7
REASAL 3:299 3:284,3:299,3:304
REASF 3:297 3:282,3:282,3:289,3:315,3:315
REASFO 3:297 3:297
REASF1 3:298 3:284
REASF8 3:298 3:282,3:282,3:285
REASFL 3:297 3:297
REASFV 3:298 3:298
REASFW 3:298 3:298,3:298,3:298
REASFX 3:298 3:298
REASFZ 3:298 3:298
REASG1 3:298 3:298,3:298,3:298
REASGT 3:298 3:282,3:285,3:289
REASLK 3:20 3:286,3:286,3:288,3:289,3:297,3:298,3:298,3:298,
3:299,3:299,3:304,3:315,3:315,3:381,3:381,3:421
REASQ 3:20 3:285,3:285,3:289,3:297
REASQE 3:20 3:286,3:289
REASST 3:20 3:285,3:289,3:297,3:297,3:298,3:298,3:299,3:304,3:315,
3:381,3:381,3:421
REGCH2 3:295 3:274
REGCHK 3:295 3:282,3:282,3:283,3:284
REGPKT 3:18,3:19
REGTYP 3:19 3:276,3:276,3:302,3:302,3:303,3:314
REL - 1:13,1:14
RELCOD 1:14 1:11,1:12,1:68,1:73,1:101,2:35,2:35,2:66,2:70,3:47,

3:95,3:102,3:353,3:357,3:445
 RELCON 3:95 3:95,3:99
 RELINI 1:7,1:7 1:11,2:35
 RELLOD 1:14 1:13,1:69,2:35
 RELOAD 3:49 3:122,3:410
 RELOOK 3:298 3:298
 RELPKG 3:49 3:49
 RELREQ 3:18
 RELTAB 1:7,1:7 1:11,2:35
 RELTIM 3:95 3:8
 RELTYP - 1:118,1:118,2:35,2:45,2:51
 RELVAR 1:14 1:12,1:70,1:98,3:445
 RELVST 1:14 1:13,1:14,1:14,2:35
 REMEMB - 2:7,2:7,2:7,2:7
 REMESS 3:20 3:285,3:298,3:315
 REPBIT 3:231 3:296,3:296
 REPCHK 3:296 3:275,3:277,3:277,3:277
 REPFIX 3:296 3:275,3:277
 REPLOS 3:279 3:278
 REPLU2 3:279 3:281
 REPLYB 3:278 3:277,3:287
 REPLYU 3:278 3:275,3:279
 REPLYX 3:278 3:274
 REPTAB 3:291 3:278
 REQTYP 3:19 3:248,3:249,3:286
 REQUESTS - 2:7
 RESREA 3:319 3:318,3:319
 RETCAL 3:181 3:200
 RETREE 3:200 3:199,3:202
 RFAIL 1:44 1:24,1:24,3:47,3:47,3:47,3:47,3:47,3:47,3:157,3:157,3:157,
 3:160,3:160,3:160,3:160,3:160,3:160
 RFAKE 3:408 3:380,3:408
 RFAL1 3:276 3:275
 RFAL1E 3:276 3:276
 RFAL1F 3:276 3:276
 RFAL1X 3:278 3:276
 RFAL8C 3:275 3:275
 RFAL8D 3:275 3:275
 RFATYP 3:19 3:291,3:304
 RFDONE 3:277 3:275,3:277,3:277
 RFLEDP 3:296 3:275,3:277
 RFNM0 3:277 3:277
 RFNM1 3:277 3:277
 RFNTYP 3:19 3:291,3:296,3:302,3:302,3:303,3:303,3:303,3:304
 RIGSEC 3:183 3:183
 RID 3:20 3:284,3:298,3:298,3:304,3:315
 RIHS 3:407 3:380,3:407
 RIHS11 3:407
 RING 3:28 3:83,3:321,3:397,3:397,3:415
 RINGC 3:28 3:83,3:83,3:83,3:321,3:321,3:397,3:397
 RINGE 3:28 3:83,3:321,3:397
 RINGF 3:28 3:83,3:321,3:321,3:397,3:397,3:415
 RINGLK - 3:28,3:83,3:83,3:321,3:321,3:417
 RINGLN 3:28 3:28,3:83,3:397
 RK - 1:23
 RKELIM 1:68 1:68
 RKEPAS 1:68 1:60,1:88
 RKERCK 1:68 1:12,1:60

RKEREN 1:12 1:68
RKERP 1:69
RKPATC 1:69 3:445
RKPLEN 1:16 1:69,1:69
RKSTAC - 1:26
RLDDEV 1:98 1:102,1:102,1:116,3:122
RLDINB 1:98 1:103,1:103,1:104,1:104,1:104,1:105,1:105,1:105
RLDINS 1:9,1:9 1:95
RLDMRK 3:25 3:423
RLDOTB 1:98 1:103,1:103,1:103,1:103,1:103,1:103,1:103
RLDSUB 1:102 1:101,1:105
RLDTYP 1:97,3:18 1:101,1:104,3:146,3:146,3:351
RLNREC 3:181 3:213
RMAX 3:20 3:285,3:286,3:289
RMBLKs 3:31 3:31,3:31,3:272,3:277,3:280,3:299,3:421
RMCTL 3:31 3:91,3:278,3:280,3:281,3:287,3:296,3:296,3:312,3:312
RMHOST 3:31 3:91,3:273,3:281,3:294,3:296,3:305,3:439
RMIMP 3:31 3:272,3:280,3:281,3:281,3:287,3:287,3:301,3:305,3:309,
3:309,3:312,3:315,3:404,3:421
RMLEN 3:31 3:31,3:264,3:272,3:280,3:280,3:301,3:301,3:309,3:309,
3:309,3:309,3:403,3:403,3:421,3:421
RMLHN 3:31 3:281,3:311,3:404
RMLOCK - 3:31,3:272,3:275,3:276,3:278,3:279,3:279,3:280,3:281,
3:281,3:296,3:301,3:301,3:302,3:303,3:303,3:303,3:309,
3:309,3:312,3:312,3:314,3:314,3:324,3:324,3:324,3:403,
3:404,3:421
RMESS 3:31 3:272,3:273,3:277,3:277,3:280,3:280,3:281,3:281,3:284,
3:287,3:288,3:288,3:288,3:289,3:298,3:302,3:303,3:304,
3:309,3:309,3:309,3:311,3:311,3:312,3:314,3:315,3:315,
3:324,3:324,3:404,3:404
RMNUM 3:31 3:31,3:264,3:272,3:280,3:301,3:309,3:309,3:403
RMODN 3:12 3:161,3:370
RMTYPE 3:31 3:274,3:281,3:286,3:287,3:288,3:288,3:288,3:288,3:289,
3:292,3:292,3:302,3:302,3:303,3:303,3:311,3:324,3:324
RNAL 3:381 3:380,3:381
RNGCHK 3:397 3:397
RNOBUF 3:27
ROUTEF 3:182 3:193,3:209,3:209,3:233,3:268,3:293
ROUTER 3:205 3:203,3:204,3:207
RPGNBF 3:181 3:219
RPKREC 3:181 3:212,3:212
RQLMAX 3:181 3:212,3:212,3:212
RQLSUM 3:181 3:212
RETRY 3:181 3:184,3:219
RPKTS 3:100 3:101,3:101
RRPTYP 3:19 3:291
RRQTYp 3:19 3:277,3:309
RRTIME 3:9 3:43,3:43,3:43,3:43,3:43,3:43,3:43
RSALL 3:20
RSALNU 3:20
RSBTYP 3:19 3:291,3:291,3:291,3:308
RSEX 3:12 3:137,3:152,3:152,3:167,3:234
RSF 3:20 3:285,3:286,3:289,3:297,3:304,3:381
RSFBT 3:20 3:285,3:286,3:289
RSFREE 3:20
RSTOO 3:83 3:83
RSTO1 3:83 3:83
RSTO2 3:83 3:83,3:83

RST03 3:83 3:83,3:83
 RSTART 3:83 3:63,3:63,3:84
 RSTATE 3:31 3:275,3:277,3:281,3:286,3:287,3:287,3:287,3:287,3:288,
 3:288,3:288,3:289,3:292,3:292,3:295,3:296,3:296,3:301,
 3:301,3:301,3:302,3:303,3:303,3:311,3:324
 RSTGO 3:321 3:321,3:426
 RSTMRK 3:25 3:423
 RSTRLD 3:25 3:369,3:369,3:423
 RSUCCE 1:44 1:44,3:41,3:41,3:41,3:41,3:47,3:47,3:47,3:47,3:47,
 3:47,3:47,3:47,3:104,3:104,3:104,3:104,3:104,3:104,3:104,
 3:104,3:104,3:104,3:104,3:104,3:104,3:104,3:316,3:316,
 3:317,3:317,3:317,3:317,3:317,3:318,3:318,3:318,3:318,
 3:318
 RTCADD 1:17 1:3,1:64,1:74,1:116,3:117
 RTCCHK 3:103 3:47,3:103
 RTCPDS 1:17 3:103,3:410
 RTCSWS 1:17 1:116,3:410
 RTCTEM 1:17 3:410,3:411
 RTICAL 3:181 3:198
 RTICD 3:12 3:139
 RTICL 3:12
 RTICS 3:12
 RTIMRL 3:12 3:105,3:134,3:134,3:138,3:138,3:139,3:139,3:139,3:222,3:223,
 3:223,3:223,3:223,3:227,3:227,3:227
 RTIMRS 3:12 3:134,3:223,3:223,3:223,3:223,3:223,3:227,3:227
 RTINC 3:198 3:190,3:199
 RTINIT 3:210 3:183,3:210
 RTLOOP 3:183 3:185
 RTOFF 3:227 3:164,3:227
 RTON 3:227 3:214,3:227
 RTRCLK 3:12 3:138,3:139,3:222,3:222
 RTRCLR 3:134 3:132,3:133,3:134
 RTRCNT 3:181 3:212,3:226
 RTRGEN 3:224 3:223,3:226
 RTRNBF 3:181 3:226
 RTRTIC 3:12 3:138,3:139,3:222
 RTRTO 3:222 3:184,3:223
 RTRYAD 1:19 1:31,2:62
 RTRYPC 1:19 1:31,2:62
 RTSET 3:227 3:227,3:227,3:227
 RUBOUT 2:105 2:98,2:99,2:103,2:105,2:108,3:64,3:69,3:71,3:72,3:72
 RUBTXT 2:106 2:105
 RUP4US 3:182 3:95,3:186,3:186,3:212,3:214,3:214,3:217,3:224,3:229
 RUPADD 3:181 3:187,3:187,3:188
 RUPAGE 3:182 3:166
 RUPBLD 3:221 3:220,3:221,3:226
 RUPCKS 3:211 3:184,3:211
 RUPCNT 3:181 3:212,3:220
 RUPDEL 3:182 3:219
 RUPDEQ 3:215 3:145,3:186,3:215
 RUPDP1 3:181 3:166
 RUPEND 3:181 3:187,3:187,3:188
 RUPENQ 3:212 3:164,3:214,3:221
 RUPFLS 3:217 3:100,3:141,3:145,3:186,3:218
 RUPGEN 3:219 3:184,3:220
 RUPMAP 3:181 3:187,3:187,3:188
 RUPMSK 3:181 3:95,3:166,3:220,3:229
 RUPNBF 3:181 3:164

RUPNEI 3:182 3:219
RUPNN 3:182 3:182,3:182,3:182,3:187,3:212,3:221,3:225
RUPNN1 3:182 3:226
RUPOBS 3:181 3:166
RUPQCK 3:229 3:125,3:229,3:316
RUPQCT 3:181 3:212,3:212,3:212,3:217,3:229,3:229
RUPQSZ 3:29
RUPRET 3:182 3:164,3:164,3:226
RUPSEA 3:188 3:187,3:188
RUPSHO 3:182 3:166
RUP SND 3:181 3:138,3:139,3:179,3:184,3:184,3:430
RUP SNO 3:182 3:166,3:220
RUP SRC 3:182 3:164,3:220
RUPT25 3:182 3:183,3:184,3:184
RUPTCK 3:182 3:183,3:184,3:184
RUPTYP 3:18 3:221
RUPWHC 3:216 3:215,3:216,3:218,3:225,3:229
RUSE 3:20 3:284,3:298,3:304,3:315
RUT - 3:182
RUTCLK 3:12
RUTCOD - 3:7,3:7,3:44,3:100,3:125,3:173,3:176,3:183,3:186,3:217,
3:219,3:228
RUTDAT 3:19
RUTDSP 3:160 3:160,3:160
RUTINI 3:182 3:132,3:183,3:184
RUTLOK - 3:182,3:185,3:185,3:185,3:210,3:417
RUTNUL 3:18 3:151
RUTOFF 3:185 3:183,3:184,3:185
RUTPI 3:24 3:117,3:138,3:139,3:185,3:210,3:214,3:426,3:430
RUTSLE 3:185 3:176,3:177,3:185,3:186,3:192,3:195,3:197,3:197,3:199,
3:201,3:202,3:204,3:207,3:210,3:211,3:211,3:221,3:223,
3:223
RUTSP 3:182 3:185,3:185
RUTSPD 3:12 3:136
RUTSPF 3:186 3:184,3:186
RUTSTA - 3:183
RUTTYP 3:18 3:144,3:151,3:167,3:221
RUTUPL 3:18
RUTVAR - 3:7,3:7,3:181,3:181
RUTWAI 3:12 3:132,3:133
RUTWAK 3:185 3:426
RXMCHE 3:121 3:118,3:121
S - 2:7
SAROO 2:60 2:35
SARPCN 2:66 2:62,2:67
SARPOL 2:64 2:62,2:65
SARWDG 2:70 2:62,2:70
SATMHI 3:9
SATMID 3:9
SAVEPA - 2:7,2:7,2:7,2:9,2:16,2:28
SBAD 1:62 1:51,1:55,1:59,1:62,1:62,1:118,2:58,2:59
SBDOO 1:73 1:26
SBDCLR 1:77 1:75,1:77
SBDQCH 1:76 1:73,1:76
SBDTIM 1:77 1:73,1:75,1:77
SBLK 3:11 3:147,3:147,3:354,3:354,3:354
SCDOO 1:78 1:26
SCDBBC 1:82 1:80,1:83

SCDBUS 1:22 1:79,1:80,1:80,1:81,1:81
 SCDSET 1:84 1:82,1:83,1:84
 SCDTAB 1:82 1:82
 SCDTST 1:80 1:79,1:81
 SCKQ 3:26 3:376,3:417
 SCLEAR 1:63 1:62,1:63,2:58,2:66
 SCLR0K 1:62 1:55,1:61,1:62,1:74,1:79,2:36,2:39
 SCN 3:181 3:200,3:201,3:201,3:201,3:201
 SCNTLP 3:347 3:343
 SDAC 3:9 3:42,3:42,3:45
 SDBBLK 2:26 2:26
 SEARCH 3:204 3:199,3:204
 SEC1 3:9 3:9,3:9,3:9
 SEC15 3:9 3:91,3:258,3:258,3:266
 SEC3 3:9 3:112,3:255,3:267,3:302
 SEC30 3:9 3:90,3:290
 SEGCON 1:28 1:27,1:78,2:61,3:369
 SEGFIX 1:28 1:27
 SEGINC 1:72 1:72,1:102
 SEGMSK 1:72 1:102
 SEMICO 3:347 3:67,3:347,3:347,3:348,3:348
 SENDST 1:115 1:110,1:111
 SEQH 3:17 1:97,3:19,3:19,3:19,3:91,3:144,3:144,3:144,3:144,3:146,3:151,
 3:151,3:151,3:153,3:159,3:161,3:182,3:187,3:187,3:188,
 3:219,3:221,3:226,3:259,3:259,3:265,3:273,3:278,3:278,
 3:281,3:281,3:282,3:285,3:285,3:293,3:299,3:307,3:322,
 3:439
 SETBLT 2:115 2:109,2:110,2:110,2:116
 SETDEP 2:109 2:107,2:107,2:109
 SETUP 1:51
 SFHCQ 3:29 3:355,3:417
 SFIXIT 1:63 1:62,1:63,1:79
 SFSBIT 3:46
 SFXBAD 1:62 1:55,1:61,1:62,1:74,1:75,1:79,1:79,2:36,2:38,2:64
 SHIHY 3:11 3:132,3:133,3:144,3:144,3:163
 SHPQ 3:14 3:89,3:90,3:90,3:91,3:109,3:407
 SHQ 3:14 3:89,3:90,3:91,3:91,3:91,3:109,3:323,3:407
 SIDOO 2:56 2:35
 SIDFLG 1:29 2:58,3:93,3:94,3:94,3:94,3:118,3:129,3:414,3:414
 SIGN 1:17,3:9 1:47,1:71,1:98,2:93,3:11,3:13,3:14,3:21,3:30,
 3:30,3:81,3:95,3:96,3:96,3:118,3:153,3:182,3:262,3:287,
 3:287,3:330,3:332,3:369,3:370,3:370,3:371,3:373,3:376,
 3:378,3:378,3:378,3:378,3:404,3:425,3:438
 SIOIN 3:58 3:58,3:154,3:155
 SJ2 1:45
 SJ6 1:45 2:128
 SJIF 1:35 1:13
 SKIP - 3:239
 SLASH 2:110 2:101,2:106,2:111
 SLAVE 3:11
 SLCOO 2:36 2:35
 SLEEP 2:20,2:32
 SLFLK - 1:28,1:39,1:58,1:58,1:58,2:39,2:39,2:127,2:127,2:127
 SLFPTR 1:28 1:34,1:58,1:58,1:114,2:126,2:126,2:126,2:131,3:50,
 3:54,3:56,3:57,3:153,3:187,3:259
 SLKOO 1:52
 SLOTS 3:12 3:101,3:148,3:171,3:233,3:236,3:382
 SLP - 2:20,2:20,2:32,2:32

SLPENT 2:20,2:32
SLSTAC 1:23 1:45,1:89,2:64
SLSTKL 1:14 1:23
SMDOO 1:54 1:26
SMDBLK 1:28 1:57,1:57,1:57,1:58,1:58,1:58
SMDBUC 1:28 1:56
SMDCON 1:27 1:27,1:57,1:57
SMDFLG 1:22 1:54,1:55,1:59
SMDTIM 1:22 1:54,1:55
SMDTS2 1:57 1:56,1:59
SMDTST 1:56 1:54,1:56
SMIQ 3:12 3:105,3:156
SMMOO 2:38 2:35
SMMBAS 1:69 1:69,2:43
SMMCHE 2:45 2:43,2:47,2:48
SMMCOP 2:52 2:49,2:50,2:50,2:52
SMMFIX 2:49 2:38
SMMFRE 1:22 2:42,2:47,2:49
SMMFTY 1:22 2:42,2:47,2:47,2:47
SMMINS 1:69 2:43
SMMOK 1:22 2:42,2:43,2:45,2:49,2:50
SMMQCH 2:54 2:45,2:54
SMMQFX 2:55 2:54,2:54,2:55
SMMSCA 2:47 2:43,2:43,2:48
SMMSEA 2:41 2:38
SMMMSMA 2:52 2:39,2:39,2:53
SMMSPA 1:22 2:42,2:43,2:43,2:46,2:49,2:50
SMOCLK 3:181 3:176,3:176
SMOOTH 3:177 3:176,3:177
SMOSHF 3:174 3:138,3:178
SMSOK 3:289 3:289
SMSTK 3:289 3:288
SNAP 1:21 1:39,1:51
SNAPBF 3:360
SNAPBG 1:19 1:19,1:39
SNAPIL 1:39 1:37,1:40
SNAPLN 1:19 1:39
SNAPLO 1:38 1:36,1:36,1:40
SNDING 3:11 3:100,3:141,3:141,3:152,3:169,3:169,3:382
SNDOUT 3:274 3:274,3:274
SNON 3:360 3:85,3:362,3:376,3:402,3:402,3:402,3:417,3:417
SNULL 3:11 3:151,3:152,3:152,3:152,3:152,3:168,3:234
SOKAY 1:63 1:53,1:62,1:63,1:118,2:59,2:62
SOMETH 2:80 2:105,2:105,2:107,3:66
SOWDTM 3:359 3:359
SP - 1:33,1:35,1:37,1:45,1:45,1:45,1:49,1:89,2:64,2:64,
2:65,2:84,2:84,2:84,2:84,2:85,2:94,2:94,2:94,2:94,
2:94,3:50,3:50,3:51,3:51,3:53,3:54,3:56,3:77,3:77,
3:77,3:77,3:77,3:89,3:94,3:94,3:94,3:127,3:128,3:128,
3:129,3:129,3:183,3:185,3:185,3:195,3:214,3:226,3:239,
3:241,3:241,3:246,3:247,3:248,3:249,3:251,3:255,3:256,
3:257,3:260,3:260,3:263,3:269,3:270,3:277,3:278,3:279,
3:290,3:293,3:293,3:300,3:324,3:326,3:326,3:326,3:334,
3:334,3:334,3:382,3:394,3:395,3:431,3:431
SPACE 2:104 2:7,2:100,2:104
SPECAL 3:14 3:89,3:90,3:90
SPF 3:189 3:187,3:188,3:190
SPFAG1 3:182 3:228

SPFAGE 3:182 3:166,3:166,3:220,3:220,3:226,3:228
 SPFCRA 3:183 3:210
 SPFDED 3:182 3:191,3:192,3:192,3:210,3:210,3:246,3:261,3:268,3:293,
 3:402,3:404
 SPFERR 3:210 3:189,3:194,3:195,3:195,3:200,3:201,3:205,3:207,3:208,
 3:209,3:211,3:211
 SPFON 3:18 3:144
 SPFPOS 3:191 3:190,3:193
 SPFRTL - 3:164,3:166,3:166,3:182,3:192,3:192,3:209,3:209,3:211,
 3:211,3:220,3:220,3:224,3:225,3:225,3:228,3:228,3:417
 SPFRUT 3:182 3:166,3:166,3:166,3:166,3:182,3:192,3:192,3:192,
 3:192,3:193,3:209,3:209,3:209,3:209,3:209,3:210,3:210,
 3:211,3:220,3:220,3:220,3:220,3:225,3:228,3:228,
 3:233,3:246,3:261,3:268,3:293,3:402,3:404
 SPFSN1 3:182 3:220
 SPFSNO 3:182 3:166,3:166,3:220,3:220,3:226
 SPFSUM 3:182 3:166,3:192,3:192,3:209,3:210,3:211,3:220,3:228
 SPFTIC 3:228 3:228
 SPRIG - 3:236
 SPRIQ 3:11 3:100,3:100,3:105,3:143,3:143,3:382
 SRCOO 1:118 1:26
 SRCEH 3:16 3:9,3:16,3:269,3:294,3:438
 SRCEHI 3:16
 SRCEI 3:16 3:9,3:16,3:269,3:294,3:438
 SRCH 3:17 1:97,3:91,3:144,3:144,3:146,3:151,3:156,3:156,3:159,
 3:161,3:164,3:166,3:182,3:187,3:220,3:221,3:225,3:226,
 3:259,3:263,3:272,3:280,3:293,3:294,3:307,3:439
 SRCHST 3:19 3:294
 SRCL - 3:91
 SRCST4 3:288 3:289
 SREQQ 3:11 3:100,3:100,3:105,3:143,3:236,3:382
 SRKOO 1:60 1:26
 SRKKER 1:22 1:60,1:60,1:60,1:60,1:60,1:61
 SRKREL 1:22 1:60,1:61,1:61
 SRQ 3:26 3:119,3:417
 SRTSHF 3:174 3:138
 SRUPQ 3:181 3:215,3:218,3:224,3:229,3:417
 SS2BIT 2:79
 SS4BIT 2:79
 SSENTQ 3:11 3:100,3:105,3:121,3:121,3:148,3:148,3:148,3:169,3:382
 SSHOST 3:46 3:355
 SSIMP 3:46 3:355
 ST 3:17 3:121,3:148,3:152,3:175
 STACK 1:23,1:23 1:23,1:23,1:23,1:23,1:23,1:23,1:23,1:23,1:23,
 1:23,1:23,1:23,2:76,2:77,3:23,3:23,3:28,3:29,3:76,
 3:182,3:429
 STAGE. - 1:1
 STAGEK - 1:15
 STAGEP 2:128
 STAIOB 3:33 3:85,3:85
 STARTI 1:96,3:23 1:100,3:59,3:328,3:329,3:330
 STARTO 1:96,3:23 1:103,3:153,3:153,3:270,3:322,3:323,3:335,
 3:336,3:336,3:338
 STARTU 2:20,2:32
 STATB 3:360 3:360
 STATD 1:96,3:23 3:112,3:328
 STATDT 3:360 3:363,3:363,3:422,3:422
 STATF 3:360 3:362,3:370,3:373,3:402,3:416,3:416,3:416,3:416

