Index

A

Abort entry point
  ddINT-2, 1–26
  explained, 1–25
  from .DRAST, 1–25
  with abort processing, 1–43
Abort processing
  internally-queued handlers, 1–48
  order of action taken by RMON, 1–46
Abort processing for handlers, 1–43
ABTIO$
  bit in device status word, 1–11
  defined by .DRDEF, 1–10
  relationship to HNDLR$, 1–45
  with abort processing, 1–43
.ABTIO programmed request
  action by RMON at abort by, 1–44
  with abort processing, 1–44
Address translation, 1–71
  $JBREL subroutine, 1–79
  $MPMEM routine, 1–71
ALLUMR routine, 2–84
  .DRDEF macro with, 2–85
  $H2UB fixed offset with, 2–85
Asynchronous directory operations to magtapes, 2–44

B

Bad-block replacement, 1–68
  clearing soft errors, 2–11
  forcing with FORMAT/VERIFY:ONLY, 2–11
  for DU handler, 2–10
    described, 2–11
Bad-block replacement, software
  See Software bad-block replacement
Bad sector error (BSE)
  DM handler with, 2–6
BBR.BD, 1–36
BBR.GD, 1–36
BBR.SZ, 1–36
BIOERR:, 1–86
$BLKMV subroutine, 1–81
BOOT keyboard command
  operation, 1–87, 1–88
/BOOT option
  COPY keyboard command operation, 1–87
Bootstrap
  discussion, 1–83
  error routine, 1–86
  part of primary driver, 1–85
  read routine, 1–85
  role of DUP program, 1–86
  use to install handlers, 1–103
Bootstrap, software, 1–85

C

C.COMP, 1–56
C.HOT, 1–56
C.JNUM, 1–56
C.LINK, 1–56
C.LOT, 1–56
C.SEQ, 1–56
C.SYS, 1–56
.CHAIN programmed request
  action by RMON at abort by, 1–44
.CLOSE programmed request
  after .ENTER with magtapes, 2–40, 2–42
  after .LOOKUP with magtapes, 2–43
  magtape hardware handlers, 2–50
.CLOSZ programmed request
  invalid with magtapes, 2–42
Communications handlers
  See also XC and XL handler
  annotated source listing of XL, A–56
  detecting EOF, A–1
  disabling interrupts for, 2–96
  obtaining handler status, 2–96
  performing a special read, 2–95
  .READx and .WRITx support for, 2–94
  setting/clearing BREAK bit, 2–95
  setting/clearing DTR signal, A–1
  when XOFF received, 2–95

Condition codes
used in .DRVTB macro, 1–18
.CSTAT programmed request
extended device-unit support, 1–102
.CTIMIO macro
described, 1–57
CTRL/C
action by RMON at abort by, 1–44

D

Data transfers
See Transferring data
dd$END, 1–115
dd$NAM, 1–97
dd$PN2, 1–97
dd$VEC
defined by .DRVTB, 1–18
ddBOOT, 1–84
ddBOT, 1–115
ddDSIZ, 1–11
ddEND, 1–115
ddINT
defined by .DRVTB, 1–18
ddINT-2, 1–26
DECFKM, 2–77
DECTCEM, 2–78
.DELETE programmed request
invalid with magtapes, 2–42
Device handler macros
.DRAST, 1–25
.DRBEG, 1–17
.DRBOT, 1–83
.DRDEF, 1–5
.DREND, 1–29
.DREST, 1–11
.DRFIN, 1–29
.DRINS, 1–12
.DRPTR, 1–13
.DRSET, 1–14, 1–51
.DRSPF, 1–13
.DRTAB, 1–13
.DRUSE, 1–14
.DRVTB, 1–17
Device handlers, 1–1
abort entry point, 1–25
abort processing, 1–43
accessing physical memory, 1–71
accessing user buffer directly, 1–75
address translation, 1–71
Device handlers (Cont.)
assembling, 1–102
bad-block replacement, 1–68
block 0 information, 1–14
.DRDEF, 1–5
.DREST, 1–11
.DRINS, 1–12
.DRPTR, 1–13
.DRSET, 1–14
.DRTAB, 1–13
.DRUSE, 1–14
block 1 information, 1–19
contents of .SYS image, 1–113
.CTIMIO with, 1–57
data structures, 1–30
relating to block 0, 1–30
relating to block 1, 1–39
defined, 1–1
device I/O timeout, 1–55
applications, 1–58
.DRAST with, 1–25
.DRBEG with, 1–17
.DRDEF with, 1–5
.DREND with, 1–29
.DREST with, 1–11
.DRFIN with, 1–29
.DRINS with, 1–12
.DRPTR with, 1–13
.DRSET with, 1–14, 1–51
.DRSPF with, 1–13
.DRTAB with, 1–13
.DRUSE with, 1–14
.DRVTB with, 1–17
editing SYSTBL.MAC, 1–106
error logger with, 1–61
example skeleton device handler, 1–41
extended device-unit structure (H.64UM), 1–38
file image block 0, 1–5
file image block 1, 1–16
flag word (H1.FLG), 1–39
.FORK with, 1–26
$GTBYT routine with, 1–72
I/O completion section
.DRFIN with, 1–28
if error, 1–28
if successful, 1–28
in mapped systems, 1–69
addressing user buffer, 1–70
installation code, 1–12