3:416,3:416,3:416,3:424
STATH2 3:374 3:371
STATHD 3:374 3:369,3:373,3:374
STATIH 3:23 3:59,3:89,3:107,3:108,3:115,3:123,3:251,3:253,3:254,
3:256,3:319,3:326,3:331,3:331,3:377,3:398
STATIM 1:96,3:23 1:103,1:104,3:108,3:115,3:124,3:137,3:154,
3:154,3:154
STATL 3:360 3:360,3:360,3:360,3:360,3:360,3:360,3:360
STATOH 3:23 3:88,3:89,3:108,3:109,3:112,3:115,3:123,3:323,3:335,
3:336,3:337,3:337,3:338,3:377
STATOM 1:96,3:23 1:103,1:103,3:108,3:115,3:124,3:137,3:141,
3:141,3:141
STATOT 3:361 3:362,3:424
STATS 3:362 3:378
STATUS 3:369 3:361,3:370
STBP 3:361 3:362,3:424
STBPSP 3:361
STERR 3:284 3:282,3:283,3:284
STETAB 2:96 2:96,2:97
STGCON 1:28 1:27
STGCYC 1:16 1:16,1:16,1:16,1:16,1:49
STGFIX 1:28 1:27
STGPAS 1:16 1:60,1:68,1:88
STGRAT 1:16 1:16,1:36,1:45,1:46,1:64
STGTIC 1:16 1:16
STGTIM 1:70 1:73,1:77
STIM2 1:19 1:47,2:70,3:359
STIME 1:19 1:47,1:48,1:48,1:49,1:49,1:54,1:55,1:73,1:77,1:87,
1:89,1:90,1:101,1:112,1:114,1:115,2:60,2:68,2:70,2:82,
2:115,3:63,3:359
STIMER 3:17 3:143,3:148,3:149
STKPAS 2:12 2:62,2:126,2:126
STKPID 3:24 3:83,3:426
STNXT 3:127 3:127
STO 3:129 3:28,3:426
STOINI 3:28 3:129,3:129,3:129,3:129
STOLOK - 3:28,3:128,3:129,3:129,3:415
STOLOO 3:127 3:129
STOPID 3:24 3:118,3:128,3:128,3:426
STOPS 3:71 3:71,3:71
STOSP 3:28 3:128,3:129
STOSTA - 3:127,3:128,3:129
STQ 3:26 3:232,3:417
STRBIT 3:46
STRIP 2:27 2:26
STS8 3:360 3:361,3:370,3:402,3:402,3:415,3:416
STUBIT 3:18
STVEC - 2:22,2:22,2:22,2:22,2:22,2:22
SUBCHN 3:86 3:86,3:211,3:211,3:211,3:211,3:316
SUBRT - 3:74,3:74
SUBTRE 3:181 3:201,3:201,3:202,3:202
SUBTYP 3:19 3:90,3:244,3:247,3:252,3:267,3:269,3:272,3:296,3:305,
3:305,3:394,3:394
SUCBUF 3:338
SUCCEE - 1:54,1:60,3:157,3:164,3:257,3:300,3:341,3:350,3:359
SUCK 3:335
SUCKLE 3:336
SUCMSG 3:337 3:337,3:342,3:347,3:350,3:359

SUKCHK 3:337 3:335,3:337
 SUKPKT 3:337 3:335,3:336,3:337,3:337,3:338,3:338
 SUKPOK 3:337 3:337
 SUM 2:80 2:98,2:104,2:105,2:107,3:66,3:69,3:69,3:72
 SVTIME 1:22 1:48,1:48,1:48
 SW 1:7 1:7,1:7
 SYNC 3:25 3:117,3:144,3:151,3:161,3:362,3:362,3:371,3:374
 SYNPAG - 3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:7,3:8
 SYSFCB 2:13 2:13
 SYSUQ 1:32 1:52
 SYSVER 3:46 3:369,3:369
 SYTIM2 1:72 1:47,3:117
 SYTIME 1:28 1:47,1:48,3:117
 T - 2:6,2:7
 T2FGET 3:346 3:343,3:343,3:346
 T2FPOK 2:78 2:88,3:346
 T2HO 3:290 3:283
 TAB - 1:5
 TALLYG 3:267 3:248,3:249
 TASK 3:24 3:55,3:119,3:232,3:426
 TB128K 3:173 3:173,3:173
 TB256K 3:173 3:173,3:173
 TB32KB 3:173 3:173,3:173
 TB39OK 3:173 3:173,3:174
 TB48KB 3:173 3:173,3:173
 TB64KB 3:173 3:173,3:173
 TBFPID 3:24
 TBKGO - 3:28,3:320,3:415
 TBKGDA 3:28 3:415
 TCGO - 3:28,3:317,3:415
 TCGOA 3:28 3:415
 TCLIP 3:28 3:308,3:308,3:308,3:313
 TDELT 3:12 3:133,3:135,3:135
 TDTAB 3:173 3:173,3:175,3:178
 TE - 3:272
 TEMP - 2:7,2:7,2:7,2:7,2:9,2:9,2:16,2:16,2:28,2:28
 TEMP1 1:21 1:87,1:89,2:61,2:61,3:286,3:286,3:292,3:292,3:293,
 3:294,3:299,3:299,3:299,3:324,3:324
 TEMP2 1:21 1:85,1:86,1:86,1:87,1:90,1:91,1:92,1:92,1:93,3:293,
 3:294,3:297,3:297
 TEMP3 1:21 1:85,1:88,1:88,1:89,1:91,1:93,1:95
 TEMP4 1:21 1:93,1:93,1:94,1:95,2:131,2:131
 TEMREF 3:46 3:410
 TEMWRD 3:341 3:410,3:410,3:411
 TENEXC 3:9
 TENEXE 3:9
 TENEXN 3:9 3:9
 TESTSW 1:7 3:5
 TEXTOU 2:121 2:96,2:113,2:121
 TF3 - 1:85
 TG - 1:91,1:91
 TH 3:9 3:25,3:25,3:96,3:102,3:106,3:114,3:407
 THD 3:182 3:161,3:193
 THE - 2:7,2:7
 THIS - 2:6,2:7,2:7
 THRDCY 3:174 3:179
 THRESH 3:181 3:179,3:179,3:179,3:219
 THRINI 3:174 3:219

THROTC 3:360 3:402,3:402,3:402,3:402,3:402
THRUPRT 3:11 3:171,3:371
THRURP 3:371 3:361,3:372
THRUSB 3:360 3:361,3:402,3:402,3:415,3:416
TICAGE 3:228 3:228,3:228
TICIMP 3:181 3:228,3:228
TICKIN 3:129 3:129,3:380
TICTIM 3:405 3:405
TIKCNT 3:29 3:118,3:131,3:132,3:133
TIKRAT 3:12 3:131
TIKTIM 3:12 3:131,3:131
TIM - 1:7,1:7,1:7,1:7
TIME 3:25 3:91,3:117,3:183,3:183,3:183,3:184,3:290,3:363
TIMEA 3:25 3:120
TIPBIT 3:46
TIPVER 3:46 3:369
TL - 1:91,3:245
TLIMIT 1:28 1:68,1:118,2:37,2:50,3:49,3:49
TLOG 3:364 3:361,3:364
TLOGFR 3:364 3:416
TLOOKU 3:66 3:66,3:69,3:71,3:71
TLSB 3:360 3:361,3:402,3:402,3:415,3:416,3:416,3:417
TLT1 3:365 3:364,3:365
TLT2 3:366 3:364,3:366
TLT3 3:366 3:364,3:366
TLT4 3:367 3:364,3:367
TLT5 3:368 3:364,3:368
TLT6 3:368 3:364,3:368
TLTDSP 3:364 3:364
TLTSub 3:365 3:365
TLTTCP 3:360 3:416,3:416,3:416,3:416,3:417
TM - 3:246
TMALL 3:32 3:267,3:308,3:313,3:313
TMALLO 3:32 3:32,3:267,3:275,3:313,3:314
TMALT 3:32 3:32,3:32,3:263,3:267,3:313,3:313,3:313
TMALTO 3:32 3:313
TMBLKs 3:31 3:31,3:272,3:277,3:299,3:421
TMCTL 3:31 3:259,3:263,3:263,3:264,3:265,3:287,3:296,3:296,3:312,
3:312,3:394,3:394
TMDEDS 3:31 3:265,3:287
TMHOST 3:31 3:261,3:263,3:264,3:294,3:296,3:394
TMIMP 3:31 3:261,3:261,3:262,3:263,3:263,3:264,3:272,3:287,3:306,
3:308,3:308,3:312,3:313,3:394,3:404,3:404,3:406,3:421
TMINIT 3:32 3:263,3:264,3:287,3:306,3:306,3:308,3:404,3:404
TMLEN 3:31 3:31,3:261,3:261,3:261,3:261,3:264,3:272,3:306,3:306,
3:308,3:308,3:308,3:308,3:313,3:313,3:403,3:403,3:405,
3:406,3:406,3:421,3:421
TMLHN 3:31 3:263,3:264,3:394,3:404
TMLOCK - 3:31,3:261,3:261,3:262,3:262,3:263,3:264,3:264,3:264,
3:265,3:265,3:267,3:267,3:272,3:276,3:279,3:296,3:306,
3:306,3:308,3:308,3:312,3:312,3:313,3:313,3:314,3:314,
3:393,3:394,3:403,3:404,3:404,3:404,3:404,3:404,3:404,
3:404,3:404,3:404,3:405,3:405,3:405,3:406,3:406,3:421
TMMESS 3:31 3:261,3:263,3:263,3:263,3:264,3:265,3:265,3:265,3:272,
3:277,3:277,3:287,3:305,3:308,3:308,3:308,3:312,3:313,
3:314,3:314,3:314,3:393,3:393,3:404,3:404,3:404,3:404,
3:405,3:405
TMNOT 3:287 3:287

TMNUM 3:31 3:31,3:261,3:261,3:264,3:272,3:306,3:308,3:308,3:313,
 3:393,3:403,3:406
 TMRSET 3:32 3:264,3:277,3:287,3:306,3:306,3:308,3:308,3:308,3:404,
 3:404
 TMSTOP 3:32 3:263,3:265,3:277
 TMSTP 3:31 3:263,3:265,3:267,3:267,3:277,3:313,3:313
 TNE - 1:92
 TNF 1 - 1:86,1:86
 TNG - 3:272
 TNO - 1:87
 TNZ - 1:92,1:92,1:95,3:242,3:243,3:243,3:244,3:248,3:261,
 3:264,3:274,3:295,3:295,3:296
 TO - 2:7,2:7,2:7,3:248,3:290
 TO15SE 3:429
 TO5SEC 3:429 3:429,3:429,3:430
 TOHOT 3:119,3:119 3:47,3:410
 TOPNTR 1:28 2:37,3:74,3:93,3:127
 TORTD 3:360 3:363,3:363
 TORTM 3:360 3:363,3:363
 TOSMOO 3:176 3:176,3:184
 TOSST 3:129 3:74,3:74,3:129,3:380
 TOSSX 3:127 3:315,3:382,3:395,3:396,3:403,3:431
 TOTAL - 1:13,1:14
 TOTMP1 3:28 3:385,3:385
 TOTSTB 1:16 1:45,1:48,1:63,1:112
 TOTSTS 1:16 1:16,1:16,1:22,1:27,1:47
 TOTXT 2:114 2:112
 TR - 1:86,1:86,1:86,1:87,1:88,1:91,1:91,1:92,1:94,2:128,
 3:49,3:63,3:63,3:63,3:63,3:88,3:91,3:129,3:129,3:185,
 3:243,3:243,3:246,3:247,3:247,3:249,3:253,3:256,3:256,
 3:261,3:263,3:264,3:274,3:274,3:274,3:279,3:279,3:280,
 3:286,3:290,3:296,3:382,3:404,3:431
 TRABIT 3:46
 TRACE - 3:236
 TRAP - 1:9,1:31,1:33,1:35,1:35,1:35,1:35,1:35,1:38,1:47,1:48,
 1:50,1:52,1:53,1:55,1:66,1:66,1:75,1:76,1:80,1:81,
 1:82,1:84,1:87,1:91,1:92,1:93,1:95,2:36,2:40,2:44,
 2:51,2:55,2:62,2:63,2:70,2:85,2:125,3:50,3:50,3:53,
 3:53,3:57,3:57,3:64,3:65,3:83,3:88,3:99,3:103,3:105,
 3:107,3:108,3:108,3:110,3:111,3:111,3:111,3:112,3:112,
 3:112,3:113,3:113,3:120,3:120,3:129,3:131,3:135,3:141,
 3:141,3:145,3:148,3:149,3:149,3:153,3:154,3:155,3:156,
 3:156,3:157,3:158,3:161,3:163,3:169,3:169,3:186,3:210,
 3:212,3:212,3:212,3:216,3:217,3:217,3:218,3:225,3:229,
 3:229,3:232,3:233,3:233,3:249,3:249,3:249,3:250,
 3:251,3:251,3:251,3:251,3:252,3:252,3:255,3:255,3:255,
 3:256,3:257,3:260,3:261,3:264,3:266,3:271,3:274,3:274,
 3:274,3:275,3:276,3:276,3:277,3:277,3:277,3:279,3:279,
 3:280,3:280,3:282,3:282,3:284,3:284,3:285,3:287,
 3:288,3:290,3:295,3:305,3:305,3:323,3:323,3:323,3:324,
 3:329,3:329,3:332,3:332,3:336,3:338,3:338,3:338,3:354,
 3:355,3:382,3:389,3:390,3:391,3:391,3:391,3:395,3:395,
 3:395,3:397,3:404,3:407,3:410,3:410,3:419,3:423,3:429,
 3:438,3:439,3:440,3:441,3:441,3:443
 TRAPCN 1:40 1:37,1:40
 TRAPRP 3:373 3:361,3:373
 TRAPV - 1:37,1:37
 TRBLCK 3:360 3:362,3:370,3:372,3:373,3:376,3:376,3:376

TRBSB2 3:375 3:369,3:373,3:375
TRBTAB 3:402 3:402,3:402
TRBTIM 3:402 3:380,3:402
TRBTL 3:402 3:402
TRCBIT 3:18 3:91,3:242,3:244,3:269
TRCIH 3:322
TRCM2I 3:171
TRDEDS 3:22 3:252,3:271,3:277,3:394,3:394
TRDSTL 3:22 3:244,3:246,3:255,3:259,3:261,3:263,3:264,3:434,3:434,
3:434,3:434,3:434,3:434
TRENDA 3:27 3:381
TRENDF 3:27 3:386
TRENDS 3:27 3:386
TRHOST 3:22 3:87,3:271,3:271,3:395,3:395,3:395
TRHSTL 3:22 3:87,3:87,3:242,3:242,3:244,3:244,3:246,3:248,3:249,
3:259,3:271,3:299,3:305,3:314,3:393,3:393,3:393,3:434,
3:434,3:434
TRLEDR 3:22 3:107,3:109,3:123,3:243,3:244,3:244,3:244,3:244,3:244,
3:244,3:244,3:244,3:319
TRLUSE 3:22 3:22,3:248,3:314,3:393
TRMIDL 3:22 3:241,3:243,3:244,3:244,3:244,3:247,3:252,3:259,3:267,
3:267,3:271,3:296,3:296,3:305,3:305,3:394,3:394,3:396
TRNBLK 3:31 3:31,3:87,3:256,3:266,3:266,3:271,3:271,3:271,3:271,
3:271,3:271,3:271,3:271,3:299,3:299,3:305,3:305,
3:393,3:395,3:396,3:396,3:414,3:414
TRNETL 3:22 3:22,3:242,3:244,3:252,3:256,3:271
TRNFLS 3:87 3:87,3:277,3:394,3:395
TRNL 3:22 3:31,3:248,3:248,3:266,3:266,3:271,3:271,3:299,3:305,
3:395,3:396,3:396
TRNLOK - 3:27,3:266,3:266,3:266,3:396,3:396,3:396,3:417
TRNNUM 3:31 3:31
TRNPT1 3:266 3:266
TRNPT2 3:266 3:266
TRNPT4 3:266 3:266
TRNPUT 3:266 3:242,3:248,3:314,3:314
TRNREP 3:19 3:272,3:274
TRNTIM 3:22 3:22,3:87,3:246,3:248,3:266,3:271,3:296,3:296,3:314,
3:394,3:394,3:395,3:395,3:395
TRNTOT 3:31 3:31,3:266,3:271,3:299,3:305,3:395,3:396
TRPACK 3:22 3:22,3:87,3:87,3:249,3:251,3:276,3:276
TRPSB 3:360 3:361,3:373,3:402,3:402,3:415,3:416
TRSTAT 3:22 3:87,3:237,3:246,3:248,3:248,3:250,3:251,3:266,
3:266,3:271,3:275,3:276,3:296,3:296,3:299,3:305,3:305,
3:305,3:314,3:393,3:393,3:394,3:394,3:395,3:395,3:396
TRYTPL 3:22 3:242,3:242,3:244,3:244,3:244,3:259,3:267,3:271,3:271,
3:277,3:296,3:305,3:394
TRUE - 2:17,2:17,2:17,2:29,2:29,2:29
TRYMOD 3:55 3:55,3:234
TRYRLD 1:9,1:116 1:118,2:36,2:50
TSEX 3:12 3:137,3:150,3:167,3:167,3:167,3:167,3:167,3:167,3:167
TSK 3:232 3:233,3:426
TSKACK 3:234 3:233,3:233,3:233,3:234
TSKAHK 3:237 3:234,3:235,3:237
TSKBTS 3:231 3:273,3:283,3:289,3:296
TSKBUF 3:231 3:272,3:278,3:278,3:278,3:279,3:282,3:283,3:283,3:283,
3:286,3:286
TSKFOK 3:13 3:234,3:237,3:255,3:255
TSKFRE 3:13 3:237,3:237,3:242,3:260

TSKFRF 3:13 3:235,3:255,3:255
 TSKHST 3:231 3:281,3:290,3:294
 TSKNAK 3:235 3:233,3:235
 TSKPUT 3:55 3:55,3:168
 TSKREQ 3:235 3:235
 TSKSF 3:236 3:233,3:236
 TSKVHA 3:439 3:293,3:439
 TSLEE2 3:128 3:127,3:128,3:129
 TSLEEP 3:128 3:127,3:128,3:129,3:385,3:385,3:386,3:405,3:408,3:442,
 3:443,3:443
 TST2DE 3:113 3:111,3:112,3:113
 TST2PI 3:115 3:105,3:107,3:111,3:112,3:115
 TSTATE 3:31 3:263,3:264,3:265,3:265,3:265,3:267,3:267,3:275,3:277,
 3:287,3:287,3:287,3:287,3:296,3:296,3:306,3:306,3:306,
 3:308,3:308,3:308,3:313,3:313,3:313,3:314,3:314,3:314,
 3:404,3:404,3:406,3:406,3:406
 TSTJAM 3:376 3:370,3:370
 TSTS 3:369 3:369,3:369
 TT - 2:77
 TTGVBA 3:22 3:250,3:305,3:314
 TTLEDR 3:22 3:87,3:299,3:305,3:393
 TTMULT 3:22 3:248,3:248,3:275
 TTOADR 3:46 3:428
 TTOPID 3:24 3:117,3:426,3:428
 TTPEND 3:22
 TTPOKE 2:78 2:89,3:63
 TTREQ 3:22 3:246,3:248,3:248,3:275,3:276
 TTRESV 3:22 3:237,3:246,3:248,3:251,3:266,3:299,3:305,3:393
 TTRFNFM 3:22 3:246,3:296,3:394
 TTSKPU 3:55 3:55,3:100,3:100,3:255,3:278
 TTSLEE 2:84 2:84,2:85,2:87,2:89
 TTSTAC - 2:85
 TTSYSS 2:77 2:84,2:84
 TTTCPE 3:360
 TTY 2:87
 TTYBIT 3:46
 TTYIBF 2:78 2:82,2:88,2:93,2:93,2:93,3:346,3:417
 TTYINI 2:85 2:84,2:85,2:85,2:88,2:89
 TTYIOB 3:33
 TTYLOK - 2:78,2:84,2:84,2:84,3:417
 TTYOBF 2:78 2:82,2:85,2:85,2:85,2:87,2:88,2:93,2:93,2:93,3:80,
 3:345,3:417
 TTYOUT 2:89 2:87,2:89
 TTYPID 3:24 3:63,3:80,3:117,3:345,3:426
 TTYPOL 2:84 2:81,3:63
 TTYSP 2:77 2:84,2:84
 TTYSTU 2:87 2:87,2:88,2:89
 TTYWAK 3:63 3:63
 TYP - 1:6,2:4,2:7,2:
 TYPE - 2:7,2:
 TYPE4K 1:28 1:40,2:37,2:45,2:45,2:51,2:51,2:53,3:389
 TYPH 1:97,3:17 1:104,3:91,3:144,3:144,3:146,3:151,3:151,3:152,
 3:152,3:157,3:159,3:160,3:163,3:167,3:167,3:221,3:236,
 3:259,3:263,3:272,3:274,3:274,3:278,3:279,3:287,3:290,
 3:307,3:351,3:355
 TYPL 3:16 3:90,3:91,3:269,3:269,3:336,3:341,3:342,3:359
 TYPUPD - 3:190
 TZ - 1:92,1:92,3:247,3:274,3:296,3:300

UCTRL 1:19 1:32, 1:39, 1:39, 1:39
UILLOP 1:19 1:39
UJIFFY 1:19 1:36
UMAP 1:19 1:39
UNHACM 3:19 3:282
UNLOOP 3:70 3:70
UNPACK 3:56 3:55, 3:56, 3:121, 3:147, 3:171, 3:212, 3:215, 3:225, 3:290,
3:322
UNPCKC 3:56 3:56, 3:155, 3:259, 3:290, 3:338
UQUIT 1:19 1:31, 1:32, 1:52
UQUITD 1:19 1:32
UQUITP 1:19
UQUITS 1:19
USEBUS 1:70 1:74, 1:74, 1:80, 1:81, 1:93, 1:102, 3:103, 3:120, 3:410
USEIO 1:72 1:72, 2:58, 2:58, 2:58, 2:131, 3:98
USEIOL 1:72 1:102, 2:131, 3:98
UTIME 1:22 1:45, 1:46
UWST 1:19 1:51
V2 2:34 2:34, 2:124, 3:30, 3:31, 3:37
V2PAT 3:31
V2PBLK 3:25 3:96, 3:96, 3:98, 3:114, 3:118, 3:377, 3:378, 3:408, 3:414
V2ST 2:34 2:34
VALUE - 2:4, 2:4
VAR - 1:13, 1:14
VARPAT 3:31 3:445
VARS 1:14 1:13, 1:14, 1:28, 1:29, 2:9, 2:70, 2:78, 2:78, 2:80, 2:124,
3:8, 3:25, 3:27, 3:31, 3:62, 3:76, 3:181, 3:182, 3:433, 3:445
VARSPA 1:5 1:13, 2:34
VARTYP - 1:40
VD.CLP 3:47 3:105, 3:111
VD.OFF 3:47
VDH - 1:102, 3:243, 3:266, 3:270, 3:318
VDHBIT 3:46 3:46
VDHCOD - 3:7
VDHLIN - 3:47
VDHP4 3:74
VDHSW 3:1 3:5, 3:46, 3:432
VDSO 3:17
VECTOR - 2:7
VFINDE 3:114 3:113, 3:113, 3:114
VHA 3:1 3:5, 3:39, 3:246, 3:269, 3:293, 3:294, 3:369, 3:369, 3:369,
3:417, 3:432
VHACOD - 3:7, 3:39, 3:440, 3:444
VHALEN 3:39 3:435, 3:440, 3:441
VHALIS 3:433 3:433, 3:437, 3:442, 3:442, 3:442, 3:443, 3:443, 3:444
VHALOK - 3:417, 3:433, 3:437, 3:437, 3:440, 3:440, 3:442, 3:442
VHAMAP 3:435 3:434, 3:435, 3:444
VHAOSE 3:433 3:440, 3:440
VHAREL 3:440 3:443, 3:443
VHARIX 3:433 3:440, 3:440
VHARXI 3:433 3:440, 3:440
VHASER 3:39 3:369, 3:440, 3:440
VHATAB 3:39 3:39, 3:435, 3:441, 3:441
VHAVAR - 3:7, 3:433, 3:445
VHINIT 3:433 3:440, 3:440, 3:441
VHINVT 3:433 3:433, 3:437, 3:437, 3:440, 3:440, 3:441, 3:442, 3:442, 3:442,
3:442, 3:442, 3:442, 3:442, 3:442, 3:443
VHLCHK 3:444 3:440, 3:443, 3:444

VHMCHK 3:444 3:440,3:444,3:444
 VHPIDO 3:24
 VLCODE - 3:8
 VLLOC 3:7
 VLST 1:5 1:5,1:6
 VMAP 3:29 3:29,3:418,3:423
 VMAPCH 3:418 3:418,3:419
 VMAPL 3:29 3:414
 WAIT 2:22,3:334 2:22,3:353
 WAITPC 3:378 3:334,3:378
 WAITSP 3:23 3:334,3:334
 WAITW 3:334 3:377
 WAKEUP 2:22 2:22
 WARM 3:7,3:7 3:7,3:7,3:7,3:40,3:125,3:240,3:262,3:265,3:268,3:270,
 3:287,3:305,3:315,3:316,3:321,3:445
 WATCH 3:25 3:400,3:401,3:415
 WATCHO 3:25 3:117,3:117
 WATCH1 2:124 2:128,3:117,3:401,3:401,3:415
 WATCH2 2:124 2:128,3:117,3:401,3:401,3:415,3:415
 WATCHH 3:25 3:399,3:400,3:415
 WATCHS 2:124 2:128,3:400,3:401,3:415
 WATM1 2:124 2:128,3:117,3:401
 WATM2 2:124 2:128,3:117,3:401
 WDGTIM 2:70 2:70,2:70,3:359,3:359
 WDIS 1:22 1:45,1:46,1:46,1:62,1:63,1:63,1:63,1:112
 WE - 2:7,2:7,2:7,2:7
 WERE - 2:7
 WHCTR 3:37 3:51,3:52,3:238,3:249,3:286,3:385
 WHEORB 3:54 3:54,3:85,3:159,3:164,3:168,3:236,3:255,3:276
 WHEORM 3:54 3:100,3:100,3:147
 WHERE 3:37 2:7,2:7,3:50,3:50,3:50,3:51,3:51,3:53,3:53,3:54,3:56,
 3:57,3:141,3:169,3:216,3:217,3:238,3:239,3:249,3:255,
 3:286,3:286,3:332,3:332,3:332,3:332,3:338,3:385,3:388,
 3:390,3:391,3:392
 WHEREV - 2:7
 WHF2H 3:37 3:85,3:147,3:159,3:351,3:352,3:355,3:356,3:357,3:357,
 3:376,3:376
 WHH2V 3:37 3:338
 WHHI 3:37 3:87,3:241,3:249,3:249,3:250,3:255,3:255,3:257,3:259,
 3:262,3:263,3:276,3:276,3:300,3:307
 WHHTP 3:37
 WHI2M 3:37 3:100,3:100,3:100,3:141,3:141,3:147,3:148,3:150,3:169,
 3:170,3:236
 WHIH 3:37 3:91,3:270,3:283,3:286,3:286,3:290,3:297,3:322,3:322,
 3:324,3:338
 WHM2I 3:37 3:85,3:154,3:155,3:156,3:157,3:158,3:159,3:164,3:167,
 3:168,3:168
 WHRUT 3:37 3:164,3:216,3:217,3:218,3:219,3:226,3:229
 WHTPH 3:37
 WHTSK 3:37 3:100,3:100,3:168,3:232,3:233,3:235,3:236,3:255,3:276,
 3:279,3:283,3:286,3:286
 WHV2H 3:37 3:333
 WMLOCK - 1:28,1:56,1:56,1:56,1:56
 WOPS 1:50 1:8,1:8,1:8,1:8,1:13,1:33,1:34,1:34
 WORDP 3:344 3:342,3:342,3:344,3:344
 WORDS 1:17 1:6,1:17,1:20,1:23,1:27,1:27,1:30,1:30,1:30,1:31,1:32,
 1:36,1:36,1:37,1:37,1:37,1:38,1:38,1:40,1:40,1:43,
 1:43,1:47,1:47,1:47,1:48,1:56,1:63,1:66,1:67,1:74,

1:78, 1:79, 1:79, 1:79, 1:79, 1:79, 1:79, 1:79, 1:80, 1:81,
1:82, 1:84, 1:85, 1:85, 1:88, 1:88, 1:90, 1:94, 1:94, 1:94,
1:94, 1:96, 1:105, 1:107, 1:107, 1:109, 1:112, 1:112, 1:116,
2:38, 2:39, 2:40, 2:43, 2:63, 2:63, 2:70, 2:96, 2:106,
2:106, 2:109, 2:115, 2:126, 3:9, 3:9, 3:23, 3:27, 3:27, 3:30,
3:30, 3:30, 3:40, 3:40, 3:40, 3:40, 3:43, 3:43, 3:43, 3:43,
3:43, 3:43, 3:43, 3:43, 3:43, 3:43, 3:43, 3:43, 3:43, 3:49,
3:57, 3:64, 3:65, 3:69, 3:69, 3:69, 3:71, 3:71, 3:72, 3:72,
3:72, 3:74, 3:78, 3:78, 3:79, 3:81, 3:81, 3:89, 3:94, 3:94,
3:95, 3:96, 3:96, 3:96, 3:96, 3:96, 3:98, 3:98, 3:98,
3:99, 3:99, 3:99, 3:99, 3:99, 3:102, 3:103, 3:112, 3:114, 3:118,
3:127, 3:128, 3:134, 3:144, 3:146, 3:151, 3:162, 3:176, 3:176,
3:176, 3:178, 3:179, 3:179, 3:182, 3:187, 3:187, 3:188, 3:189,
3:189, 3:191, 3:192, 3:194, 3:194, 3:195, 3:195, 3:195, 3:195,
3:195, 3:195, 3:195, 3:197, 3:197, 3:197, 3:197, 3:197, 3:197,
3:200, 3:201, 3:202, 3:204, 3:208, 3:208, 3:208, 3:208, 3:209,
3:210, 3:211, 3:211, 3:211, 3:211, 3:223, 3:224, 3:225,
3:225, 3:226, 3:228, 3:233, 3:242, 3:243, 3:243, 3:244, 3:244,
3:247, 3:247, 3:252, 3:265, 3:269, 3:269, 3:282, 3:293, 3:293,
3:293, 3:294, 3:301, 3:301, 3:307, 3:319, 3:319, 3:319, 3:319,
3:319, 3:329, 3:329, 3:335, 3:336, 3:336, 3:344, 3:347, 3:352,
3:352, 3:359, 3:362, 3:362, 3:371, 3:373, 3:373, 3:382, 3:386,
3:386, 3:386, 3:386, 3:388, 3:396, 3:396, 3:396, 3:399, 3:400,
3:405, 3:407, 3:412, 3:415, 3:415, 3:422, 3:437, 3:440, 3:440,
3:442, 3:442, 3:442, 3:442, 3:443

WPCINT 1:26 1:26, 1:63
WS 1:51 1:13, 2:70, 3:122, 3:210, 3:210
WSLEEP 1:45 1:49, 1:54, 1:60, 1:65, 1:66, 1:74, 1:76, 1:78, 1:79, 1:80,
1:118, 2:48, 2:51, 2:54, 2:56, 2:57, 2:60, 2:65
WSLSP 1:22 1:45, 1:49, 1:63
WSPINT 1:26 1:63
WST 1:51 1:13, 1:32, 1:35, 1:35, 1:36, 1:38, 1:39, 1:50, 1:50, 1:50,
1:50, 1:50, 1:53, 1:74, 1:75, 1:84, 1:119, 2:52, 2:55, 2:127,
3:49, 3:99, 3:212
WSTAGE 1:22 1:45, 1:46, 1:46, 1:47, 1:62, 1:63, 1:63, 1:63
WSTCOM 1:51 1:47
WSTINI 1:50 1:86, 1:94
WSUB 1:51 1:51, 1:51
WT - 3:23
WTSTAC - 3:334
X - 2:17, 2:17, 2:17, 2:17, 2:17, 2:17, 2:29, 2:29, 2:29, 2:29,
2:29, 2:29, 3:5
XBTC 3:320 3:320
XCMOVE 3:238 3:239, 3:239
XIHTC 3:317 3:317
XSIOIN 1:100 1:99, 1:100, 1:105, 3:41, 3:58
YES - 2:6
Z - 2:129, 3:42, 3:42, 3:125, 3:316
\$ 2:17, 2:29 1:5, 2:6, 2:17, 2:17, 2:17, 2:17, 2:21, 2:21, 2:27,
2:29, 2:29, 2:29, 2:29, 2:30, 2:33, 2:33, 3:72
\$%PLST - 2:27
\$ADDPA 1:5 1:5, 1:5, 1:5, 1:5
\$APPLY - 1:6, 1:6, 1:6, 1:6, 1:6, 1:12, 2:6, 2:6, 2:7
\$ARGTS 2:17, 2:29 2:17, 2:29
\$CLKFL 2:27, 2:27
\$CLOCK 2:27 2:30, 2:30
\$DOPAG - 2:6, 2:7
\$DOPAT 1:11 1:11, 1:11, 1:11, 1:11, 1:11, 1:11, 1:11, 1:11, 1:11,

1:11,1:11,1:11,1:11,1:11,1:11,1:11,1:11,1:11,1:11,
 1:11,1:11,1:11,1:12,2:81,3:35,3:64,3:65,3:69,3:69,
 3:71,3:71,3:72,3:72,3:72,3:74,3:74,3:81,3:81
 \$DOPPA - 2:4,2:4,2:7,2:7,2:16,2:28
 \$ENTER - 1:5,2:4,2:4,2:4,2:4,2:4,2:4,2:6,2:6,2:7,2:7,2:8,2:8,
 2:27
 \$EXIT 2:19,2:19,2:31,2:31 2:19,2:27
 \$FAST 2:30
 \$FCBAR 2:15,2:15 2:15,2:15,2:15,2:15,2:16,2:16
 \$FINCH 1:12
 \$FINIM 3:35 3:445
 \$FINKE 1:12 1:119
 \$FINST 1:11 1:119,2:138,3:445
 \$FIXCH 1:12 1:12
 \$IFDF1 - 2:7
 \$INIT 2:4 2:9
 \$INITR 1:5
 \$IR1 2:7 2:6,2:6
 \$IROUT 2:6
 \$ITABL 2:7 2:136
 \$LCCHK 2:16 2:15
 \$LIST - 1:5,1:5,1:5,1:5,2:4,2:4,2:4,2:6,2:6,2:7
 \$LSCHK 2:28 2:27
 \$MAPCD - 3:125,3:125,3:125,3:185,3:316,3:316
 \$MAPCH 2:4 2:4
 \$MAPPK - 3:125,3:125,3:125,3:125,3:125,3:212,3:217,3:217
 \$MAPRT - 3:125,3:125,3:125,3:185,3:212,3:217,3:217,3:316,3:316
 \$NO 2:17,2:29
 \$OPT 2:17,2:29 2:15,2:27
 \$PGKEY 1:5,1:5 1:5,1:5
 \$PHEAD 2:16 2:16,2:16
 \$PID 2:27 2:27,2:27
 \$PID1 2:6 2:6
 \$PINIT 2:6 2:7,2:8
 \$RATE 2:15 2:16
 \$RINIT 2:6 2:7,2:8
 \$SHEAD 2:30 2:28,2:28
 \$SLOW 2:30
 \$STACK 2:15,2:15,2:15 2:15,2:15,2:15,2:16
 \$TIMER 1:5
 \$TITLE 3:5 3:5
 \$XINIT 2:4 2:4,2:4
 % - 1:4,2:15,2:23
 %%PHYS - 2:7,2:7
 %3SPAC 2:112 2:96,2:112,2:113,3:65,3:70,3:71,3:72,3:72
 %CCHN 2:27 2:30
 %CNT 1:6,1:6,1:6,1:6 1:6,1:6,1:6,1:6
 %CODE - 1:6,1:6,1:6
 %CR 1:18
 %DOT 2:16,2:28 2:16,2:28
 %DR 1:18 1:34
 %FLG 1:5,2:7,2:17,2:17,2:17,2:29,2:29,2:29 1:5,2:7,2:7,2:17,2:29
 %ICHAI 2:7 2:7,2:7
 %IKEY - 2:4,2:4
 %IKEY3 2:4
 %IKEY4 2:5
 %IKEY5 2:5
 %ILST 2:4,2:6 2:4,2:4,2:4,2:4,2:4,2:4,2:6

%IMAP - 2:4,2:4
%IMAP3 2:4
%IMAP4 2:4
%IMAP5 2:4
%INI 1:5,1:5 1:5,1:5
%IPAGE - 2:7,2:7
%KEY - 2:4,2:4,2:4
%LIM 1:5 1:5,1:5
%M 2:4 2:4
%MAP - 2:16,2:16,2:28,2:28
%MAPO - 1:12,1:48,1:49,1:56,1:56,1:91,1:91,1:92,1:92,2:16,
2:28,2:51,2:64,2:64,2:65,2:81,2:126,2:131,2:131,3:93,
3:127,3:326,3:334
%MAP1 - 1:34,1:34,1:39,1:47,1:49,1:58,1:65,1:66,1:67,1:88,
1:88,2:4,2:44,2:44,2:49,2:50,2:51,2:54,2:63,2:81,2:126,
3:49,3:422,3:422
%MAP2 - 1:47,1:47,1:87,2:4,2:49,2:50,2:50,2:126,2:128,2:128,
2:128,2:129,3:49,3:50,3:50,3:52,3:54,3:54,3:56,3:56,
3:57,3:85,3:85,3:91,3:247,3:259,3:260,3:270,3:278,
3:279,3:285,3:288,3:290,3:290,3:297,3:299,3:328,3:330,
3:335,3:401,3:401,3:401,3:410,3:410,3:418,3:419,3:420,
3:423,3:423
%MAP3 - 1:48,1:49,1:51,1:56,1:91,2:4,2:39,2:50,2:81,2:126,
3:259,3:329,3:420
%NCODE 1:6 1:11,1:11,3:93,3:127,3:423
%NDOT 2:7,2:7 2:7,2:7,2:7
%NOVAR 1:6 1:11,1:51,1:72
%NSPAR 1:6 1:6,1:11
%NTYP 2:7 2:7,2:7
%NTYPE - 2:7,2:7
%NVARS 1:6 1:6,1:11
%OPTV - 1:6
%PHY 1:5 1:5,1:5
%PLST 2:4,2:6 2:6,2:27
%RADIX 2:15,2:15 1:4,2:15
%RLST 2:4,2:7 2:6,2:6,2:6,2:7,2:7,2:7,2:8,2:8
%SR 1:18 1:34
%SRADI 1:4,2:15 1:4,2:15
%TAB 1:5,1:5 1:5,1:5
%TOTAL - 1:4
%VARS - 1:6
.INIT 2:15 2:18
.VARS - 3:35
.FOO 1:28 1:28,1:28
.INSER - 1:1,1:9,1:15,1:99,2:1,2:10,2:24,2:34,2:71,2:123,3:8,
3:38,3:48,3:61,3:75,3:82,3:92,3:116,3:126,3:130,3:140,
3:172,3:180,3:230,3:240,3:325,3:340,3:379,3:432,3:432
.LALL - 1:23
.XALL - 1:23

| rap (Source/Page) | Loc: | Description |
|----------------------|-------|---|
| 1 (STAGEK/LCode) | 426: | Unexpected QUIT--mem ref fail (H) - page 31 |
| 3 (STAGEC/Re1Code) | 5338: | Ctd STAGE mem mgmnt--OK if Strtup/Rstrt (H) - page 51 |
| 4 (STAGEK/LCode) | 58C: | RTC stppd--switched to new RTC (H) - page 35 |
| 5 (STAGEK/LCode) | A74: | Local Kernel Cksum Broken-FATAL to Proc (H/S) - page 52 |
| 6 (STAGEK/LCode) | 9DA: | Unexpctd intrpt (Poss RESET/ATTN) (H) - page 50 |
| 7 (STAGEK/Re1Code) | 44AE: | BBC map fail (Poss bad coupler) (H) - page 82 |
| A (STAGEC/Re1Code) | 5570: | Jiffy clock stopped (H) - page 63 |
| B (STAGEK/LCode) | 906: | Sys missed a tick (H) - page 48 |
| C (STAGEK/LCode) | E86: | Quit in cksum param (H) - page 66 |
| D (STAGEK/LCode) | E5C: | Quit during cksuming (H) - page 66 |
| E (STAGEC/Re1Code) | 5036: | Stge LC: Local code cksum broken (H/S) - page 36 |
| F (STAGEC/Re1Code) | 510A: | Stage MM: Not Enough Memory (H) - page 40 |
| 10 (STAGEK/Re1Code) | 4260: | Stge vars area quit--12 TRAP means fix (H) - page 76 |
| 11 (STAGEK/LCode) | 8C6: | Lost our comm pg (H/S) - page 47 |
| 13 (STAGEK/Re1Code) | 444A: | QUIT on BCM coupler *** CALL BBN MAINT. *** (H) - page 81 |
| 14 (STAGEK/LCode) | 66C: | Hung on invld s/w lock - page 38 |
| 15 (STAGEK/LCode) | A8C: | Stge LK: can't find a clk - page 53 |
| 16 (STAGEK/LCode) | 4BC: | Rmt pwr fail E/F Bus (NOT PROCS) (H) - page 23 |
| 17 (STAGEK/LCode) | 588: | Can't find an RTC(CALL MAINT if RSTRT fails)(H) - page 35 |
| 1A (STAGEK/Re1Code) | 44EC: | Stge CD:Bad Proc Id (H/S) - page 84 |
| 1B (STAGEK/Re1Code) | 467A: | B1k Trans Timeout (proc trb1) - page 87 |
| 1C (STAGEK/Re1Code) | 4844: | BLT Target proc not in tbl - page 91 |
| 1D (STAGEK/Re1Code) | 48D6: | Non-existant proc in b1t?? (S) - page 92 |
| 1E (STAGEK/Re1Code) | 48FC: | No I/O bus for BBC (H) - page 93 |
| 1F (STAGEK/Re1Code) | 43EE: | Broken Coupler *** CALL BBN MAINT. *** (H) - page 80 |
| 20 (STAGEK/Re1Code) | 49E4: | BBC fail (CALL MAINT if 1E TRAP occurs) (H) - page 95 |
| 21 (STAGEK/LCode) | 562: | Proc RSTRTD after Pwr Fail (H)- page 35 |
| 22 (STAGEK/LCode) | 55A: | Proc Pwr Fail (H) - page 35 |
| 23 (STAGEK/LCode) | 56C: | illeg level 4 intrpt (H) - page 35 |
| 24 (STAGEK/Re1Code) | 4226: | Stge vars mem fail *** CALL BBN MAINT. *** (H) - page 75 |
| 25 (STAGEC/Re1Code) | 51D6: | Spare pg cksum differs (H/S) - page 44 |
| 27 (STAGEC/Re1Code) | 53E2: | Solid mem parity err *** CALL BBN MAINT *** (H) - page 55 |
| 28 (STAGEK/LCode) | B3A: | No usble comm mem (MEM PROB. CALL MAINT) (H) - page 55 |
| 2C (STAGEC/Re1Code) | 5538: | Quit retries ok on 2nd mem ref (H) - page 62 |
| 42 (OPSYS/LCode) | 1064: | got illegal pid value - page 125 |
| 50 (STAGEC/Re1Code) | 5648: | S/W watchdog timer expired} - page 70 |
| CO (DDT/DDTCode) | 416A: | DDT changed psbs (VISTAR using new BUS) - page 85 |
| 100 (FAKREL/FakCode) | 591C: | IMP reinit (H) - page 421 |
| 101 (FAKREL/FakCode) | 509C: | smashed buf ptr--may cause retrans - page 387 |
| 102 (CONFIG/Re1Code) | 57D2: | Changing buffer page allocation - page 98 |
| 102 (FAKREL/FakCode) | 57CO: | Changing buf pg alloc - page 417 |
| 103 (CONFIG/Re1Code) | 5B68: | Swapping to F device - page 112 |
| 104 (STO/LCode) | 1C7E: | lock timed out - page 128 |
| 108 (FASTTO/Warm) | 43B2: | Main clock stppd swtchd to bkup RTC (H) - page 119 |
| 10A (FASTTO/Warm) | 43D4: | no working backup RTC (H) - page 119 |
| 10B (FAKREL/FakCode) | 5544: | IMP num invld (CK RTC SWTCHS) (H) - page 408 |
| 10C (CONFIG/Re1Code) | 5834: | RTC lost--assoc w/traps 10A and 108 (H) - page 102 |
| 10D (FAKREL/FakCode) | 5578: | Node ovrrhtng--R2=BUS R4=TEMP (H) - page 408 |
| 10E (IMPDDE/DDTCode) | 4CAA: | Invld attmpt to use Xpatch - page 63 |
| 10F (IMPDDE/DDTCode) | 4CDA: | Invld attmpt to enable ovride - page 64 |
| 110 (CONFIG/Re1Code) | 5A8G: | modem lost on Prim bus CALL MAINT (H) - page 110 |
| 111 (CONFIG/Re1Code) | 5A0D: | Host lost on Prim buss CALL MAINT (H) - page 111 |
| 112 (CONFIG/Re1Code) | 5B7A: | spare interface disappeared CALL MAINT (H) - page 112 |
| 114 (CONFIG/Re1Code) | 5A90: | swapping modem interfaces CALL MAINT (H) - page 110 |
| 115 (CONFIG/Re1Code) | 5AD8: | swapping host interfaces CALL MAINT (H) - page 111 |
| 203 (CONFIG/Re1Code) | 59F8: | 2 interfaces, one device (CALL MAINT) - page 107 |
| 205 (CONFIG/Re1Code) | 59FO: | Dbld PID intfc differs CALL MAINT (H) - page 107 |