Device handlers (Cont.)
installation verification routines, 1–108
installing, 1–103
bypassing hardware requirement, 1–110
precedence, 1–103
requires device hardware, 1–107
with INSTALL command, 1–104
with the bootstrap, 1–103
internal data descriptor table, 1–37
internal queueing of I/O, 1–47
interrupt entry point, 1–24
interrupt service section, 1–24
coding guidlines, 1–26
complete transfer, 1–27
incomplete transfer, 1–27
linking, 1–103
lowering priority after interrupt, 1–26
$MPMEM routine with, 1–71
$MPPHY routine with, 1–72
multiterminal handler hooks support in, 1–89
multivector, 1–17
naming conventions, 1–69
PIC code, requires, 1–3
planning, 1–1
preamble macros, 1–4
pseudodevices, 1–29
$PUTBYT routine with, 1–73
$PUTWRD routine with, 1–74
queue element offsets, 1–7
routines for SET options, 1–52
second flag word (H1.FG2), 1–40
Second handler status word, 1–36
service routines, 1–30
.DRPTR with, 1–13
entry environment, 1–30
entry point word (H1.NOP), 1–39
error processing, 1–32
SET options for
enabling, 1–49
example routines, 1–53
software bad-block replacement geometry table, 1–35
Software bad-block replacement table, 1–36
source listings of
DL handler, A–23
DX handler, A–1
XL handler, A–56
Device handlers (Cont.)
special directory devices, 1–68
if error, 1–69
SPFUNs with, 1–69
special functions, 1–65
special functions, defining for, 1–13
special functions, finding names for, 2–2
special functions, supporting, 1–67
SPFUN support table, 1–34
.SPFUN with, 1–65
structure, 1–3
structure of
abort entry point, 1–25
handler termination section, 1–29
I/O completion section, 1–28
I/O initiation, 1–19
interrupt service section, 1–24
preamble, 1–4
SYSGEN conditionals, 1–7
SYSGEN options byte (H.GEN), 1–37
SYSMAC.SML with, 1–4
SYSTEM.MLB with, 1–4
system devices, 1–83
creating, 1–83
testing and debugging, 1–110
DBG–11 recommended for, 1–110
.TIMIO with, 1–55
transferring data
byte transfer, 1–21
$GTBYT routine, 1–22
initiating, 1–19
$MPMEM routine, 1–24
non-DMA, 1–24
$PTBYT routine, 1–22
$PTWRD routine, 1–23
word transfer, 1–23
UMR support structure (H.64UM), 1–38
variable-size volumes, 1–67
writing, 1–1
Device handlers,specific
See individual names
Device identifier byte, 1–9
Device size word
described, 1–11
Device status word
described, 1–10
Device timeout
applications, 1–58
disk handlers, 1–59
line printer, 1–60