207 (CONFIG/RelCode) 5A56: BLDBLK: dynamic blocks area full - page 109
208 (FAKREL/LCode) 3D2A: free list in loop - page 389
209 (FAKREL/LCode) 3D3A: lost the free list - page 389
20A (IMPSUB/LCode) 1448: threw away free list tail - page 52
281 (FAKREL/FakCode) 50CC: Recovd a timed-out buf - page 388
2A2 (IMPSUB/LCode) 1576: buffer ownership error - page 56
2C2 (FAKREL/LCode) 3D12: free list buf err--WHERE nonzero - page 389
2C8 (LOCAL/LCode) 177C: ringc overflow in restart - page 82
2C9 (FAKREL/DDTCode) 5622: ring structure broken in timeout - page 395
2E1 (IMPSUB/LCode) 1438: free list error, non-zero where - page 52
2E3 (IMPSUB/LCode) 1356: tried to flush non-buffer - page 49
2E5 (IMPSUB/LCode) 1364: tried to flush non-owned buffer - page 49
2FO (IMPSUB/LCode) 1582: fixed half-empty queue - page 56
300 (VHA/DDTCode) 5764: VHAREL: finished VHALIS recomputation
301 (VHA/DDTCode) 5684: VHAREL: Detected VHA table error
302 (VHA/LCode) 3E76: IHVHA: No virtual address found
303 (VHA/LCode) 3EA8: TSKVHA: No virtual address this source
304 (VHA/DDTCode) 56BE: VHAREL: Too many VHA numbers
305 (VHA/DDTCode) 56AC: VHAREL: VHA IMP number too big
3C0 (FAKREL/FakCode) 5A4A: Imp going down (OPRTR INIT) - page 427
3C1 (FAKES/FakCode) 4866: Neighbor IMP wants a reload - page 353
3C2 (FAKES/FakCode) 481E: Flushing Reload Packet (NO BUF SP) - page 352
3F8 (FAKSUB/FakCode) 41B6: Host wanted a buffer - page 327
3F9 (FAKSUB/FakCode) 41AC: No host block - page 327
3FA (FAKSUB/FakCode) 43C0: Host sending a buffer - page 334
3FB (FAKSUB/FakCode) 446A: Host sending leader - page 336
3FC (FAKSUB/FakCode) 4288: Host wanted a leader - page 330
3FD (FAKSUB/FakCode) 427E: No host block? - page 330
3FE (FAKSUB/FakCode) 4478: No host block? - page 336
3FF (FAKSUB/FakCode) 4490: Bad buffer from IH - page 336
401 (MODEM/LCode) 2166: Modem bad end ptr *** CALL MAINT *** (H) - page 155
402 (MODEM/LCode) 208A: modem input quit (INPUT LOST) - page 153
403 (MODEM/LCode) 2162: Modem input short (BAD DATA INPUT LOST) - page 155
404 (MODEM/LCode) 1D5E: modem output got quit - page 140
405 (MODEM/LCode) 2020: Start pointer write failed - page 152
406 (MODEM/Warm) 4824: Bad cksum in routing update - page 144
407 (ROUTE/Warm) 4D1C: bad update checksum - page 184
409 (CONFIG/RelCode) 5AC0: scrambled modem parameter block - page 110
40A (MODEM/LCode) 22F8: bad sentq - page 168
40B (MODEM/LCode) 1D7A: lost SNDING buffer - page 140
40C (UPDWN/Warm) 469C: Master line died - page 134
410 (MODEM/LCode) 2178: Modem s/w failure - page 156
411 (MODEM/LCode) 1E72: broken cksum on retransmission - page 147
412 (MODEM/LCode) 1E46: 64 retransmissions: killed line - page 148
413 (MODEM/LCode) 1EAE: 32 retransmissions: discard packet - page 148
414 (MODEM/LCode) 232E: Unexpected ACK - page 168
415 (UPDWN/Warm) 4522: Too many modem h/w cksum errors - page 130
420 (TASK/LCode) 2490: No rte for task pkt - page 231
421 (TASK/LCode) 2468: Pkt w/discard bit discarded - page 230
4C1 (MODEM/LCode) 2104: filling buffer error - page 154
4C8 (FAKREL/DDTCode) 54A4: modem state mismatch - page 380
500 (ROUTE/Warm) 53D2: SPF error forced restart - page 208
503 (ROUTE/LCode) 2420: routing broken queue - page 214
504 (ROUTE/Warm) 55EC: Buffer no longer owned by routing - page 215
505 (ROUTE/Warm) 55FE: Caller's bit not on - page 215
506 (ROUTE/Warm) 561C: Rupq buffer missing - page 216
507 (ROUTE/Warm) 588C: Retrans w/bad length or IMP - page 223
508 (ROUTE/Warm) 598E: Rupqct wrong - page 227
509 (ROUTE/Warm) 59B0: Recovered unused buffer - page 227
50A (ROUTE/Warm) 54A2: Queuing packet for no one - page 2

| | |
|----------------------|---|
| 555 (ROUTE/Warm) | 5514: Queue count too high - page 210 |
| 557 (ROUTE/Warm) | 54EC: Queue count check - page 210 |
| 5C2 (MODEM/LCode) | 21EE: Suddenly looped line - page 157 |
| 5C3 (TASK/LCode) | 24BA: Flushing pkt for dead IMP - page 231 |
| 5C5 (MODEM/Warm) | 49DE: Master/slave mismatch - page 162 |
| 5C6 (MODEM/Warm) | 4976: Neighbor IMP number changed - page 160 |
| 600 (WARM/LCode) | 3398: No message for incm or incq - page 280 |
| 602 (WARM/LCode) | 2A8C: Host input err (Numerous traps call maint) - page 249 |
| 603 (WARM/LCode) | 2COA: Host input err in leader - page 254 |
| 604 (LOCAL/LCode) | 1904: Host output err - page 87 |
| 605 (WARM/LCode) | 34AC: No reas blk for alloc 8-pkt msg - page 283 |
| 606 (WARM/LCode) | 3372: No alloc to give back - page 280 |
| 607 (WARM/LCode) | 338A: Incq/incm w/gvb, but no alloc to gb - page 280 |
| 608 (WARM/LCode) | 344E: Rstate violation - page 282 |
| 60A (WARM/LCode) | 3294: Reply lost-no space - page 277 |
| 60B (WARM/Warm) | 5DB6: Start ptr write failed - page 321 |
| 611 (WARM/LCode) | 2E48: illegal message blk in hi - page 262 |
| 619 (FAKREL/FakCode) | 54BO: ihwq is a mess - page 405 |
| 61A (CONFIG/Re1Code) | 5974: BASE/MBLKS wrong for HI/IH - page 106 |
| 628 (CONFIG/Re1Code) | 5B44: scrambled host parameter block - page 111 |
| 642 (WARM/LCode) | 3110: No trnb1k for alloc - page 273 |
| 643 (WARM/LCode) | 31B6: No trnb1k for RFNM or dead RFNM - page 275 |
| 644 (WARM/Warm) | 5C6E: Res rep when not resetting - page 285 |
| 645 (WARM/LCode) | 3298: Got msg w/illegal pkt code 13 - page 277 |
| 646 (WARM/LCode) | 31DC: No trnb1k for inc RFNM - page 275 |
| 647 (WARM/DDTCode) | 4F84: No trnb1k for inc query - page 303 |
| 650 (WARM/LCode) | 317E: Got a duplicate Allocate 1 - page 274 |
| 658 (FAKREL/FakCode) | 53EE: Bad local Host in message block - page 402 |
| 681 (FAKREL/DDTCode) | 5596: Flushing an old trnb1k - page 393 |
| 683 (FAKREL/DDTCode) | 55B6: requeueing trnb1k for IH - page 393 |
| 684 (FAKREL/DDTCode) | 55E4: trnb1k/tmb1k mismatch - page 393 |
| 68A (WARM/LCode) | 2C78: Host b1kd awaiting free buffer - page 255 |
| 68C (WARM/LCode) | 2E3C: Host b1kd awaiting mes num or b1k - page 259 |
| 68D (WARM/LCode) | 29D8: Host b1kd awaiting alloc - page 247 |
| 68E (WARM/LCode) | 2BF8: Host b1kd awaiting task - page 253 |
| 68F (WARM/LCode) | 29CO: Host b1kd requesting alloc - page 247 |
| 690 (WARM/Warm) | 5AD4: Host b1kd awaiting trnb1k - page 264 |
| 691 (WARM/LCode) | 2A76: Host b1kd middle of 8-pkt - page 248 |
| 692 (WARM/LCode) | 2AC6: Task b1kd inc msg - page 249 |
| 698 (WARM/Warm) | 5E6C: Bad rm blk for mes on host q - page 322 |
| 699 (WARM/LCode) | 2BD4: Lost buffer in hi2tsk - page 253 |
| 69A (WARM/LCode) | 2BF4: Bk b1kd awaiting task - page 253 |
| 6AO (WARM/LCode) | 2A90: Error during host input data - page 249 |
| 6C2 (WARM/Warm) | 5D92: Bad buffer on host queue - page 321 |
| 6C6 (WARM/LCode) | 29C6: Clbbrd hisp requesting alloc - page 247 |
| 6C7 (WARM/LCode) | 29DC: Hisp clbbrd in pkt - page 247 |
| 6C8 (WARM/LCode) | 2AD6: Bad hisp for bad message - page 249 |
| 6CA (WARM/LCode) | 317A: Bad trnb1k buffer - page 274 |
| 6D0 (WARM/LCode) | 3644: Bad buffer in t2h - page 288 |
| 6D8 (WARM/LCode) | 2F64: ih lost a trnb1k - page 269 |
| 6E8 (WARM/Warm) | 5D8E: Bad ih queue struct - page 321 |
| 6FO (WARM/LCode) | 2DC6: HI bad packet length - page 258 |
| 7C0 (WARM/LCode) | 3186: Got an Out-of-range - page 275 |
| 7C1 (WARM/LCode) | 30E2: Sending out-of-range - page 272 |
| 7C2 (WARM/LCode) | 32F2: No free rm blk - page 278 |
| 7C4 (WARM/LCode) | 2ADA: dest died in hi - page 250 |
| 7C5 (WARM/LCode) | 30AE: Sending duplicate reply - page 272 |
| 7C6 (WARM/LCode) | 3306: Rcvd dup Get-a-block - page 278 |
| 7C7 (WARM/DNTCode) | 4FCC: Sending inc query - page 303 |

7CA (WARM/LCode) 3460: No alloc for 1-pkt msg - page 282
FC8 (WARM/LCode) 2BOC: Host sent err w/id - page 250
FDO (WARM/LCode) 37CE: Na1 gone neg - page 293
FD8 (WARM/LCode) 35B4: Illegal rstate/type - page 286

| Lock (Source/Page) | Label: Description |
|-------------------------|---|
| A082 (STAGEK/Vars) | wmlock: memory test lock - page 28 |
| A08E (STAGEK/Vars) | s1f1k: locked copy (+2) of s1fptr - page 28 |
| A098 (STAGEK/Vars) | memory discovery consensus - page 28 |
| AOAA (STAGEK/Vars) | Common Kernel Discovery Consensus - page 28 |
| A25E (DDT/Vars) d2f1: | |
| A260 (DDT/Vars) f2d1: | |
| A262 (DDT/Vars) ttylok: | |
| A264 (DDT/Vars) ddtlok: | |
| A2EA (VARS/Vars) | clklok: Lock on RTC counters - page 25 |
| A3B6 (VARS/Vars) | 1tq: task queue lock - page 26 |
| A3C4 (VARS/Vars) | free: free buffer list - page 27 |
| A3C6 (VARS/Vars) | freend: end of free buffer list - page 27 |
| A3C8 (VARS/Vars) | nf: size of shared buffer pool plus minf - page 27 |
| A4A4 (VARS/Vars) | lockro: routing send buffers lock - page 27 |
| A4A6 (VARS/Vars) | cycle: timeout clock counters - page 27 |
| A4A8 (VARS/Vars) | trnlok: free transaction blocks lock - page 27 |
| A4AA (VARS/Vars) | messt: message number timeout non-lock - page 27 |
| A4B2 (VARS/Vars) | ringlk: restarter ring lock - page 28 |
| A4D4 (VARS/Vars) | tcgo: host wakeup lock - page 28 |
| A4D8 (VARS/Vars) | tbkgo: back host wakeup lock - page 28 |
| A506 (VARS/Vars) | stolok: slow timeout lock - page 28 |
| A550 (VARS/Vars) | conlok: configuration lock - page 29 |
| A5B0 (VARS/Vars) | rmlock: (and every D20) rcv mes block locks - page 31 |
| A930 (VARS/Vars) | tmlock: (and every D20) xmit mes block locks - page 31 |
| ACB0 (VARS/Vars) | reas blk lock (and every H10) - page 32 |
| AE44 (VARS/Vars) | Fake 0 DOZE lock - page 33 |
| AE46 (VARS/Vars) | Fake 0 WAIT lock - page 33 |
| AEC6 (VARS/Vars) | Fake 1 DOZE lock - page 33 |
| AEC8 (VARS/Vars) | Fake 1 WAIT lock - page 33 |
| AF48 (VARS/Vars) | Fake 2 DOZE lock - page 33 |
| AF4A (VARS/Vars) | Fake 2 WAIT lock - page 33 |
| AFCA (VARS/Vars) | Fake 3 DOZE lock - page 33 |
| AFCC (VARS/Vars) | Fake 3 WAIT lock - page 33 |
| B038 (VARS/Vars) | back host 0 (back5) lock - page 34 |
| B058 (VARS/Vars) | back host 1 (back7) lock - page 34 |
| B078 (VARS/Vars) | back host 2 (back9) lock - page 34 |
| B098 (VARS/Vars) | back host 3 (back6) lock - page 34 |
| BOCO (VARS/Vars) | hi host lock fake 0 - page 35 |
| B10E (VARS/Vars) | ih hardware lock fake 0 - page 35 |
| B11C (VARS/Vars) | ih software lock fake 0 - page 35 |
| B130 (VARS/Vars) | hi host lock fake 1 - page 35 |
| B17E (VARS/Vars) | ih hardware lock fake 1 - page 35 |
| B18C (VARS/Vars) | ih software lock fake 1 - page 35 |
| B1AO (VARS/Vars) | hi host lock fake 2 - page 35 |
| B1EE (VARS/Vars) | ih hardware lock fake 2 - page 35 |
| B1FC (VARS/Vars) | ih software lock fake 2 - page 35 |
| B210 (VARS/Vars) | hi host lock fake 3 - page 35 |
| B25E (VARS/Vars) | ih hardware lock fake 3 - page 35 |
| B26C (VARS/Vars) | ih software lock fake 3 - page 35 |
| B2A4 (DISPLAY/Vars) | dsplok: display variables lock - page 75 |
| B2C6 (ROUTE/Vars) | spfrtl: Lock on common SPF tables - page 180 |
| B2C8 (ROUTE/Vars) | rutlok: Lock on routing processing - page 180 |
| B3FC (VHA/Vars) vhalok: | Lock on VHA inverse translation table |
| BD52 (STAGEK/RelVars) | Common Bus Discovery Consensus - page 70 |
| BD5E (STAGEK/RelVars) | processor and bus coupler discovery consensus - page 70 |
| BD68 (STAGEK/RelVars) | bbclok: lock on bus coupler states - page 70 |
| BDBA (STAGEK/RelVars) | bltlok: Block transfer lock - page 71 |

BE30 (STAGEK/RelVars) Consensus for Local Checksum - page 72
BE38 (STAGEK/RelVars) memory configuration consensus - page 72
BE92 (STAGEK/RelVars) consensus for i/o discovery - page 72
BEA6 (STAGEK/RelVars) initialization consensus lock - page 72
BEEAE (PKCORE/RelVars) pkclok: lock on packet core parameters - page 98

Page Use Summary

From DDTCode 4000 DDT: 7, List: 82
To DDTCode 4236 DDT: 12, List: 90
From DDTCode 4236 DDT: 12, List: 91
To DDTCode 483E DDT: 30, List: 121
From DDTCode 483E HACCON: 1, List: 39
To DDTCode 4C42 HACCON: 2, List: 40
From DDTCode 4C42 HACCON: 7, List: 46
To DDTCode 4C46 HACCON: 8, List: 47
From DDTCode 4C46 IMPDDT: 2, List: 63
To DDTCode 4C62 IMPDDT: 2, List: 63
From DDTCode 4C62 IMPDDT: 2, List: 63
To DDTCode 4CC8 IMPDDT: 3, List: 64
To DDTCode 4CC8 IMPDDT: 3, List: 64
From DDTCode 4CC8 IMPDDT: 3, List: 64
From DDTCode 4CC8 IMPDDT: 3, List: 64
To DDTCode 4D3E IMPDDT: 4, List: 65
To DDTCode 4D3E IMPDDT: 4, List: 65
From DDTCode 4D3E IMPDDT: 4, List: 65
From DDTCode 4D3E IMPDDT: 4, List: 65
To DDTCode 4DE8 IMPDDT: 8, List: 69
To DDTCode 4DE8 IMPDDT: 8, List: 69
To DDTCode 4DE8 IMPDDT: 8, List: 69
From DDTCode 4DE8 IMPDDT: 8, List: 69
From DDTCode 4DE8 IMPDDT: 8, List: 69
From DDTCode 4DE8 IMPDDT: 8, List: 69
To DDTCode 4E68 IMPDDT: 10, List: 71
To DDTCode 4E68 IMPDDT: 10, List: 71
To DDTCode 4E68 IMPDDT: 10, List: 71
From DDTCode 4E68 IMPDDT: 10, List: 71
From DDTCode 4E68 IMPDDT: 10, List: 71
From DDTCode 4E68 IMPDDT: 10, List: 71
To DDTCode 4E84 IMPDDT: 11, List: 72
To DDTCode 4E84 IMPDDT: 11, List: 72
From DDTCode 4E84 IMPDDT: 11, List: 72
From DDTCode 4E84 IMPDDT: 11, List: 72
To DDTCode 4EA2 IMPDDT: 11, List: 72
To DDTCode 4EA2 IMPDDT: 11, List: 72
From DDTCode 4EA2 IMPDDT: 11, List: 72
From DDTCode 4EA2 IMPDDT: 11, List: 72
To DDTCode 4ED6 IMPDDT: 11, List: 72
To DDTCode 4ED6 IMPDDT: 11, List: 73
From DDTCode 4ED6 IMPDDT: 11, List: 73
From DDTCode 4ED6 IMPDDT: 12, List: 74
To DDTCode 4EFO IMPDDT: 12, List: 74
To DDTCode 4EFO DISPLAY: 6, List: 81
To DDTCode 4EFO IMPDDT: 12, List: 74
From DDTCode 4EFO IMPDDT: 12, List: 74
From DDTCode 4EFO DISPLAY: 6, List: 81
From DDTCode 4EFO DISPLAY: 6, List: 81
To DDTCode 4F16 DISPLAY: 6, List: 81
To DDTCode 4F16 DISPLAY: 6, List: 81
From DDTCode 4F16 DISPLAY: 6, List: 81
From DDTCode 4F16 DISPLAY: 6, List: 81
To DDTCode 4F1A LOCAL: 1, List: 83
From DDTCode 4F1A ROUTE: 46, List: 228
To DDTCode 4F54 ROUTE: 46, List: 228
To DDTCode 4F54 ROUTE: 46, List: 228

From DDTCode 4F54 WARM: 65, List: 305
To DDTCode 52DA WARM: 74, List: 314
From DDTCode 52DA WARM: 75, List: 315
To DDTCode 5338 WARM: 75, List: 315
To DDTCode 5338 WARM: 75, List: 315
From DDTCode 5338 WARM: 75, List: 315
From DDTCode 5338 WARM: 77, List: 317
To DDTCode 5382 WARM: 77, List: 317
From DDTCode 5382 WARM: 77, List: 317
To DDTCode 5424 WARM: 80, List: 320
From DDTCode 5424 WARM: 80, List: 320
To DDTCode 542C WARM: 80, List: 320
From DDTCode 542C WARM: 80, List: 320
To DDTCode 5440 WARM: 80, List: 320
From DDTCode 5440 FAKREL: 3, List: 382
To DDTCode 54C4 FAKREL: 3, List: 382
To DDTCode 54C4 FAKREL: 3, List: 382
From DDTCode 54C4 FAKREL: 3, List: 382
From DDTCode 54C4 FAKREL: 14, List: 393
To DDTCode 55F6 FAKREL: 16, List: 395
To DDTCode 55F6 FAKREL: 16, List: 395
From DDTCode 55F6 FAKREL: 16, List: 395
From DDTCode 55F6 FAKREL: 18, List: 397
To DDTCode 5626 FAKREL: 18, List: 397
To DDTCode 5626 FAKREL: 18, List: 397
From DDTCode 5626 FAKREL: 18, List: 397
From DDTCode 5626 VHA: 8, List: 440
To DDTCode 5768 VHA: 11, List: 443
To DDTCode 5768 VHA: 12, List: 444
From DDTCode 5768 VHA: 11, List: 443
From DDTCode 5768 VHA: 12, List: 444
To DDTCode 57A8 MAIN: 7, List: 445
From DDTVars 5B00 DDT: 3, List: 76
To DDTVars 5B3A DDT: 4, List: 78
From DDTVars 5B3A DDT: 5, List: 80
To DDTVars 5B3E DDT: 6, List: 81
From DDTVars 5B3E DISPLAY: 1, List: 76
To DDTVars 5B5A DISPLAY: 2, List: 77
From DDTVars 5B5A VHA: 1, List: 433
To DDTVars 5F80 VHA: 2, List: 434
From FakCode 40CO HACCON: 7, List: 46
To FakCode 40EO HACCON: 7, List: 46
From FakCode 40EO FAKSUB: 1, List: 326
To FakCode 40FO FAKSUB: 1, List: 326
From FakCode 40FO FAKSUB: 1, List: 326
To FakCode 40FC FAKSUB: 1, List: 326
From FakCode 40FC FAKSUB: 1, List: 326
To FakCode 4154 FAKSUB: 2, List: 328
From FakCode 4154 FAKSUB: 2, List: 328
To FakCode 41A2 FAKSUB: 3, List: 329
From FakCode 41A2 FAKSUB: 3, List: 329
To FakCode 4200 FAKSUB: 4, List: 330
From FakCode 4200 FAKSUB: 4, List: 330
To FakCode 42DA FAKSUB: 7, List: 333
From FakCode 42DA FAKSUB: 7, List: 333
To FakCode 42EA FAKSUB: 8, List: 334
From FakCode 42EA FAKSUB: 8, List: 334
To FakCode 42F6 FAKSUB: 8, List: 334
From FakCode 42F6 FAKSUB: 8, List: 334
To Fal .le 4340 FAKSUB: 9, List: 335

From FakCode 4340 FAKSUB: 9, List: 335
To FakCode 43A2 FAKSUB: 10, List: 336
From FakCode 43A2 FAKSUB: 10, List: 336
To FakCode 44C0 FAKSUB: 12, List: 338
To FakCode 44C0 FALES: 1, List: 341
From FakCode 44C0 FAKSUB: 12, List: 339
From FakCode 44C0 FALES: 1, List: 341
To FakCode 462C FALES: 7, List: 347
From FakCode 462C FALES: 7, List: 347
To FakCode 46FC FALES: 10, List: 350
From FakCode 46FC FALES: 10, List: 350
To FakCode 47F4 FALES: 13, List: 353
From FakCode 47F4 FALES: 14, List: 354
To FakCode 4832 FALES: 15, List: 355
From FakCode 4832 FALES: 15, List: 355
To FakCode 48DA FALES: 17, List: 357
From FakCode 48DA FALES: 18, List: 359
To FakCode 4918 FALES: 19, List: 360
From FakCode 4918 FALES: 20, List: 361
To FakCode 49BA FALES: 22, List: 363
From FakCode 49BA FALES: 22, List: 363
To FakCode 4E68 FAKREL: 2, List: 381
From FakCode 4E68 FAKREL: 2, List: 381
To FakCode 4EB2 FAKREL: 3, List: 382
From FakCode 4EB2 FAKREL: 3, List: 382
To FakCode 5106 FAKREL: 12, List: 391
From FakCode 5106 FAKREL: 13, List: 392
To FakCode 5140 FAKREL: 14, List: 393
From FakCode 5140 FAKREL: 16, List: 395
To FakCode 51B2 FAKREL: 18, List: 397
From FakCode 51B2 FAKREL: 18, List: 397
To FakCode 54C6 FAKREL: 29, List: 408
From FakCode 54C6 FAKREL: 29, List: 408
To FakCode 5A38 FAKREL: 49, List: 429
From FakCode 5A38 FAKREL: 49, List: 429
To FakCode 5AE8 VHA: 1, List: 433
To FakCode 5AE8 MAIN: 7, List: 445
From FakCode 5AE8 MAIN: 7, List: 445
From FakVars 5E00 FALES: 1, List: 341
To FakVars 5E12 FALES: 1, List: 341
From FakVars 5E12 FALES: 7, List: 347
To FakVars 5E1E FALES: 7, List: 347
From FakVars 5E1E FALES: 10, List: 350
To FakVars 5E2A FALES: 10, List: 350
From FakVars 5E2A FALES: 15, List: 355
To FakVars 5E36 FALES: 15, List: 355
From FakVars 5E36 FALES: 18, List: 359
To FakVars 5E44 FALES: 18, List: 359
From FakVars 5E44 FALES: 19, List: 360
To FakVars 5F88 FALES: 20, List: 361
From FakVars 5F88 FALES: 22, List: 363
To FakVars 5F8C FALES: 22, List: 363
From FakVars 5F8C FAKREL: 49, List: 429
To FakVars 5F96 FAKREL: 49, List: 429
From LCode 2F8 STAGEK: 9, List: 24
To LCode 3AO STAGEK: 13, List: 28
From LCode 3AO STAGEK: 15, List: 30
To LCode ED6 STAGEK: 53, List: 68
From LCode ED6 PKCORE: 4, List: 99

From LCode FOC MAIN: 2, List: 119
To LCode FOE MAIN: 2, List: 119
To LCode FOE MAIN: 2, List: 119
From LCode FOE STAGEC: 18, List: 64
From LCode FOE MAIN: 2, List: 119
To LCode F96 STAGEC: 19, List: 66
From LCode F96 DDT: 6, List: 81
To LCode FD2 DDT: 6, List: 81
To LCode FD2 DDT: 6, List: 81
From LCode FD2 DDT: 6, List: 81
From LCode FD2 DDT: 12, List: 90
To LCode 103C DDT: 12, List: 91
From LCode 103C DDT: 30, List: 121
To LCode 1064 OPSYS: 1, List: 124
From LCode 1064 OPSYS: 2, List: 125
To LCode 12BE MAIN: 1, List: 138
To LCode 12BE MAIN: 1, List: 138
From LCode 12BE HACCON: 3, List: 41
From LCode 12BE MAIN: 1, List: 138
To LCode 12DE HACCON: 4, List: 42
From LCode 12DE HACCON: 5, List: 43
To LCode 12EG HACCON: 5, List: 44
From LCode 12EG HACCON: 6, List: 45
To LCode 12EC HACCON: 7, List: 46
From LCode 12EC HACCON: 8, List: 47
To LCode 12EE HACCON: 8, List: 47
From LCode 12EE HACCON: 8, List: 47
To LCode 1308 HACCON: 8, List: 47
From LCode 1308 IMPSUB: 1, List: 49
To LCode 1314 IMPSUB: 1, List: 49
From LCode 1314 IMPSUB: 2, List: 50
To LCode 161A IMPDDT: 1, List: 62
From LCode 161A IMPDDT: 2, List: 63
To LCode 1622 IMPDDT: 2, List: 63
From LCode 1622 IMPDDT: 2, List: 63
To LCode 162A IMPDDT: 2, List: 63
From LCode 162A DISPLAY: 2, List: 77
To LCode 1636 DISPLAY: 2, List: 77
From LCode 1636 DISPLAY: 2, List: 77
To LCode 1750 DISPLAY: 6, List: 81
From LCode 1750 LOCAL: 1, List: 83
To LCode 17C8 LOCAL: 3, List: 85
From LCode 17C8 LOCAL: 4, List: 86
To LCode 188A LOCAL: 5, List: 87
From LCode 188A LOCAL: 5, List: 87
To LCode 18DA LOCAL: 6, List: 88
From LCode 18DA LOCAL: 6, List: 88
To LCode 1B06 CONFIG: 2, List: 93
From LCode 1B06 CONFIG: 2, List: 93
To LCode 1BBA CONFIG: 4, List: 95
From LCode 1BBA CONFIG: 10, List: 101
To LCode 1BEE CONFIG: 10, List: 101
From LCode 1BEE STO: 1, List: 127
To LCode 1CDC UPDWN: 1, List: 131
From LCode 1CDC UPDWN: 6, List: 136
To LCode 1CF4 UPDWN: 6, List: 136
From LCode 1CF4 UPDWN: 9, List: 139
To LCode 1D3A MODEM: 1, List: 141
From LCode 1D3A MODEM: 1, List: 141
To LCode 1DD6 MODEM: 2, List: 142

From LCode 1DD6 MODEM: 2, List: 142
To LCode 1E18 MODEM: 4, List: 144
From LCode 1E18 MODEM: 8, List: 148
To LCode 1F16 MODEM: 10, List: 151
From LCode 1F16 MODEM: 10, List: 151
To LCode 2200 MODEM: 18, List: 159
From LCode 2200 MODEM: 25, List: 167
To LCode 2382 DEL: 2, List: 173
From LCode 2382 DEL: 3, List: 175
To LCode 23C4 DEL: 4, List: 176
From LCode 23C4 ROUTE: 5, List: 185
To LCode 2414 ROUTE: 6, List: 186
From LCode 2414 ROUTE: 34, List: 216
To LCode 2436 ROUTE: 35, List: 217
From LCode 2436 TASK: 2, List: 232
To LCode 268A WARM: 1, List: 241
From LCode 268A WARM: 1, List: 241
To LCode 2E42 WARM: 22, List: 262
From LCode 2E42 WARM: 24, List: 264
To LCode 2EFA WARM: 25, List: 265
From LCode 2EFA WARM: 27, List: 267
To LCode 2F40 WARM: 28, List: 268
From LCode 2F40 WARM: 31, List: 271
To LCode 3560 WARM: 47, List: 287
From LCode 3560 WARM: 47, List: 287
To LCode 398E WARM: 60, List: 300
From LCode 398E WARM: 60, List: 300
To LCode 39DE WARM: 61, List: 301
From LCode 39DE WARM: 61, List: 301
To LCode 3BA4 WARM: 65, List: 305
From LCode 3BA4 WARM: 74, List: 314
To LCode 3BBC WARM: 75, List: 315
From LCode 3BBC WARM: 77, List: 317
To LCode 3BCA WARM: 77, List: 317
From LCode 3BCA WARM: 77, List: 317
To LCode 3BD4 WARM: 77, List: 317
From LCode 3BD4 WARM: 80, List: 320
To LCode 3BE2 WARM: 80, List: 320
From LCode 3BE2 WARM: 80, List: 320
To LCode 3BEC WARM: 81, List: 321
From LCode 3BEC FAKSUB: 1, List: 326
To LCode 3CO4 FAKSUB: 1, List: 326
From LCode 3CO4 FAKSUB: 1, List: 326
To LCode 3C10 FAKSUB: 1, List: 326
From LCode 3C10 FAKSUB: 2, List: 328
To LCode 3C26 FAKSUB: 2, List: 328
From LCode 3C26 FAKSUB: 3, List: 329
To LCode 3C3C FAKSUB: 3, List: 329
From LCode 3C3C FAKSUB: 4, List: 330
To LCode 3C52 FAKSUB: 4, List: 330
From LCode 3C52 FAKSUB: 7, List: 333
To LCode 3C68 FAKSUB: 7, List: 333
From LCode 3C68 FAKSUB: 8, List: 334
To LCode 3C7E FAKSUB: 8, List: 334
From LCode 3C7E FAKSUB: 8, List: 334
To LCode 3C8A FAKSUB: 8, List: 334
From LCode 3C8A FAKSUB: 9, List: 335
To LCode 3CA0 FAKSUB: 9, List: 335
From LCode 3CA0 FAKSUB: 10, List: 336

From LCode 3CB6 FAKSUB: 12, List: 338
To LCode 3CCC FAKSUB: 12, List: 339
From LCode 3CCC FALES: 13, List: 353
To LCode 3CE2 FALES: 13, List: 353
From LCode 3CE2 FALES: 17, List: 357
To LCode 3CF8 FALES: 17, List: 357
From LCode 3CF8 FAKREL: 12, List: 391
To LCode 3D40 FAKREL: 13, List: 392
From LCode 3D40 VHA: 2, List: 434
To LCode 3D98 VHA: 3, List: 435
From LCode 3D98 VHA: 3, List: 435
To LCode 3DD2 VHA: 5, List: 437
From LCode 3DD2 VHA: 5, List: 437
To LCode 3E26 VHA: 6, List: 438
From LCode 3E26 VHA: 6, List: 438
To LCode 3E7A VHA: 7, List: 439
From LCode 3E7A VHA: 7, List: 439
To LCode 3EBE VHA: 8, List: 440
From LCode 3EBE VHA: 11, List: 443
To LCode 3ED6 VHA: 11, List: 443
To LCode 3ED6 MAIN: 7, List: 445
From LCode 3ED6 MAIN: 7, List: 445
From LTVars 40 CONFIG: 2, List: 93
To LTVars 42 CONFIG: 2, List: 93
From LVars 50 STAGEK: 4, List: 19
To LVars 27E MAIN: 1, List: 138
To LVars 27E MAIN: 2, List: 119
To LVars 27E STAGEK: 8, List: 23
From LVars 27E TASK: 1, List: 231
From LVars 27E MAIN: 1, List: 138
From LVars 27E MAIN: 2, List: 119
To LVars 286 TASK: 2, List: 232
From LVars 286 FAKSUB: 1, List: 326
To LVars 288 MAIN: 7, List: 445
To LVars 288 FAKSUB: 1, List: 326
From LVars 288 MAIN: 7, List: 445
From Lc1Stk 288 STAGEK: 8, List: 23
To Lc1Stk 2F8 STAGEK: 9, List: 24
From PkgVars 5E80 ROUTE: 1, List: 181
To PkgVars 5E8A ROUTE: 1, List: 181
From PkgVars 5E8A ROUTE: 1, List: 181
To PkgVars 5EB4 ROUTE: 2, List: 182
From RelCode 4000 STAGEK: 53, List: 68
To RelCode 4166 STAGEK: 55, List: 70
From RelCode 4166 STAGEK: 58, List: 73
To RelCode 4A38 PKCORE: 1, List: 96
From RelCode 4A38 PKCORE: 4, List: 101
To RelCode 5016 MAIN: 2, List: 119
From RelCode 5016 MAIN: 2, List: 119
To RelCode 5020 STAGEC: 1, List: 35
To RelCode 5020 MAIN: 2, List: 119
To RelCode 5020 MAIN: 2, List: 119
From RelCode 5020 STAGEC: 1, List: 35
From RelCode 5020 STAGEC: 1, List: 35
From RelCode 5020 MAIN: 2, List: 119
To RelCode 559E STAGEC: 18, List: 64
From RelCode 559E STAGEC: 19, List: 66
To RelCode 5624 STAGEC: 21, List: 70
From RelCode 5624 STAGEC: 21, List: 70
To RelCode 5650 DDT: 3, List: 76

To RelCode 5650 MAIN: 1, List: 138
From RelCode 5650 HACCON: 8, List: 47
From RelCode 5650 MAIN: 1, List: 138
To RelCode 567C IMPSUB: 1, List: 49
From RelCode 567C CONFIG: 4, List: 95
To RelCode 57DA CONFIG: 9, List: 100
From RelCode 57DA CONFIG: 11, List: 102
To RelCode 5COC FASTTO: 1, List: 117
From RelCode 5COC FALES: 13, List: 353
To RelCode 5C10 FALES: 13, List: 353
From RelCode 5C10 FALES: 13, List: 353
To RelCode 5C50 FALES: 14, List: 354
From RelCode 5C50 FALES: 17, List: 357
To RelCode 5C52 FALES: 17, List: 357
From RelCode 5C52 FALES: 17, List: 357
To RelCode 5CBA MAIN: 7, List: 445
To RelCode 5CBA FALES: 18, List: 359
From RelCode 5CBA MAIN: 7, List: 445
From RelVars 5D50 STAGEK: 55, List: 70
To RelVars 5EAC STAGEK: 58, List: 73
From RelVars 5EAC PKCORE: 3, List: 98
To RelVars 5FFE PKCORE: 4, List: 99
To RelVars 5FFE MAIN: 2, List: 119
From RelVars 5FFE MAIN: 2, List: 119
From V2 80CO OPSYS: 1, List: 124
To V2 82CO OPSYS: 1, List: 124
From V2 82CO VARS: 22, List: 30
To V2 8902 VARS: 23, List: 31
From V2 8902 VARS: 23, List: 31
To V2 8F62 VARS: 23, List: 31
From V2 8F62 VARS: 29, List: 37
To V2 A000 HACCON: 1, List: 39
From Vars O STAGEK: 13, List: 28
To Vars 60CO STAGEK: 14, List: 29
From Vars 60CO STAGEK: 14, List: 29
To Vars 625A STAGEK: 15, List: 30
From Vars 625A STAGEC: 21, List: 70
To Vars 625E STAGEC: 21, List: 70
From Vars 625E DDT: 4, List: 78
To Vars 627E DDT: 4, List: 78
From Vars 627E DDT: 4, List: 78
To Vars 6280 DDT: 5, List: 80
From Vars 6280 DDT: 5, List: 80
To Vars 62CE DDT: 5, List: 80
From Vars 62CE OPSYS: 1, List: 124
To Vars 62DA OPSYS: 2, List: 125
From Vars 62DA VARS: 17, List: 25
To Vars 63D8 VARS: 19, List: 27
From Vars 63D8 VARS: 19, List: 27
To Vars 656E VARS: 22, List: 30
From Vars 656E VARS: 23, List: 31
To Vars 7280 VARS: 29, List: 37
From Vars 7280 IMPDDT: 1, List: 62
To Vars 72A4 IMPDDT: 2, List: 63
From Vars 72A4 DISPLAY: 1, List: 76
To Vars 72A6 DISPLAY: 1, List: 76
From Vars 72A6 ROUTE: 1, List: 181
To Vars 72B8 ROUTE: 1, List: 181
From Vars 72B8 ROUTE: 2, List: 182

From Vars 73FC VHA: 1, List: 433
To Vars 7400 VHA: 1, List: 433
From Warm 40CO HACCON: 2, List: 40
To Warm 40F2 HACCON: 3, List: 41
From Warm 40F2 HACCON: 4, List: 42
To Warm 4142 HACCON: 5, List: 43
From Warm 4142 HACCON: 5, List: 44
To Warm 4162 HACCON: 6, List: 45
From Warm 4162 IMPSUB: 1, List: 49
To Warm 41AA IMPSUB: 2, List: 50
From Warm 41AA LOCAL: 3, List: 85
To Warm 41DA LOCAL: 4, List: 86
From Warm 41DA CONFIG: 9, List: 100
To Warm 4256 CONFIG: 10, List: 101
To Warm 4256 CONFIG: 11, List: 102
From Warm 4256 CONFIG: 10, List: 101
From Warm 4256 FASTTO: 1, List: 117
To Warm 44D8 FASTTO: 9, List: 125
To Warm 44D8 STO: 1, List: 127
To Warm 44D8 FASTTO: 9, List: 125
To Warm 44D8 FASTTO: 9, List: 125
From Warm 44D8 UPDWN: 1, List: 131
To Warm 46B2 UPDWN: 6, List: 136
From Warm 46B2 UPDWN: 6, List: 136
To Warm 479A UPDWN: 9, List: 139
From Warm 479A MODEM: 2, List: 142
To Warm 47A4 MODEM: 2, List: 142
From Warm 47A4 MODEM: 4, List: 144
To Warm 4834 MODEM: 6, List: 146
From Warm 4834 MODEM: 6, List: 146
To Warm 4870 MODEM: 7, List: 147
From Warm 4870 MODEM: 7, List: 147
To Warm 48B6 MODEM: 8, List: 148
From Warm 48B6 MODEM: 18, List: 159
To Warm 4AF8 MODEM: 25, List: 167
From Warm 4AF8 DEL: 2, List: 173
To Warm 4B64 DEL: 3, List: 175
From Warm 4B64 DEL: 4, List: 176
To Warm 4C6C ROUTE: 1, List: 181
From Warm 4C6C ROUTE: 3, List: 183
To Warm 4DOA ROUTE: 5, List: 185
From Warm 4DOA ROUTE: 6, List: 186
To Warm 55D2 ROUTE: 34, List: 216
From Warm 55D2 ROUTE: 35, List: 217
To Warm 564A ROUTE: 37, List: 219
From Warm 564A ROUTE: 37, List: 219
To Warm 5924 ROUTE: 45, List: 227
From Warm 5924 ROUTE: 45, List: 227
To Warm 5970 ROUTE: 46, List: 228
From Warm 5970 ROUTE: 46, List: 228
To Warm 59C6 TASK: 1, List: 231
From Warm 59C6 WARM: 22, List: 262
To Warm 5A96 WARM: 24, List: 264
From Warm 5A96 WARM: 25, List: 265
To Warm 5B24 WARM: 27, List: 267
From Warm 5B24 WARM: 28, List: 268
To Warm 5C06 WARM: 30, List: 270
From Warm 5C06 WARM: 30, List: 270
To Warm 5C3E WARM: 31, List: 271
From War 5C3E WARM: 47, List: 287


```
version: Loading "IMP.VARS"
Closing "IMP.VARS"
Loading "HACCON"
Closing "HACCON"
Loading "IMPSUB"
Closing "IMPSUB"
Loading "IMPDDT"
Closing "IMPDDT"
Loading "DISPLAY"
Closing "DISPLAY"
Loading "IMP.LOCAL"
Closing "IMP.LOCAL"
Loading "CONFIG"
Closing "CONFIG"
Loading "FASTTO"
Closing "FASTTO"
Loading "STO"
Closing "STO"
Loading "UPDWN"
Closing "UPDWN"
Loading "MODEM"
Closing "MODEM"
Loading "DEL"
Closing "DEL"
Loading "ROUTE"
Closing "ROUTE"
Loading "TASK"
Closing "TASK"
Loading "IMP.WARM"
Closing "IMP.WARM"
Loading "FAKSUB"
Closing "FAKSUB"
Loading "FAKES"
Closing "FAKES"
Loading "FAKREL"
Closing "FAKREL"
Loading "VHA"
Closing "VHA"
Loading "IMP.VARS"
Closing "IMP.VARS"
Loading "HACCON"
Closing "HACCON"
Loading "IMPSUB"
Closing "IMPSUB"
Loading "IMPDDT"
Closing "IMPDDT"
Loading "DISPLAY"
Closing "DISPLAY"
Loading "IMP.LOCAL"
Closing "IMP.LOCAL"
Loading "CONFIG"
Closing "CONFIG"
Loading "FASTTO"
Closing "FASTTO"
Loading "STO"
Closing "STO"
Loading "UPDWN"
Closing "UPDWN"
```

Closing "MODEM"
Loading "DEL"
Closing "DEL"
Loading "ROUTE"
Closing "ROUTE"
Loading "TASK"
Closing "TASK"
Loading "IMP.WARM"
Closing "IMP.WARM"
Loading "FAKSUB"
Closing "FAKSUB"
Loading "FAKES"
Closing "FAKES"
Loading "FAKREL"
Closing "FAKREL"
Loading "VHA"
Closing "VHA"
Page LCode patch space: 3ED6 to 4000
Page Re1Code patch space: 5CBA to 5D50
Page DDTCode patch space: 57A8 to 5B00
Page Warm patch space: 5E70 to 5E80
Page FakCode patch space: 5AE8 to 5E00
Page LTVars patch space: 42 to 50
Page LVars patch space: 288 to 288
Page Vars patch space: 7400 to 8000
Page Re1Vars patch space: 5FFE to 6000
Page DDTVars patch space: 5F80 to 6000
Page PkgVars patch space: 5EB4 to 6000
Page FakVars patch space: 5F96 to 6000
Page VarPat patch space: 656E to 65B0
Page LKPatch patch space: 332 to 352
Page RKPatch patch space: 4126 to 4166

332 SECONDS RUN-TIME

Loading "LPMAC"
Closing "LPMAC"
Loading "PROC"
Closing "PROC"
Loading "STRIP"
Closing "STRIP"
Loading "STAGEC"
Closing "STAGEC"
Loading "DDT"
Closing "DDT"
Loading "OPSYS"
Closing "OPSYS"
Loading "LPMAC"
Closing "LPMAC"
Loading "PROC"
Closing "PROC"
Loading "STRIP"
Closing "STRIP"
Loading "STAGEC"
Closing "STAGEC"
Loading "DDT"
Closing "DDT"
Loading "OPSYS"
Closing "OPSYS"

66 SECON C RUN-TIME

Loading "STAGE.CFG"
Closing "STAGE.CFG"
Loading "STAGEK.PLR"
 Loading "IMP.PKCORE"
 Loading "XSIOIN"
 Closing "XSIOIN"
 Closing "IMP.PKCORE"
Closing "STAGEK.PLR"
Loading "STAGE.CFG"
Closing "STAGE.CFG"
Loading "STAGEK.PLR"
 Loading "IMP.PKCORE"
 Loading "XSIOIN"
 Closing "XSIOIN"
 Closing "IMP.PKCORE"
Closing "STAGEK.PLR"

84 SECONDS RUN-TIME

luribus IMP PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.PLR;1 PAGE 1

PAGE 1

```

.title pluribus IMP

;Get symbols from the IMP Operating System (IMPOPS) Assembly

.insym impoops.sym

0001      LBig = 1

;Assume we are debugging

0000      Debug = 0

;Assemble VDH with the program

0000      VDHSw = 0

;Don't assemble support for dual M/I busses

0001      MISw = 1

;Don't assemble Platform Satellite Extension features

0001      PSE = 1

;Platform Virtual Host Addressing (Added 3 NOV 82 by Callis)

0001      VHA = 1

;Maximum number of IMP/IMP modem channels

0080      Nmd1im = D128 ;multiple of 16 (changed to 128 as per JR)
                  ;(3 NOV 82 by Callis)

.INSRT IMP.MAIN

```

pluribus IMP PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 2
IMP.MAIN;1 PAGE 1

.comnt |
Pluribus IMP system

Maintenance/statistics stuff:
NCC stuff: trace, various stats, etc.