Device timeout
applications (Cont.)
multiterminal service, 1–58
.CTIMIO with, 1–57
discussion, 1–55
timer queue element format, 1–56
.TIMIO with, 1–55
DEV macro
described, 1–106
Directory structure
special directories, 1–68
DISCSR
defined by .DRINS, 1–12
Display CSR, 1–12
DL handler
annotated source listing for, A–23
bad-block replacement for, 2–3
bad-block replacement geometry table (H.REPL), 1–35
Bad-block replacement table for, 1–36
programming information for, 2–3
DMA support
UNIBUS, See UMR and UB handler
DM handler
bad-block replacement geometry table (H.REPL), 1–35
Bad-block replacement table for, 1–36
programming information for, 2–5
.DRAST macro
described, 1–25
.DRBEG macro
described, 1–17
.DRBOT macro
described, 1–83
.DRDEF macro
calls .QELDF, 1–7
for extended device-unit support, 1–97
format, 1–6
in a device handler, 1–5
UNIBUS DMA handlers, 2–82
UNIBUS non-DMA handlers, 2–82
variable-size device, for a, 1–68
.DREND macro
described, 1–29
system devices, with, 1–84
.DREST macro
described, 1–11
.DRFIN macro
canceling .TIMIO requests, 1–59
described, 1–29
.DRINS macro
defines installation entry point, 1–12
described, 1–12
installation verification routine, 1–109
.DRPTR macro
described, 1–13
.DRSET macro
described, 1–51
examples of, 1–53
.DRSPF macro
described, 1–13
.DRTAB macro
with STATU$ for DU handler, 2–9
.DRUSE macro
described, 1–14
.DRVTB macro
described, 1–17
DSE.8, 1–51
DSE.ES, 1–51
DSE.NA, 1–51
DSE.NO, 1–51
DSE.NU, 1–51
DSE.R3, 1–51
DSE.SB, 1–51
DSP.NN, 1–35
DSP.NX, 1–35
DSP.TY, 1–35
DSP.Xn, 1–34
.DSTAT programmed request
variable-size device, for a, 1–68
DT.EOL, 1–38
DT.ESZ, 1–38
DT.ID, 1–38
DT.PTR, 1–38
DT.SIZ, 1–38
DU$PORTS=n
for multiple DU ports, 2–19
DU.ENT, 2–15
DU.ESZ, 2–15
DU.ID, 2–15
DU.NUM, 2–15
DU.PAR, 2–16
DU.POR, 2–16
DU.UNI, 2–16
DU handler
bad-block replacement
described, 2–11
support for, 2–10
block numbers illustrated, 2–20

DU handler (Cont.)
bypassing unit number translation, 2–17
configuring for multiple ports, 2–19
creating a second, 2–22
directly addressing MSCP port (bypass), 2–17
finding device status, 2–9
finding volume size, 2–8
MSCP controller port numbers, 2–19
MSCP partition numbers, 2–20
multiport booting, 2–23
non-file-structured read/write operations, 2–12
partition number, 2–16
physical MSCP unit number, 2–16
port controller number, 2–16
programming information for, 2–8
translation table, 2–14
accessing, 2–16
two ports illustrated, 2–21
DV2.DM, 1–39
DV2.NF, 1–39
DV2.NL, 1–39
DV2.V2, 1–39
DW handler
programming information for, 2–25
DX handler
annotated source listing of, A–1
programming information for, 2–26
DY handler
programming information for, 2–26
DZ handler
programming information for, 2–28

E

ELBLDR macro
described, 1–65
$ELPTR
pointer to error logger routine, 1–64
End-of-file condition reporting (magtapes), 2–49
.ENTER programmed request
and .CLOSE with magtapes, 2–42
and .PURGE with magtapes, 2–43
errors defined for magtapes, 2–38
on a special directory device, 1–69
sequence numbers defined for magtapes, 2–38
with magtapes, 2–38
ERL$G, 1–61
ERL$S, 1–61
ERL$U, 1–62
ERLG$, 1–37
Error logger
discussion, 1–61
ERL$G, 1–61
ERL$S, 1–61
ERL$U, 1–61
Error Logger
adding a device, 1–64
calling, 1–64
ELBLDR macro with, 1–65
hard errors, 1–63
logging successful I/O transfers, 1–62
register usage, 1–62
soft errors, 1–63
Errors
difference between hard and soft, 1–64
Ethernet handlers
See also NC, NQ, or NU handler
allocate/deallocate unit for a job, 2–66
enable/disable multicast address for a unit, 2–69
enable/disable protocol type for a unit, 2–67
example of allocating a unit, 2–74
programming information for, 2–65
receive Ethernet frame, 2–73
restrictions for, 2–65
transmit Ethernet frame, 2–71
.EXIT programmed request
action by RMON at abort by, 1–44
Extended device units, 1–97
.CSTAT with, 1–102
handler LOAD/FETCH routines with, 1–98
handler UNLOAD/RELEASE routines with, 1–98
linking the handler for, 1–98
$OWNER table with, 1–98
PSECT requirements for, 1–98
Q.FUNC with, 1–101
Extended memory
addressing user buffer from a device handler, 1–70
device handlers, 1–69
.FETCH support, 1–70
Extended memory subroutines, 1–78
$BLKMV, 1–81

Extended memory subroutines (Cont.)
FINDGR, 1–82
$JBREL, 1–79
$USRPH, 1–82
XALLOC, 1–81
XDEALC, 1–81