Modem stuff

Reliability stuff
HI reliability
RTC reliability
Base reliability
Queue reliability

Host stuff
non-blocking hosts
Host test
handle host hardware gone away
? reliable allocate protocol

DDT stuff
extension DDT

General changes:
make POINT a logical pointer
make CHAIN, POINT, etc. live with buffer (i.e. chain thru all memory)
RADIX H10
bandwidth improvements
processor use, high-speed lines, buffering, strip lengths

Stage/IMP stuff:
Amputation OPHELPs/TENEX support
processor test?
random amputation
split without restarting
copy of local in common?

pluribus IMP PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.MAIN;1 PAGE 2

PAGE 3

484-7326 JR
926-3072 Kats
Drew.

NCC 661-0100, x3571, 800-225-1604, 1922
TENEX 491-6169, x4358

pluribus IMP PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 4
IMP.MAIN;1 PAGE 3

.comnt |

ophelp routine

IMP Ophelps

Ophelps are used to simplify common operations on the IMP. Generally Ophelps consist of two letter commands whose initials indicate somewhat the purpose of that ophelp. The current list of available ophelps follow.

```
#HB    return the parameter block address of host #
#HL    loop (crosspatch) host # at the IMP interface
#HM    loop VDH # at the Bell modem
#HU    remove all loops from host #

KI    return the address of the IOKILL table
KP    return the address of PROHLT to kill processors
      (not yet implemented)

#MB    return the address of modem block for modem #
#ML    loop modem # at the imp interface
#MM    loop modem # at the Bell modem
#MU    remove loops from modem #

#NH    Nice halt the IMP forever
#NS    Nice halt and restart the IMP
#NR    Nice halt and reload the IMP

#PH    Panic halt the IMP forever
#PS    Panic halt andrestart the IMP
#PR    Panic reload the IMP

#RN    Reload neighbor from imp #

<esc>C Clear the trap table
```

\ /
> # = password

pluribus IMP PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.MAIN;1 PAGE 4

PAGE 5

```
;System Version Numbers
;get version number for the IMP and title the listing
; this could be extended so that the version number's oddness
; or eveness would determine the setting of LBIG.
.if p1                      ;just pass one
.print /version: /
.ttyma impver
    hsmver= 0'impver        ;system version number
.endm
.endc

.macro $title x
    .title Pluribus IMP 'x
.endm

$title \hsmver
0000      .comptv=0           ;routing compatibility version number

;Test the switches we need for rest of assembly
testsw <LBIG.Debug,VDHSw,PSE,VHA>

.if ndf Nmdlim
.print |Nmdlim undefined, assumed 16|
Nmdlim = D16
.endc ;ndf Nmdlim
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 6
IMP.MAIN;1 PAGE 5

;Define the initial pages to load

```
0400    codlod= H400
0600    faklod= H600
0001    .if z LBig
          pkglod= H800
          ncplod= HA00
          .endc

;Page origins

0040    loctvs= H40      ;local vars in exec core for TIP
5E80    Pkgvst= H5E80   ;Package vars always start here.
5EO0    fakvst= H5EO0   ;Fake vars
```

;Define the physical pages for the LMAP table

.comnt |

The system memory can be laid out in two ways; one will minimize the use of local memory so that the TIP can be run; the other will maximize the use of local memory to obtain higher throughput.

If LBig is zero, some amount of common memory code and the routing code will be placed on a new code page called the "Package" page. Some parts of the IMP normally (i.e. when LBIG = 1) in local and defined as being in "HLCODE" will be moved onto the "Warm" page.

If LBig is one, the "Package" page is defined to be the common memory page which holds "Warm" code. Routing code will be placed on the Warm page, making room for the routing variables as necessary. Those parts of the IMP in "HLCODE" are located in local memory.

VDH code, if present, always goes onto the DDT page.

|

```
0001    .if z LBig
        CODEPAGE COD, PHYSICAL CODLOD, LIMIT M1
        CODEPAGE FAK, PHYSICAL FAKLOD, LIMIT FAKVST
        CODEPAGE PKG, PHYSICAL PKGLOD, LIMIT PKGVST

        Defpage Warm, map code, limit m1, physical cod, org *
        Defpage PkgCode, map code, limit pkgvst, physical pkg, org *
        Defpage PkgVars, org pkgvst, limit m1
        Synpage HLCODE,Warm
        Synpage RutCode,PkgCode
        Synpage RutVars,PkgVars
        maprput = mappkg
        .iff
        CODEPAGE COD, PHYSICAL CODLOD, LIMIT PKGVST
        CODEPAGE FAK, PHYSICAL FAKLOD, LIMIT FAKVST

        Defpage Warm, map code, limit pkgvst, physical cod, org *
        Synpage PkgCode,Warm
00B4      mappkg = mapcod
        Defpage PkgVars, org pkgvst, limit m1
        Synpage RutCode,Warm
00B4      maprput = mapcod
        Synpage RutVars,PkgVars
        Synpage VHACode,DDTCode
00B2      mapvha = mapddt
        Synpage VHAVars,DDTVars
        Synpage HLCode,LCode
        .endc

0001      VLoc=1 ;VDH in common memory.
        Synpage VDHCode,DDTCode
00B2      mapvdh = mapddt
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 8
IMP.MAIN;1 PAGE 7

;Define remaining pages

DESVPAGE B1
DESVPAGE B2

OPTVPAGE B3
OPTVPAGE B4
OPTVPAGE B5
OPTVPAGE B6

;Define the logical pages

Defpage LTVars, org loctvs, limit locvst
Defpage FakCode, map code, limit fakvst, physical fak, org *
Defpage FakVars, org fakvst, limit m1
Defpage PkgVars, org pkgvst, limit m1

Synpage VLCode,LCode

;Define some Stage Configuration Stuff

DEFREL ,reltim ;Rely page TO pointer

;Minimum number of processors to run the IMP

DEFPCNT 2

;Now load in all the IMP files

.INSERT "IMP.VARS",VARS
.INSRT IMP.VARS

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 1 IMP Variables

PAGE 9

.stitle IMP Variables

PAGE Dummy

```
;some system parameters
0018 nrh= D24      ;number of real hosts
0004 nfh=4       ;number of fake hosts
0008 nfh2=nfh*words ;and in bytes
001C th=nfh+nrh ;total hosts in system
0008 nmd= D8      ;number of modems (neighbors)
007F nimp= D127   ;number of imps
0190 nline = D400  ;(one-way) lines to allow in network
0002 noimp0=words ;imp 0 not in routing tables
;nach=8          ;number of ack channels per line

000A arpano= D10  ;ARPANet Net number
000A mynet=arpano ;and that's where we are
001E bbntip= D30  ;IMP no of BBN TIP
003F bbn63= D63   ;IMP no of Backroom BBN machine (proto 516)
001F cca= D31     ;CCA IMP number
0027 sdac= D39   ;SDAC IMP number
00F1 tenexc= D361 ;Host TENEX System C at RCC IMP
0005 tenexe=5    ;Host TENEX System E at BBN IMP
0000 tenexn= 00   ;Host NSA TENEX on Platform IMP 1
0001 ncc=1        ;NCC is on IMP 1 in Platform
0000 ntenex=tenexn ;NCC TENEX Host address
0000 n10xh=ntenex_-6 ;srceh ;Host number for NCC TENEX
0000 n10xi=ntenex&srcei   ; and IMP number
0028 sec1= D1000/ D25  ;RTC ticks (pids) per second
;actually off by a bit
04B0 sec30=sec1* D30 ;number of ticks in 30 secs
0258 sec15=sec1* D15 ; and in 15
0078 sec3=sec1*3    ; and 3 for tty
01D4 medmin=6*< D10000/ D128> ;medium ticks in a minute
03E8 rrttime= D1000  ;nominal retransmit time, times 8 for slow line
000A iniall= D10    ;normally hold allocates 256 milliseconds
8000 sign= H8000

0100 modid= H100
0200 hostid= H200
0500 hmodid= H500  ;high-speed (eia) modem
0600 satmid= H600  ;satellite modem
0700 satmhi= H700  ;high-speed satellite
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 10
IMP.VARS;9 PAGE 2 IMP Variables

```
;order of locks (seventh shot) 8 August 77
;
; tolock
; back lockhi
; rmlock
; rea1/fake lockhi,ih1oc,i2m1oc,m2i1oc
; tmlock,trnlok,lockm
; lockih,lockro
; nf,free/freend,ltq,s1f1k,reas1k,bbc1ok,all consensus locks,
; ...d2f1,f2d1,t2f1,f2t1,lmiq,b5q1
```

uribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 11
IMP.VARS;9 PAGE 3 IMP Variables

0000 .=0 ;*** modem parameter block
0000 4000 lockm=.+foo
0000 .blkw 1 ;lock
0002 phflag: .blkb 1 :line up/down state flags.
0001 master = 1 ;master bit
FFFE slave = -1?master ;to mask out master
0002 inhrst = 2 ;if set, don't restart reset
0004 phup = 4 ;phantom UP bit
FFFB phdown = -1?phup ;to mask out UP bit
0008 rcdhel = H8 ;rec'd hello this tick
0080 shihy = H80 ;sign ;set -> send hello/i heard you
FF05 helst = HFF00+master+phup ;state bits to send in H/IHY
FFF7 clrst = -1?rcdhel ;mask for clearing states
0003 auxcnt: .blkb 1 ;counter for line logic:
;reset: ticks since hardware reset
;master down: consec. hits
;slave: consec misses
0004 lnei: .blkb 1 ;last neighbor on this line (for SPF)
0005 neigh: .blkb 1 ;neighbor on this line
;hardware statistics
0006 ckerrs: .blkw 1 ;hardware checksum errors
0008 lword: .blkw 1 ;modem hardware throughput
000A hiword: .blkw 1 ; double precision
000C flipper: .blkw 1 ;counts half words
;hardware configuration
000E motpid: .blkb 1 ;output pid \ these must correspond to
000F minpid:.blkb 1 ;input pid / hipid/hopid, bhpid }
0010 nullhd: .blkw 5 ;send nulls from here (4 bit boundary})
001A modem: .blkb 1 ;logical modem number * 2 (from 0)
001B clockm:.blkb 1 ;line "speed": protocol set to use
001C maxchn: .blkw 1 ;max channel number this line (times 2)
001E mloop: .blkw 1 ;loop bits for interface
0020 ;hardware
0021 iobloc: .blkw 1 ;io address
0022 altio: .blkw 1 ;address of spare interface
4024 i2mloc=.+foo ;output side variables
0024 .blkw 1 ;output hardware lock
0026 snding: .blkw 1 ;data buffer being sent (0=none)
0028 later: .blkw 1 ;flush this buffer when sent
002A i2mpok: .blkw 1 ;rare events flag for modem out
002C demand: .blkw 1 ;demand-reload on this line
002E sblk: .blkw 1 ;send pkc block pointer
0030 snull: .blkw 1 ;send null flags (1 per ACK group)
0032 mrtime: .blkw 1 ;retransmit time interval
0034 ssentq: .blkw 1 ;packets not yet acked
0036 esentq: .blkw 1 ;
0038 spriq: .blkw 1 ;priority output q | keep as block}
003A epriq: .blkw 1 ;
003C sreqq: .blkw 1 ;regular output q
003E ereqq: .blkw 1 ;
0040 thrput: .blkw 1 ;number of ack'ed packets

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 12
 IMP.VARS;9 PAGE 4 IMP Variables

```

 4042     m2iloc=.+foo      ;input side vars
0042         .blkw 1       ;receive side hardware lock
0044         fillin: .blkw 1   ;current input buffer
0046         nxtbf: .blkw 1   ;buffer to input to next
 4048     lmiq=.+foo
0048         .blkw 1       ;modem input queue lock
004A         smiq: .blkw 1   ;input q
004C         emiq: .blkw 1   ;input queue
004E         mithru: .blkw 1  ;input throughput from acks

0050     slots: .blkw 1 ;free slots on this line

0052     tsex: .blkw Nmdlim/ D16      ;transmit odeven bit (i2m)
0062     rsex: .blkw Nmdlim/ D16      ;receive odeven bit (m2i)
0072     infree: .blkw Nmdlim/ D16    ;one if input chan free
0082     chnbsy: .blkw Nmdlim/ D16    ;one if i2m chan is busy

 4092     dlock=.+foo      ;modem delay variables
0092         .blkw 1       ;lock
0094         dpcnt: .blkw 1   ;packet count
0096         delhi: .blkw 1   ;high delay bits.
0098         dellow: .blkw 1   ;low delay bits
009A        delave: .blkb 1   ;current average delay.
009B        delbas: .blkb 1   ;current base delay.

          ;line up/down vars
009C     rutwait:.blkw 1      ;tikcnt at entry to reset or line up
009E     rutclk: .blkb 1      ;routing clock to use
009F     rmodn: .blkb 1      ;unused
00A0     rutspd: .blkb 1      ;fast tick interval for routing

00A1     kpoint: .blkb 1 ;current pointer into odelt
00A2     odelt: .blkb H8      ;saved delta for ea miss
 0008     odelt = .-odelt
00AA     lmiss: .blkw 1      ;tikcnt at last miss
00AC     tdelts: .blkw 1      ;total ticks over last k-1 misses
00AE     k: .blkb 1       ;K-out-of-N line down parameters
00AF     N: .blkb 1       ;N
00B0     NUP: .blkb 1      ;ticks to come up
00B1     tikrat: .blkb 1     ;clock for line ticks
00B2     tictim: .blkw 1     ;protocol clock
00B4     ihytik: .blkw 1     ;protocol ticks of line
00B6     gotihy: .blkw 1     ;received hellos/IHY's

          ;update retransmission vars
 40B8     rtimrl=.+foo      ;lock on retrans variables
00B8         .blkw 1
00BA     rtrclk: .blkb 1      ;fast ticks left in tick(must be left byte)
00BB     rtrtic: .blkb 1      ;fast ticks per retrans tick(must be right byte)
 0019     rticd = D25        ;tick dead lines every 640 ms
 0001     rtic1 = D1          ;tick live land lines every 25 ms
 000F     rtics = D15         ;tick high delay (satellite) lines at 375 ms
 00BC     rtimrs: .blkw <>nimp-1>-3>+1 ;two-bit retrans timers
 00DC     mspare: .blkw 4      ;minimum spares
 00FO     modlen=.+ HE& HFFFO ;length in bytes
  
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 13
IMP.VARS;9 PAGE 5 IMP Variables

```
0000          .=0      ;*** host parameter block
;*** host-to-imp section

        4000      lockhi=.+foo
0000          .blkw 1 ;lock on parameters
0002      hostyp: .blkw 1 ;nz => fake host, m => back host, 0 => real host
        8000      forbak=sign ;if back host
0004      hilo:   .blkw 1 ;nominal dispatch location
0006      htemp:  .blkw 1
0008      htemp7: .blkw 1 ;save r7
000A      hisav7: .blkw 1
000C      hitt:   .blkw 1 ;timer, 1->0 => host blocked by imp,
                           ; -1->0 => host took too long
        motpid:
000E      hotpid: .blkb 1 ;output PID
        minpid:
000F      hinpid:.blkb 1 :pid level for hi

; exact 4 bit boundary ( Hxxx0)

0010      ihldr: .blkw 6 ;imp-host leader (ihloc)
001C      homode: .blkb 1 ;80 -> extended leaders
                           ; 1F field is padding length in bytes
001D      holhn: .blkb 1 ;logical host number (h2pb1k offset)
001E      hibits: .blkb 1
        0001      tskfok=1 ; (odd) => task took it
        0002      tskfrf=2 ; 2 => task refused it
        0004      tskfre=4 ;flag for task to free trn b1k for host
        0008      hirset= H8 ;set by hlc (15 sec) timeout
001F      hihd:  .blkb 1 ;host state
        0000      hostup=0 ;0 => up
        0001      hrdown=1 ;1 => ready line down
        0002      htardy=2 ;2 => tardy
        0003      hnexist=3 ;3 => nonexistent
        0004      hstqut=4 ;4=> host received quit
        0004      hninit=4 ;:chini ;4 => imp software uninitialized

0020      iobloc: .blkw 1 ;io interface or fake iobloc ptrn
0022      altio: .blkw 1 ;spare interface address (or zero)
0024      hiloop: .blkw 1 ;bits that crosspatch interface
0026      hipkth: .blkw 1 ;construct packet header here
0028      hihost: .blkw 1 ;host pair word in format used by
                           ;msg block system
002A      hihand: .blkw 1 ;handling type
002C      deadsc: .blkw 1 ;dead subcodes
                           ;from ihldr to here may be sent to host as padding
        0009      himaxp= D9 ;maximum padding, in words
002E      hitran: .blkw 1 ;ptr(transaction block) for current msg
0030      hioldb: .blkw 1 ;ptr(transmit msg block)
0032      hibf:   .blkw 1 ;active hardware input buffer
0034      hisp:   .blkw 1 ;buffer which has an input packet
0036      hiendi: .blkw 1 ;endi for buffer in hisp
0038      hiptip: .blkw 1 ;PTIP temp
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 14
IMP.VARS;9 PAGE 6 IMP Variables

```
;*** host thruput statistics (continuation of host parameter block)

003A      htpmtn: .blkw 1 ;messages to net
003C      htpmfn: .blkw 1 ;messages from net
003E      htpptn: .blkw 1 ;packets to net
0040      htppfn: .blkw 1 ;packets from net
0042      htpmtl: .blkw 1 ;messages to local
0044      htpmfl: .blkw 1 ;messages from local
0046      htpptl: .blkw 1 ;packets to local
0048      htppfl: .blkw 1 ;packets from local
004A      htpwti: .blkw 1 ;words to imp
004C      htpwfi: .blkw 1 ;words from imp
0014      hcctrl=-htpmtn ;length of counters

;*** imp-to-host (continuation of host parameter block)

404E      ihloc=.+foo
004E      .blkw 1 ;lock on output hardware status bit, etc.
0050      ihlo: .blkw 1 ;nominal dispatch (locked by ihloc)
0052      ihtt: .blkw 1 ;software timer (ihloc)
0054      ihgoin: .blkw 1 ;host timeout (ihloc)
0056      ihwq: .blkw 1 ;which queue got serviced last (ihloc)
0058      ihlstp: .blkw 1 ;flag, just sent last pkt of msg (ihloc)
005A      specal: .blkw 1 ;flag => send leader (ihloc)
405C      lockih=.+foo
005C      .blkw 1 ;locks ih queues and one trnb lk state
005E      nxtled: .blkw 1 ;num replies,,<last reply>/ H10 (lockih)
0060      shq: .blkw 1 ;regular host msg queue (lockih)
0062      ehq: .blkw 1
0064      shpq: .blkw 1 ;start, host priority queue
0066      ehpq: .blkw 1 ;end, host priority queue
0068      ihbuff: .blkw 1 ;assign a new variable in host block
006A      ihbubb: .blkw 1 ;copy of BUFB from buffer
006C      ihpkth: .blkw 1 ;copy of PKTH from buffer
006E      ihseqh: .blkw 1 ;copy of SEQH from buffer

0070      holen=<.+ HE>& HFFF0 ;bytes in host block

;values for HOSTYP

8000      forbak = sign ;background "host"

0000      .=0
0000      hosreal: .blkw 1 ;real host
0002      hosfake: .blkw 1 ;IMP software ("fake") host
0004      hosvdh: .blkw 1 ;Very Distant Host
0006      hostip: .blkw 1 ;Arpanet TIP
0008      hosty8: .blkw 1 ;unused
000A      hostyA: .blkw 1 ;unused
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 15
IMP.VARS:9 PAGE 7 IMP Variables

;*** common block for replies

0000 .=0

0000 ;lockhi=.+foo ;back lock
0000 .blkw 1
0002 hostyp: .blkw 1 ;nz => fake host, m => back host
0004 hilo: .blkw 1 ;nominal dispatch location
0006 htemp: .blkw 1
0008 htemp7: .blkw 1 ;save r7
000A hisav7: .blkw 1 ;could be separate background temp
000C hitt: .blkw 1 ;timer, 1->0 => host blocked by imp.
000E .blkb 1 ;unused
000F hinpid: .blkb 1 ;back host input PID.
0010 btypb: .blkw 1
0012 bseqh: .blkw 1
0014 bpkth: .blkw 1
0016 bdsth: .blkw 1
0018 bmidh: .blkw 1
001A bdata: .blkw 1
001C bmessb: .blkw 1
001E hibits: .blkb 1 ;flag bits, see host block.
001F .blkb 1

0020 baklen=.

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 16
IMP.VARS;9 PAGE 8 IMP Variables

```
;New format leader words
.0000      .=0
.0000      netl: .blkw 1 ;network address, 0FOO indicates new format
.0002      typ1: .blkw 1 ;message type, trace, octal bits
.0004      hst1: .blkw 1 ;handling type(priority)..host
.0006      dst1: .blkw 1 ;destination(or source) imp
.0008      mid1: .blkw 1 ;message id and subtype
.000A      len1: .blkw 1 ;message length in bits sans leader and padding

;Bits in an old format leader

;hiledr:
.4000      forimp= H4000    ;for IMP fake host
.0FO0      hicode= HF00    ;message type
.0OC0      desth= HCO      ;destination host
.003F      desti= H3F      ;destination IMP
.0OFF      destih=desth}desti   ;both of the above

;ihledr:
.4000      frmimp= H4000    ;from IMP fake host
.0FO0      ihicode= HF00    ;message type
.0OC0      srceh= HCO      ;source host
.003F      srcei= H3F      ;source IMP
.0OFF      srcehi=srcei}srceh ;the reader is left this as an exercise
```

:Words in a packet

0000 .=0

0000 vdso: .blkw 1 ;VDH temp
0000 neth: .blkw 1 ;network header
0002 typh: .blkw 1 ;packet type header
0004 chkh: .blkw 1 ;software checksum of packet
0006 srch: .blkw 1 ;source IMP header
0008 seqh: .blkw 1 ;message sequencing header
000A pkth: .blkw 1 ;packet flags header
000C dsth: .blkw 1 ;destination IMP header
000E midh: .blkw 1 ;message-ID header
0010 data: .blkw D63 ;beginning of real data
008E bufend: .blkw 0 ;end of the buffer for software
008E .blkw 1 ;unused
0090 bufe: .blkw 1 ;buffer end pointer
0092 inch: .blkw 1
0094 stimer: .blkw 1
0096 it: .blkw 1
0098 qt: .blkw 1
009A st: .blkw 1
009C bufb: .blkw 1 ;mess blk for ih if any, else odd (raw pkt)
009E .blkw 1 ;spare
00A0 buflen: .blkw 0 ;buffer plus extra words (bufc, etc.)

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 10 IMP Variables

PAGE 18

417

;Bits in neth

```
8000  odeven= H8000 ;odd/even bit
4000  endbit= H4000 ;I am high numbered IMP
2000  dscpkt= H2000 ;ack and discard this packet
0FO0  chanum= HF00 ;ack channel this packet
```

;Bits in typh

```
c000  paktyp= HC000 ;packet type this packet
0000  regpkt=0 ;regular packet
4000  qertyp= H4000 ;query or getblk type
8000  ruttyp= H8000 ;routing type
C000  rldtyp= HC000 ;reload type
2000  combit= H2000 ;compatibility of this type
1000  pribit= H1000 ;priority bit
1000  rutupl= H1000 ;null 'up' bit
0800  trcbit= H800 ;trace if one
0700  pflags= H700 ;packet flags (depends on type):
0600  compat= H600 ;routing compatibility number (routing)
0400  stubit= H400 ;"I am not a stub" (null)
0400  octbit= H400 ;packet for octal print (reg. req)
0400  alloc= H400 ;packet carries allocate (inc ?, getblk)
0200  1drrut= H200 ;route this packet by SPF(set from new leader)
0200  hac= H200 ;dead/hacc (getblk reply)
0200  relreq= H200 ;reload request (reload)
0100  noblk= H100 ;got-no-block (getblk reply)
0300  ruptyp= H300 ;routing update (routing)
0200  iherdu= H200 ;hello/I heard you (routing)
0002  spfon= H2 ;i=>SPF running (set in Hello/I heard you)
0100  rutnul= H100 ;0 -> routing, 1 -> null (routing)
0100  demrel= H100 ;demand reload (reload)
0000  pcore=0 ;packet core (reload)
0OFF  akbits= HFF ;acknowledge bits
```

```
;Bits in seqh
CAFE      dempas= HCAFE    ;reload password
0008      rutdat=seqh     ;beginning of routing data (routing)
0100      mesn0= H100      ;low-order bit of message number
FF00      mesnum= D255*mesn0   ;message number (regular, query)
FF00      dsthst= HFF00    ;destination host (raw pkt, getblk)
0009      fblock=seqh+1    ;foreign block )byte address) (reg, query)
0009      srchst=seqh+1    ;source host (raw pkt, getblk)

;Bits in pkth
000A      unhacm=pkth     ;source hacmem (raw pkt)
8000      m1tpkt= H8000    ;multi-pkt message (reg mes)
4000      lstpkt= H4000    ;last packet of message (reg mes)
0F00      fuse= HF00      ;foreign use number (reg mes)
00F0      pktnum= HF0      ;packet message number (reg mes)
000F      ptkcod= HF       ;packet code (reg mess, query)
                    ;Regular message (type 0) codes:
0000      regtyp=0         ;regular message
0001      reqtyp=1         ;request for allocation
0002      gvbtyp=2         ;giveback an allocation
0003      inmtyp=3         ;incomplete this message
0004      rfntyp=4         ;ready-for-next-message
0005      rfatyp=5         ;rfnm with allocatiiion
0006      dedtyp=6         ;rfnm and destination dead
0007      inrtyp=7         ;rfnm and this message incomplete
                    ;Query message (type 1) codes:
0008      inqtyp= H8        ;incomplete query
0009      gabtyp= H9        ;get a receive message block
000A      rsbtyp= HOA       ;reset this receive message block
                    ;unused=
000C      outtyp= HOC       ;got out-of-range message
000D      gbrtyp= HOD       ;get-a-block reply
000E      rrqtyp= HOE       ;request a reset on this transmit mess..blk.
000F      rrptyp= HOF       ;reply to a reset from trasmit block
0004      trnrep=4          ;reply if this is a 1
                    ;forus bits (in fuse field)
0100      hostng=-fuse&fuse   ;low bit of fuse
0200      meseen=hostng^2
0400      mesout=meseen^2
0800      mesnxt=mesout^2

;Bits in midh
FFFO      messid= HFFFFO    ;message ID (reg mess, raw pkt)
000F      subtyp= HF        ;subtype (reg,raw pkt):
0000      regpkt=0          ;regular message
0003      rawpkt=3          ;raw packet (uncontrolled packet)
F000      gethan= HF000    ;handling type (getblk)
8000      getpri= H8000    ;priority handling
7000      getmax= H7000    ;max length message
0F00      getuse= HF00      ;local use (getblk)
00FF      locblk= HFF       ;local block (getblk)
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 12 IMP Variables

PAGE 20

```
;***words in reassembly block
.0
0000      reaslk=.+foo    ;lock on eachreas block
0000          .blkw 1 ;lock
0002      rsf:   .blkb 1 ;num of pckts so far
0003     reasst: .blkb 1 ;reas blk state:
0001          rsfree=1   ;free
0002          rsalnum=2  ;allocated with message number
0003          rsall=3   ;allocated, no message number
0004      rid:   .blkw 1 ;id = rec mes blk offset
0006      remess: .blkb 1 ;mess number
0007      ruse:   .blkb 1 ;rm blk use numb
0008      rmax:   .blkb 1 ;highest pkt num this msg
0009      ral:   .blkb 1 ;number allocated in this block
000A     reasq: .blkw 1 ;start ofreas queue
000C     reasqe: .blkw 1 ;end
000E      rsfbt: .blkw 1 ;total length in bytes so far
0010          1rb1k= H10
```

; ih codes

00FF mestyp= HFF ;message type field

0001 ;creg=0 :reg
cerrld=1 ;error in hi leader

0000 cerr32=0 ;error in first 32 bits

0001 cshort=1 ;less than 32 bits in message

0002 ci11g1=2 ;illegal hi code

0003 cwrqft=3 ;wrong format

0002 cimpdn=2 ;imp going down
; unused=3

0004 cnop=4 ;nop

0005 crfnm=5 ;rfnm

0006 chstat=6 ;destination host status

0007 cdestd=7 ;destination dead

0000 cimpd=0 ;dest imp dead

0001 chstd=1 ;dest host dead

0002 cldrp=2 ;dest has old leaders

0003 chacc=3 ;host access failure

0004 chini=4 ;host reinit in progress

000F chcup= D15 ;host just coming up
; (was dead a millisecond ago)

FFEO chnoti= 0177740 ;constant for no time known

0008 cerror= D8 ;error in hi data

0009 cinctr= D9 ;incomplete trans

0001 ;cslowd=0 ;dest host took >30 secs

0002 clong=1 ;more than 8095 bits

0003 cs lows=2 ;source host took >15 secs

0003 clost=3 ;lost in subnet

0004 cblock=4 ;source imp took > 15 secs

0005 ciper=5 ;imp detected error after eom

000A creset= D10 ;imp-to-host reset - ready line flap

8000 priled=sign ;priority in leaders

0700 maxled= H700 ;max pkts per msg

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 14 IMP Variables

PAGE 22

```
; words in transaction block
0000      .=0
0000      trnetl: .blkw 1 ;network leader
0000      ttledr=trnetl ;old format leader area
0002      trtyp1: .blkw 1 ;message type leader
0004      trhstl: .blkw 1 ;host and handling leader
                  ; also messno and local tm blk
0006      trdstl: .blkw 1 ;destination IMP leader
0008      trmid1: .blkw 1 ;message-id leader
000A      trluse: .blkw 1 ;local use of transmit mes blk
000A      trdeds=trluse ;dead reply reason(if trstat = ttledr)
000C      trntim: .blkw 1 ;timeout for this tran blk (byte)
000D      trstat=trntim+1 ;status of tran blk (byte)
                  ;if trhost pnts to hstblk, trnblk is locked under that LOCKIH
000E      trpack: .blkw 1 ;packet pointer
000E      trhost=trpack ;what host queue(if trstat = ttledr)
0010      trnl=. ;length of trnblk

;bits in trstat:
;if trstat=0 block is free and locked under trnlok
;if ttresv=1 it's owned by HI
0001      ttrfnm=1 ;reply to host expected
0002      ttmult=2 ;this is a multi-packet req or msg
0004      ttreq=4 ;request is outstanding
0008      ttledr= H8 ;contains a leader for host (trhost)
0020      ttpend= H20 ;ready pending to host
0040      ttgvba= H40 ;message used an allocate
0080      ttresv= H80 ;reserved by host (LOCKHI)
;if non-zero without ttresv,ttledr, or ttreq, locked by TMLOCK
```

```
;hardware and fakes

; words in io block

0000      .=0
0000
0000      statd: .blkw 1
0000      magmod= H8000 ;bit saying modem is magic
0002      starti: .blkw 1
0004      endi: .blkw 1
0002      hrloff=words ;hardware vs software end pntr offset
0000      mendf= H8000 ;last buffer when set in endo,
0000          ;or read in endi
0000      hendf= H8000 ;ditto for host
0001      merrf=1 ;error in this transfer (reading endi)
0001      herrf=1 ;ditto for host
0006      statih:
0006      statim: .blkw 1
0000      mbusy= H2000
0000      hbusy= H2000
0000      hready= H1000
0000      hloopi= H4000
0000      mloopi= H4000
0000      mloope= H8000
0100      hreset= H100
0100      mreset= H100
0100      hquit= H100
0100      mquit= H100
0008      starto: .blkw 1
000A      endo: .blkw 1
000C      statoh:
000C      statom: .blkw 1
000E      motpid:
000E      hbpid: .blkb 1      ;"output" from IH
000F      minpid:
000F      bhpid: .blkb 1      ;"input" to HI
0010      fakesi: .blkw 1
0012      fakeso: .blkw 1
4014      lockfd=.+foo          ;lock for DOZE side
0014          .blkw 1
0016      fioend: .blkw 0 ;VDH extension starts here
4016      lockfw=.+foo          ;lock for WAIT side
0016          .blkw 1
0018      dozesp: .blkw 1        ;doze sp save.
001A      waitsp: .blkw 1        ;wait sp save.
001C          stack dz,< D25> ;doze stack.
004E
004E
004E      stack wt,< D25>    ;wait stack.
0080
0080
0080      ioblen= .           ;length of fake IO block.
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 16 IMP Variables

PAGE 24

;*** various assigned pid levels

0000 .=0
0000 emtpid: .blkw 1 ;this PID is empty, go to next
0002 stkipid: .blkw 1
0004 bltpid: .blkw 1
0006 dsppid: .blkw 1
0008 bhpid0: .blkw 4
0010 bktpid: .blkw 1
0012 ddtpid: .blkw 1
0014 ttypid: .blkw 1
0016 ihtpid: .blkw 1
0018 hbpid0: .blkw 4
0020 tbfpid: .blkw 1 ;PTIP buffer reliability. Also maybe CBT (and next).
0022 ttopid: .blkw 1 ;PTIP fast timeout.
0024 .blkw 1 ;spare
0026 bk1pid: .blkw 1
0028 bk2pid: .blkw 1
002A bk3pid: .blkw 1
002C bk0pid: .blkw 1
002E conpid: .blkw 1
0030 fakip0: .blkw 4
0038 fakop0: .blkw 4
0040 task: .blkw 1
0042 stopid: .blkw 1
0044 rutpi: .blkw 1
0046 vhpid0: .blkw D10 ;VDH to HI (also PTIP to HI)
005A hvpid0: .blkw D10 ;IH to VDH
006E .blkw 1 ;unassigned
0070 .blkw D62+ H8 ;hardware pids go here
00FC nopid: .blkw 1 ;pid when all real PIDs are empty

00FE .blkw 1 ;reserved highest PID

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 25
IMP.VARS;9 PAGE 17 IMP Variables

;imp system variables

Page Vars

62DA time: .blkw 1 ;time in 25.6 ms ticks by main clock
62DC timea: .blkw 1 ;time by alternate clock
62DE sync: .blkw 1 ;time synchronized to network
62EO myimp: .blkw 1 ;IMP-coming-up timer for hosts
62E2 mine: .blkw 1 ;my IMP number
62E4 clk1up: .blkw 1 ;kept 3 if main clock is running
62E6 clk2up: .blkw 1 ;kept 3 if backup clock is running
62E8 clock: .blkw 1 ;addr of clock the system is using
locdef clklok,<;Lock on RTC' counters - page 25>

6000 watch0= H6000 ;loc. for ops to watch

62EC watch: .blkw 1 ;H16-like lites
62EE watchh: .blkw 1 ;lites for 16 hosts

;PSE host/modem status display

62FO ;;;.if nz PSE
mdisp: .blkw D8
6300 hdisp: .blkw D16
;;;;.endc

;flag that is set for reporting restarts, reloads

6320 rstrld: .blkw 1 ;mark restarts, reloads here
0001 rstmrk = 1 ;we restarted
0002 ridmrk = 2 ;we reloaded

;offsets to highest numbered devices

6322 hosts: .blkw 1 ;highest host
6324 modems: .blkw 1 ;highest modem

;the following must be kept contiguous

6326 h2pb1k: .blkw th ;4 fakes (BHIn), then reals or -1s
h2pend:
635E m2pb1k: .blkw nmd ;modem block pointers, then -1s
m2pend:
636E v2pb1k: .blkw th ;pointers to Fake or VDH blocks, or 0
63A6 husend=.
63A6 m2nghb: .blkw nmd ;our n neighbor imps

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 18 IMP Variables

PAGE 26

425

```
locdef l1tq,<;task queue lock - page 26>
63B8    stq:    .blkw 1 ;task queue head
63BA    etq:    .blkw 1
63BC    srq:    .blkw 1 ;secondary task queue
63BE    erq:    .blkw 1
        ; sckq is locked under lockf for fake 3
63C0    sckq:   .blkw 1
63C2    eckq:   .blkw 1

;locdef b5q1,<;back5 queue lock - page 26>
;b5qs:   .blkb 1 ;relative rmbblk pointer
;b5qe:   .blkb 1
```

```
;buffer system variables

locdef free,<;free buffer list - page 27>
locdef freend,<;end of free buffer list - page 27>
locdef nf,<;size of shared buffer pool plus minf - page 27>
;nf also locks counters and 'where' words

63CA      junk: .blkw 1 ;chain word for hardware bit bucket

63CC      maxbuf: .blkw 1 ;index of size of chain
63CE      maxnf: .blkw 1 ;total number of buffers in the system
63DO      minf: .blkw 1 ;how many buffers reserved total
63D2      trendf: .blkw 1 ;for smoothing nf adjustment
63D4      nal: .blkw 1 ;number allocated (contained in nre too)
63D6      trenda: .blkw 1 ;for smoothing nal adjustment

;format of the CNTRS block. see WHERE also
0016      numhsts= D11*words ;number of hosts the PTIP needs room for
page Dummy
0000      .=0
0000      nmi: .blkw 1 ;modem input
0002      nvddhi: .blkw 1 ;VDH to hi
0004      nre: .blkw 1 ;host output (ie reassembly)
0006      nsf: .blkw 1 ;modem output (ie store-and-forward)
0008      nrut: .blkw 1 ;routing send buffers
000A      ntipit: .blkw 1 ;IMP to PTIP
000C      nbak: .blkw 1 ;background hosts input
;the following are per host tables
000E      nhi: .blkw 20 ;host input
002E      ntipti: .blkw numhsts/2 ;PTIP to IMP
0022      numcts=./words

page Vars ;the counters
63D8      cntrs: .blkw numcts ;buffer usage counts
641C      trends: .blkw numcts ;for massaging the counts
6460      newcts: .blkw numcts ;for computing new values for counters

locdef lockro,<;routing send buffers lock - page 27>
locdef cycle,<;timeout clock counters - page 27>
locdef trnlok,<;free transaction blocks lock - page 27>
locdef messt,<;message number timeout non-lock - page 27>
64AC      buftim: .blkw 1
64AE      mnobuf: .blkw 1 ;number modem input lost - no buffers
64BO      rnobuf: .blkw 1 ;number routings lost - no buffers
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 28
 IMP.VARS;9 PAGE 20 IMP Variables

```

    ; *** restarter variables

64B4      locdef ringlk,<;restarter ring lock - page 28>
64B6      ringc: .blkw 1           ; count of entries in ring.
0018      ringf: .blkw 1           ; address of first word in ring.
64B8      ringln=30              ; length in bytes of ring buffer.
64D0      ring: .blkb ringln     ; ring buffer.
64D0      ringe=.                ; address of end of ring buffer.

    ; *** back variables

64D0      rclip: .blkw 1
64D2      tclip: .blkw 1

    ;*** timeout variables

64D6      locdef tcgo,<;host wakeup lock - page 28>
64D6      tcgoa: .blkw 1
64DA      locdef tbkgo,<;back host wakeup lock - page 28>
64DA      tbkgoa: .blkw 1

64DC      stack sto,< D20>

6504
6504
6504
6504      stosp: .blkw 1 ;save stack pointer for slow timeout.
6508      locdef stolok,<;slow timeout lock - page 28>
6508      stoini: .blkw 1 ;slow timeout init; 0 => needs init; nz => ok.
650A      mintim: .blkw 1
650C      bt0a1: .blkw 1
650E      bt0a5: .blkw 1
6510      bt0a7: .blkw 1
6512      ophgo: .blkw 1        ;non-0 means send a trouble report
6514      totmp1: .blkw 1       ;r1 temp over tsleep

    ;**** nicestop and reload variables****

6516      nsftof: .blkw 1        ;OBAD => fast T.O should do stop
6518      nspc: .blkw 1          ;Saved PC of caller
651A      nsrutd: .blkw 1        ;nonzero to send infinite routing
651C      nsrtf: .blkw 1          ;stop flag:
0003      p.halt=3              ;panic halt
0002      p.reload=2             ;panic reload
0001      p.restart=1            ;panic restart
FFFF      n.halt=-1              ;nice halt
FFFE      n.reload=-2            ;nice reload
FFFD      n.restart=-3            ;nice restart
651E      nstlin: .blkw 1         ;if nz, line on which to ask for reload
  
```

```
;*** line logic vars
6520    tikcnt: .blkw 1 ;count of slowticks
        ; *** idle counter
6522    idles: .blkw 1
        ;*** config variables
6524    dynxt: .blkw 1 ;next free location in dynamic area
6526        stack con,< D20>
654E
654E
654E    confsp: .blkw 1 ;save stack pointer for config.
       locdef conlok,<;configuration lock - page 29>
0001    .1if nz MISw ;For M/I bus machines
6552    ckkill: .blkw 1 ;flag - CONFIO doesn't want 0 mem buffers.

        ;*** reliability vars
6554    concer: .blkw 1 ;error in config checksum
        ;*** vars and bufs maps ***
0010    1mapct= D16
6556    vmap: .blkb 1mapct ;copies of mapvar,v2,b1-b6.. (see mapct)
0010    vmap1=-vmap

        ;*** core fake host ***
6566    pchelp: .blkw 1 ;imp to route empty setups to
6568    sfhcq: .blkw 1 ;start of queue initialized in init
656A    efhcq: .blkw 1 ;end of queue
656C    rupqsz: .blkw 1 ;size of entire routing queue
```

```

uribus IMP 1301      PLURIBUS V2.9B 25-Jun-87 10:57:29      PAGE 30
IMP.VARS;9          PAGE 22           IMP Variables

;imp system tables

;SPF Routing Tables (used by SPF routines only)

    page V2

;tables by line

82CO      table ltb                      ;line topology info
82CO      1tbdst: .blk 1                  ;distance across this line
        dlinf = HFF                     ;infinite distance (dead lines)
OOFF      1tnbay: .blk 1                  ;neighbor IMP number*2
82C1      .blk nline-1
82C2      endtable ltb

85EO      table lflag                   ;line use flags
CO00      lflg = HCO00                  ;overlay NTB,LST,PDIST }}
8000      lflgf = H8000                 ;just take left 2 bits
CO00      lflgb = HCO00                 ;:sign ;forward line
0000      lflgu = 0                     ;:sign ;backward line
        :used = H3FFF                  ;unused line
        :bits for following tables
endtable lflag

;tables by IMP

85EO      table ntb.words              ;table of INDEX values
85EO      .blk nimp                   ;indexed by IMP*2
        ;used = HCO00
        ;unused = H3C00
        ;LFLG
O3FE      ntbindx = H3FE                ;offset for this node's lines
        ;unused = 1
86DE      .blk 1                      ;end of last IMP's lines
endtable ntb
86DE      ntbindi = .-2                 ;ptr to last word in table

86EO      table 1st.words              ;table of 1st structure stuff
        ;indexed by IMP*2
        ;various flags:
        ;LFLG
0002      .blk 1                      ;on-list flag ONLIST
0001      1stol = 2                   ;on-subtree flag ONSUBT
86E1      .blk 1                      ;list pointer by IMP*2
0001      1st1st = 1                 ;use odd bytes only
86E2      .blk nimp-1
endtable 1st

87DE      table pdist.words            ;path distance to each IMP
87DE      .blk nimp                  ;indexed by IMP*2
        ;used = HCO00
        ;LFLG
3FFF      pdst = H3FFF                ;path distance to this IMP
3FFF      pdinf = H3FFF                ;unreachable IMP value
endtable pdist

```

```
.endc  
; ; endtable 1flag ; it could reach here
```

uribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 31
 IMP.VARS;9 PAGE 23 IMP Variables

```

      Page V2
;just some space for v2 vars
defpage v2pat, org .., limit trmb1k

8910 ;assign space for transaction blocks
     .=<.+ HE>& HFFFO      ;hosts read into trmb1ks
     trnnum= D100
     0064
     0640
     trntot=trn1*trnnum
     8910 trnblk: .b1kb trntot

;input padding scratch area
8F50 .=<.+ HE>& HFFFO
8F50 hipad: .b1kw himaxp    ;garbage input from hosts

      Page Vars
;Define a page for assigning new variables.
defpage Varpat, org .., limit rmb1ks
656E     .b1kw H20          ;variables patch area

;message blocks

0038     rmnum= D56
0038     tmnum= D56

65B0     .=<.+ HE>& HFFFO

rmb1ks:
locdef rmlock,<;(and every 020) rcv mes block locks - page 31>
65B2     rmimp: .b1kw 1 ;source IMP#, <0 -> free block
65B4     rmhost: .b1kw 1 ;remote,,local hosts
65B6     rmctl: .b1kw 1 ;handling,fuse,,tmb1k
65B8     rmmess: .b1kw 1 ;messno,,luse,age
65BA     rstate: .b1kw 1 ;eight two bit message states
65BC     rmtype: .b1kw 1 ;eight corresponding two bit types
65BE     .b1kb 1 ;unused
65BF     rmlhn: .b1kb 1 ;offset of host in h2pb1k
     0010     rmlen=-rmb1ks
65C0     .b1kb rmlen*<rmnum-1>

tmb1ks:
locdef tmlock,<;(and every 020) xmit mes block locks - page 31>
6932     tmimp: .b1kw 1 ;dest IMP#, <0 -> free block
6934     tmhost: .b1kw 1 ;remote host,,local host
6936     tmctl: .b1kw 1 ;handling,foreign use,,rmb1k
6938     tmmess: .b1kw 1 ;messno(subtype if dead),,local use,age
693A     tstate: .b1kw 1 ;rset,init,tmail1,,message bits
693C     tmdeds: .b1kw 1 ;saved dead host status
693E     tmstp: .b1kb 1 ;tmstop,tmalt
693F     tmlhn: .b1kb 1 ;offset of host in h2pb1k
     0010     tmlen=-tmb1ks
6940     .b1kb tmlen*<tmnum-1>
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 32
IMP.VARS;9 PAGE 24 IMP Variables

```
;***bits in rmmess,tmmess
0100 mess0= H100      ;low bit of messno
FF00 messno=mess0* HF ;message no: last sent(t) or highest to rec(r)
0010 luse0= H10      ;local use bit
OOFO luse= HF*luse0 ;local use no.
0001 age0=1
000F age= HF+age0      ;timeout bits

;***bits in tstate
8000 tmrset=100000 ;reset in progress or acquisition error(if init)
4000 tminit=40000   ;just acquired, waiting got
0400 tmall0= H400    ;low bit of allocate count
3C00 tmall= HF*tmall0 ;allocate count
0300 msto= H300     ;incomplete timeout clock bits
0100 msto0=msto&~msto ;incomplete timeout, bit 0

;***bits in tmstp
0080 tmstop= H80     ;got destination dead
003F tmalt= H3F     ;set by tallyg, times alloc, then gives back
0001 tmalt0=~tmalt&tmalt

.comnt |
bits in rstate,rmtyp
rstate, rmtyp each have 8 2-bit fields, with the following values
rstate: 0=idle 1=request 2=message 3=reply
rtype: 0 rfnm sent illegal mes going send rfnm
      1 r/al sent req8 recd givb recd send r/al
      2 dead sent send all11 dead recd send dead
      3 incr sent send all18 incr recd send incr
|
0018 nrb1k= D24
6CBO messtk: .b1kb 0
          locdef ,<:reas blk lock (and every H10) - page 32>
6E30 .=,-2+<1rb1k*nrb1k>
emssstk:
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.VARS;9 PAGE 25 IMP Variables

PAGE 33

```
ttyiob: ;this block used by TTY
6E30    iob13: .b1kb lockfd-foo
        locdef ,<;Fake 0 DOZE lock - page 33>
        locdef ,<;Fake 0 WAIT lock - page 33>
6E48    .b1kb ioblen+foo-lockfw

ddtiob: ;this block for DDT fake host
6EB2    iob14: .b1kb lockfd-foo
        locdef ,<;Fake 1 DOZE lock - page 33>
        locdef ,<;Fake 1 WAIT lock - page 33>
6ECA    .b1kb ioblen+foo-lockfw

coriob: ;this block used by core transfer
6F34    iob15: .b1kb lockfd-foo
        locdef ,<;Fake 2 DOZE lock - page 33>
        locdef ,<;Fake 2 WAIT lock - page 33>
6F4C    .b1kb ioblen+foo-lockfw

staiob: ;stats IO block
6FB6    iob16: .b1kb lockfd-foo
        locdef ,<;Fake 3 DOZE lock - page 33>
        locdef ,<;Fake 3 WAIT lock - page 33>
6FCE    .b1kb ioblen+foo-lockfw

; statd starti endi stati
; starto endo stato hbpid/bhpid
; fakesi fakeso lockfd lockfw
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 34
IMP.VARS;9 PAGE 26 IMP Variables

;back host parameter blocks

bbk0:
703A locdef ,<;back host 0 (back5) lock - page 34>
.blk baken-2

bbk1:
705A locdef ,<;back host 1 (back7) lock - page 34>
.blk baken-2

bbk2:
707A locdef ,<;back host 2 (back9) lock - page 34>
.blk baken-2

bbk3:
709A locdef ,<;back host 3 (back6) lock - page 34>
.blk baken-2

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 35
 IMP.VARS;9 PAGE 27 IMP Variables

```

;fake host parameter blocks

70C0      .=<,+ HE>& HFFFO          ;16-byte boundary

bhi0:
locdef ,<;hi host lock fake 0 - page 35>
.b1kb lockih-lockhi-2
locdef ,<;ih software lock fake 0 - page 35>
.b1kb ihloc-lockih-2
locdef ,<;ih hardware lock fake 0 - page 35>
.b1kb ihloc-lockih-2
.=bhi0+holen

bhi1:
locdef ,<;hi host lock fake 1 - page 35>
.b1kb lockih-lockhi-2
locdef ,<;ih software lock fake 1 - page 35>
.b1kb ihloc-lockih-2
locdef ,<;ih hardware lock fake 1 - page 35>
.b1kb ihloc-lockih-2
.=bhi1+holen

bhi2:
locdef ,<;hi host lock fake 2 - page 35>
.b1kb lockih-lockhi-2
locdef ,<;ih software lock fake 2 - page 35>
.b1kb ihloc-lockih-2
locdef ,<;ih hardware lock fake 2 - page 35>
.b1kb ihloc-lockih-2
.=bhi2+holen

fakeh3:
bhi3:
locdef ,<;hi host lock fake 3 - page 35>
.b1kb lockih-lockhi-2
locdef ,<;ih software lock fake 3 - page 35>
.b1kb ihloc-lockih-2
locdef ,<;ih hardware lock fake 3 - page 35>
.b1kb ihloc-lockih-2
.=bhi3+holen

;dynamic area starts here

000E      nmdblk=<m2-.>/modlen    ;max modems per IMP
001E      nhoblk=<m2-.>/holen     ;max hosts per IMP

;Macro to patch dynamic allocator

.macro $finimp
$dopatch dyinit,fakcode
<%vars+ Hf>& HFFFO+ H20      ;give 250 for pkgs
.endm

8000      dyblk=1%vars           ;eats up rest of page

```

;IMP PID assignments
;BASE and MBLKS defined in OPSYS.PLR

```
: 0 emty,rstgo,bltcal,jdsply,dozew,dozew,dozew,dozew
: 10 btc,jjddt,jjtty,ihtc,waitw,waitw,waitw,waitw
: 20 tbfrly,ttochk,back,back,back,back,back,con
: 30 4*hi (fakes), 4*ih (fakes)
: 40 tsk,toss, rutpi, 5*hi (VDH/TIP)
: 50 5*hi (VDH/TIP), 3*ih (VDH/TIP)
: 60 7*ih (VDH/TIP), bad

;70-F2 - assorted hardware devices
;F4-F6 - qto (1.6 ms clock)
; FC   - nopids (all real PIDs are empty)
; FE   - unused
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 37
IMP.VARS;9 PAGE 29 IMP Variables

page V2 ;buffer variables on V2 page

8F70 .=<.+ HE>& HFFFO ;must be valid boundary for buffers
0350 mnbuf2=<<m3-.>/5>& HFFFO ;allocate 5 equal-size tables
; on good buffer boundaries

8F70 point: .b1kb mnbuf2
92C0 chain: .b1kb mnbuf2
;contents of chan vary depending on where buffer is used:
; if i2m, output channel number (0-maxchn, by twos)
; if m2i, 0 (used as a flag to task, see hi)
; if ih, time message was queued (1st pkt only)
; if hi, input hi block address (for task)
; if reas blk, packet number (see oldmes)

9610 chan: .b1kb mnbuf2
;contents of where indicate buffer usage(s) and source cntr
; transitions are done by eorm with nf locked
; zero in where indicates the buffer is free

8000 whm2i= H8000 ;filling
4000 whf2h= H4000 ;fhcqbf,fhcq,fhcbsf,sb1k
2000 whhi= H2000 ;hispc,trpack,bsend
1000 wthph= H1000 ;tp2h (PTIP)
0800 whv2h= H0800 ;VDH input
0400 whh2v= H0400 ;VDH output
0200 whhtp= H0200 ;h2tp (PTIP)
0100 whi2m= H0100 ;sentq,sending,reqq,priq
0080 whrut= H0080 ;rutq,rutbuf (could use whm2i)
0040 whih= H0040 ;(ihwq) hq,hpq
003E whctr= H003E ;source counter
0001 whtsk= H0001 ;tskbuf,rq,tq,reass

9960 where: .b1kb mnbuf2
9C80 flushd: .b1kb mnbuf2
A000 .=. ;end of buffer tables

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29
IMP.MAIN;1 PAGE 7.1 IMP Variables

PAGE 38

.INSERT "HACCON"
.INSRT HACCON

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 39
HACCON.PLR;1 PAGE 1 Configuration and Host Access

439

.stitle Configuration and Host Access

0001 .if nz VHA

.comnt |

Note: VHA lives on the DDT page.

| FC00 0200 PAGE VHACode

;VHA table for PSE
;Assemble a default identity mapping

483E 0001 vhaser: 1 ;first table (it's pre-loaded)
4840 0004 vhalen: D4 ;full VHA map

4842 table vhatab ;forward translation table
4842 0000 0 ;Never a VHA 0
4844 0001 1 ;Put NOC in initially
01FE .rept D512-2 ;for all other VHAs
 .xlist ;why look
 0 ;make an empty table
 .list ;okay, now look
 .endr
.endtable vhatab

.endc ;VHA

;hacmem and haccom words

FC00 0400 Page Warm
40B8 7E80
40BA 0000
40BC 5C9C
40BE 0004

40C0 0004 hacmem: 4,4,4 ;one real default, 2 specials
40C2 0004
40C4 0004
 0001 .lif nz PSE
40C6 0007 7 ;PLATFORM NOC, IMP 1, HOST 0 (3 Nov 82)
 0001 .lif z PSE ;but ARPANet has raw packets
 5 ;CCA Host 1: Speech Host
40C8 FFFC 0177774, 0100000.2, 0177777 ;fake hosts 0-3
40CA 8000
40CC 0002
40CE FFFF

40D0 0000 haccom: 0,0,0,0 ;one real default, 3 specials
40D2 0000
40D4 0000
40D6 0000
40D8 0000 0,0,0,0 ;four fakes
40DA 0000
40DC 0000
40DE 0000

40E0 FFFF hacspc: -1,-1,-1 ;indices of special hosts
40E2 FFFF
40E4 FFFF
 0001 .lif nz PSE ;no default specials in Platform
40E6 00 .byte ncc_-6,ncc&desti ;NU gets raw pkts, pkt core
40E7 01
 0001 .lif z PSE ;ARPANet has one
 .byte 1*words,CCA ;CCA Host 1 is special
40E8 F8 .byte H100-<4*words>, HFF ;TTY Fake
40E9 FF
40EA FA .byte H100-<3*words>, HFF ;DDT Fake
40EB FF
40EC FC .byte H100-<2*words>, HFF ;Pkt Core Fake
40ED FF
40EE FE .byte H100-<1*words>, HFF ;Statistics Fake
40EF FF
 0010 1haccs=-hacspc ;length of hacc search
 ;special haccoms: 2 for system a,e;
 ; 2 for 30,40 tty fake (0)

0001 .if z PSE
concks: H8DDA ;checksum on all of the above from hacmem
 .iff ;;nz PSE
concks: H8FFG
.endc

0032 conlen=-hacmem ;length to checksum
002F .1if lt D72-<conlen/2>
.error host access tables too long

40F2 hacend: .b1kw 0

;Dispatch tables for host functions
;see HOSTYP, HOSREAL, HOSFAKE, etc.
;these table are kept in LOCAL but are called by routines
;on common. The proper usage should be to call routines on
;the same page as the caller or else local.

page LCode

;host input subroutines
12BE table hitable
12BE 0ED6 xsioin ;real host gets real input
12CO 15D4 fhsioin ;fake IOBLOC input
12C2 15D4 fhsioin ;VDH IOBLOC input
12C4 15D4 fhsioin ;TIP IOBLOC input
12C6 07D6 rsucceed ;spare
12C8 07D6 rsucceed ;spare
endtable hitable

;host output subroutines
12CA table hotable
12CA 5DA6 hsiout ;real host output - checks
12CC 5DBE fsiout ;fake host output in IOBLOC
12CE 5DBE fsiout ;VDH host output in IOBLOC
12DO 5DBE fsiout ;TIP host output in IOBLOC
12D2 07D6 rsucceed ;spare
12D4 07D6 rsucceed ;spare
endtable hotable

```
;line logic control words

FC00 0400      page PkgCode

0001      .if z PSE          ;all these for ARPANET
           ;initial line speed set,,routing interval (med. ticks-1)
           table lineprot
               .byte 4,4      ;default all lines to 50 kbit
               .byte 4,4      ; and 1 slow tick to route ,
               .byte 4,4
               .byte 2, D9    ;:SDAC ;SDAC line 4 to Norway
               .byte 0, D24    ;:SDAC ;SDAC line 5 to London
               .byte 4,4
               .byte 4,4
               .byte 4,4
           endtable lineprot

           .iff   ;:z PSE        ;use these for PSE

           ;initial line speed set,,routing interval (med. ticks-1)
           table lineprot
               .byte kb64,4    ;default all lines to 50 kbit
               .byte kb64,4    ; and 1 slow tick to route
               .byte kb64,4
               .byte kb64,4
               .byte kb64,4    ;10 Kbit, 2 slow ticks to route
               .byte kb64,4
               .byte kb64,4
               .byte kb64,4
               .byte kb64,4
               .byte kb64,4
           endtable lineprot

           .endc ;:z PSE
```

```
;maximum channels by use on 1ines, indexed by 1ine number.  
4102      table channel  
        0001    .if z PSE  
                  D8*words          ;8 chan for 1ines 1-4  
                  D8*words  
                  D8*words  
                  D8*words  
        .iff      D16*words          ;PSE is all 16 channels.  
4102 0020  D16*words          ;16 chan for 1ines 1-4  
4104 0020  D16*words  
4106 0020  D16*words  
4108 0020  D16*words  
        .endc  
410A 0100  NMDL1m*words       ;High 4 1ines are satellites  
410C 0100  NMDL1m*words       ; for both networks.  
410E 0100  NMDL1m*words  
4110 0100  NMDL1m*words  
        endtable channel  
  
;Init retransmit time based on 1ine number.  
4112      table irrtime       :initial retransmit time  
4112 03E8  rrtme            ;nominal 100 ms retransmit time  
4114 03E8  rrtme            ;nominal 100 ms retransmit time  
4116 03E8  rrtme            ;nominal 100 ms retransmit time  
4118 03E8  rrtme            ;nominal 100 ms retransmit time  
411A 3E80  D16*rrtme         ;special 1.6 sec retransmit time  
411C 3E80  D16*rrtme         ;special 1.6 sec retransmit time  
411E 3E80  D16*rrtme         ;special 1.6 sec retransmit time  
4120 3E80  D16*rrtme         ;special 1.6 sec retransmit time  
        endtable irrtime  
  
;initial Nup counter,, medium ticks per logical tick  
4122      table protcnt  
        3C      .byte  D60, D24 ;4.8 Kbit: 60 up, 5 slow ticks  
4123 18     .byte  D60, D9   ; 10 kbit: 60 up, 2 slow ticks  
4124 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4125 09     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4126 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4127 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4128 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4129 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412A 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412B 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412C 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412D 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412E 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
412F 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4130 3C     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
4131 04     .byte  D60, D4   ; 50 kbit: 60 up, 1 slow tick  
        endtable protcnt  
  
;initial k,,n values for k-out-of-n 1ine-down  
4132      table knvalu  
        05      .byte  D5, D5   ;4.8 kbit: 5 out of 5  
4133 05  
        00C    .if nz PSE
```

```
4134 05      .byte D5, D5 ; 10 kbit: 5 out of 5
4135 05
4136 05      .byte D5, D5 ; 50(or 128) kbit: 5 out of 5
4137 05
        .iff :;nz PSE
            .byte D4, D20 ; 10 kbit: 4 out of 20
            .byte D4, D20 ; 50 kbit: 4 out of 20
        .endc :;nz PSE
4138 05      .byte D5, D5 ;250 kbit: 5 out of 5
4139 05
413A 05      .byte D5, D5 ;250 kbit: 5 out of 5
413B 05
413C 05      .byte D5, D5 ;250 kbit: 5 out of 5
413D 05
413E 05      .byte D5, D5 ;250 kbit: 5 out of 5
413F 05
4140 05      .byte D5, D5 ;250 kbit: 5 out of 5
4141 05
        endtable knvalu

        page LCode

table propd1          ;propagation delay 800us units
```

```
12D6
12D6 0001      .word 1          ;default is 1
12D8 0001      .word 1          ;should be 4 for sdac-cca line
12DA 0001      .word 1
12DC 0001      .word 1
12DE 014C      .word D332
12EO 014C      .word D332
12E2 014C      .word D332
12E4 014C      .word D332
        endtable propd1
FC00 0400      page Rutcode

4142      table bias
4142 0001      .word 1          ;Routing bias for each line; 6.4 ms units
4144 0001      .word 1
4146 0001      .word 1
4148 0001      .word 1
414A 0001      .word 1
414C 0001      .word 1
414E 0001      .word 1
4150 0001      .word 1
        endtable bias

4152      table delmax           ;Maximum allowed delay by line
4152 00FE      .word d1inf-1
4154 00FE      .word d1inf-1
4156 00FE      .word d1inf-1
4158 00FE      .word d1inf-1
415A 00FE      .word d1inf-1
415C 00FE      .word d1inf-1
415E 00FE      .word d1inf-1
4160 00FE      .word d1inf-1
        endtable delmax
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 45
HACCON.PLR;1 PAGE 6 Configuration and Host Access

447

```
page HLCode

;some host parameters

12E6 0008    pkts8:   D8      ;packets in a multi
12E8 0030    lngall:  D48     ;extra allocate timer setting (1.2288 sec)
1ngimp:
0001          .if nz PSE      ;all IMPs in PSE get long allocates
12EA FFFF    -1
.iff ;:nz PSE
           sdac    ;which IMP gets to use it
.endc ;:nz PSE
```

; IMP configuration status.

F000 0600 Page Fakcode

40B8 7E00

40BA 5792

40BC 4E42

40BE 0006

 ;define package bits for pakior

0001 vdhibit = 1 ;VDH

0002 tipbit = 2 ;TIP

0004 expbit = 4 ;experimental package

0008 cmsbit = H8 ;cumulative statistics

0010 trabit = H10 ;trace

0020 ttybit = H20 ;TTY

0040 ddtbit = H40 ;full DDT

0080 sfsbit = H80 ;store and forward statistics

0100 eesbit = H100 ;end-to-end statistics

0200 strbit = H200 ;strip time measurements

0000 pakior = 0 ;initially, no packages

0000 .1if nz VDHSw ;if the VDH is being assembled in

 pakior = pakior+vdhibit ;report the VDH

;Some patchable configuration constants

40C0 0000 tipver: 0 ;tip version #

40C2 02C1 sysver: hsmver ;imp system version

40C4 00 sshost: .byte ncc_-6 ;send empty setup sends to NU

40C5 01 ssimp: .byte ncc&desti

40C6 01 nccimp: .byte ncc&desti ;NCC IMP number

40C7 00 ncchst: .byte ncc_-6 ;NCC host number

40C8 F003 ncclnk: < 0360_ D8>+rawpkt ;NU link, raw pkts

40CA 0000 cpflgs: pakior ;package flags (VDH, TIP etc.)

40CC 128E qtoaddr: do1.6 ;dispatch for highest priority 1.6 ms clock

40CE 113C ttoaddr: loop3 ;dispatch for a routine to call every 25 ms

40D0 0000 intpti: 0 ;nominal setting for counter ntipi

40D2 0000 intpit: 0 ;ditto for ntipit

40D4 0016 intpsh: numhst ;2* server hosts in use (PTIP only)

40D6 0000 dyinit: 0 ;defined and set up by finimp.

TEMREF:

40D8 80 .BYTE D32*4 ;E-BUS, SENSOR 1

40D9 80 .BYTE D32*4 ;E-BUS, SENSOR 2

40DA 80 .BYTE D32*4 ;E-BUS, SENSOR 3

40DB 80 .BYTE D32*4 ;E-BUS, SENSOR 4

40DC 80 .BYTE D32*4 ;F-BUS, SENSOR 1

40DD 80 .BYTE D32*4 ;F-BUS, SENSOR 2

40DE 80 .BYTE D32*4 ;F-BUS, SENSOR 3

40DF 80 .BYTE D32*4 ;F-BUS, SENSOR 4

;table of legal overrides

FC00 0200

Page DDTCode

```
4C42      table ovrtab
    0001    .1if z PSE           ;extra override in ARPANet
            .byte HFC,bbnG3     ;backroom prototype 516
    4C42    00    .byte n10xh,n10xi ;NCC TENEX system
    4C43    00    .byte 0.0        ;us before IMP comes up
    4C44    00
    4C45    00
endtable ovrtab
```

:VDH package linkage and control

Page LCode

```
.1if ndf c.vd      :VDH clip
  c.vd= HCO        :VDH modems start here.

.
.1if ndf o.vd      :VDH offset
  o.vd= HCO

12EC  CO  vd.clp: .byte c.vd    ;modems above clip are VDH
12ED  CO  vd.off: .byte o.vd    ;subtract from modem # to get host
```

Page Lcode

:Dispatch Tables for CONFIG

```
;vdhlinkages
12EE 07D6  icvdh:     rsucceed

12FO          table chostest           ;routines to call for V2PBLK
12FO 07D8    rfail      ::hosreal   ;no real hosts in V2PBLK
12F2 07D6    rsucceed   ::hosfake   ;no checking for fakes
12F4 07D8    rfail      ::hosvdh   ;initially no VDHs
12F6 07D8    rfail      ::hostip    ;initially no TIPs
12F8 07D8    rfail      ::spare
12FA 07D8    rfail      ::spare
endtable chostest

12FC          table chosini            ;routines if checks (CHOSTEST) fail
12FC 07D6    rsucceed   ;none for reals
12FE 07D6    rsucceed   ;none for fakes
1300 07D6    rsucceed   ;none for VDHs yet
1302 07D6    rsucceed   ;none for TIPs yet
1304 07D6    rsucceed   ;spare
1306 07D6    rsucceed   ;spare
endtable chosini
```

FC00 0000 Page RelCode
:tables to drive BASE/MBLKS checking; parallel to each other

```
5650          table ccbbase           ;BASE entry to match
5650 4262    tohot
5652 205A    m2i
5654 1D3A    i2m
5656 2B46    hi
5658 18DC    ih
565A          ccbbspare = ccbbase+,
               -1,-1,-1,-1,-1,-1       ;6 spares (2 VDH, 4 other)

565A FFFF
565C FFFF
565E FFFF
5660 FFFF
5662 FFFF
5664 FFFF
endtable ccbbase
```

5666 table ccheck ;routine to call for match

```
5666 5814      rtcchk          ;RTC dispatch checker  
5668 583C      modchk,modchk ;modem dispatch checker  
566A 583C  
566C 57FO      hoschk,hoschk ;host dispatch checker  
566E 57FO  
5670            cchspare = ccheck+. -1,-1,-1,-1,-1,-1  
5670 FFFF  
5672 FFFF  
5674 FFFF  
5676 FFFF  
5678 FFFF  
567A FFFF  
endtable ccheck
```

Pluribus IMP 1301 PLURIBUS V2.9B 25-Jun-87 10:57:29 PAGE 48
IMP.MAIN;1 PAGE 7.2 Configuration and Host Access

.INSERT "IMPSUB"
.INSRT IMPSUB