F

F.BADR
with handler abort, 1–26
.FETCH programmed request
in mapped monitors, 1–70
File structure module
See FSM
FILST$
bit in device status word, 1–11
defined by .DRDEF, 1–10
FINDGR, 1–82
.FORK macro
with handler abort, 1–26
with handler retries, 1–27
FORMAT/VERIFY:ONLY
forces bad-block replacement, 2–11
FSM
compatibility with magtape operations, 2–34
summary of, 2–34
described, 2–33
effect on UMR allocation, 2–82
search by file name, 2–37
search by sequence number, 2–36

G

$GETBYT routine
described, 1–72
GETUMR routine, 2–86
$H2UB fixed offset with, 2–86
.GFxxx programed requests
invalid with magtapes, 2–42
$GTBYT
pointer to $GETBYT routine, 1–72
$GTBYT routine
illustrated, 1–22

H

H.64UM, 1–16, 1–38
with extended device-unit support, 1–97
H.BLEN, 1–15
H.BPTR, 1–15
H.CLAS, 1–15
H.DATA, 1–16, 1–37
H.DCSR, 1–16
H.DLEN, 1–16, 1–37
H.DSIZ, 1–15
from .DRDEF, 1–11
H.DSTS, 1–15
H.FETC, 1–14
H.GEN, 1–15, 1–37
H.HAN, 1–14
H.HANV, 1–14
H.ICSR, 1–16
H.IDK, 1–16
H.ISY, 1–16
H.LOAD, 1–14
H.MOD, 1–15
H.READ, 1–15
H.RELE, 1–14
H.REPL, 1–15
structure of, 1–35
H.SET, 1–16
H.SIZ, 1–15
H.SPF1, 1–15
H.SPF2, 1–15
H.SPFX, 1–15
structure of, 1–34
H.STS2, 1–15, 1–36
H.TYPE, 1–15, 1–37
H.UNIT, 1–16
H.UNLO, 1–15
H.USER, 1–16
H1.ABT, 1–19
H1.BR, 1–19
H1.CQE, 1–19
H1.FG2, 1–19, 1–40
defined by H1.NOP, 1–39
H1.FLG, 1–19, 1–39
H1.HLD, 1–19
H1.LDT, 1–19
H1.LQE, 1–19
H1.NDF, 1–19
H1.NOP, 1–19, 1–39
H1.SCK, 1–19
H1.VEC, 1–19
Handler flag word, 1–39
second (H1.FG2), 1–40
HB.BAD, 1–36
Header validity error (HVRC)
DM handler with, 2–6

HF2.64, 1–40
HF2.DM, 1–40
HF2.F3, 1–40
HF2.LD, 1–40
HF2.ND, 1–40
HF2.S6, 1–40
HF2.SC, 1–40
HF2.SD, 1–40
HF2.SR, 1–40
HNDLR$
  bit in device status word, 1–11
  defined by .DRDEF, 1–10
  relationsip to ABTIO$, 1–45
  with abort processing, 1–43
HNP.F2, 1–40
HNP.FE, 1–39
HNP.LO, 1–39
HNP.N1, 1–40
HNP.N2, 1–40
HNP.RE, 1–39
HNP.UN, 1–39
.HRESET programmed request
  action by RMON at abort by, 1–44
HRR.AB, 1–32
HRR.FF, 1–32
HRR.LO, 1–32
HRR.RE, 1–32
HRR.SY, 1–32
HRR.UN, 1–32
HS2.BI, 1–36
HS2.KI, 1–36
HS2.KL, 1–36
HS2.KU, 1–36
HS2.MO, 1–36
HUM.64, 1–38
HUM.DM, 1–38
HUM.PU, 1–38
HUM.S6, 1–38
HUM.UB, 1–38

I

I.IOCT
  relationship to HNDLR$ and ABTIO$, 1–45
  with abort processing, 1–44
I/O
  device timeout, 1–55
    applications, 1–58
    .CTIMIO with, 1–57
I/O
  device timeout (Cont.)
    .TIMIO with, 1–55
  when serialized I/O satisfaction needed on UNIBUS, 2–81
I/O queue element offsets, 1–7
INSCSR
  defined by .DRINS, 1–12
INSDAT
  defined by .DRINS, 1–12
Installation CSR, 1–12
Installation verification routines
  bypassing hardware requirement, 1–110
  checking hardware registers, 1–109
  described, 1–108
  ID bit or bytes for, 1–109
  read/write bits for, 1–110
Installing handlers
  See Device handlers, installing
INSTALL keyboard command
  bypassing hardware requirement, 1–110
  discussion, 1–104
  requires device hardware, 1–107
Internal handler queues, 1–47
  abort procedures for, 1–48
  implementing, 1–47
  modified .DRFIN with, 1–47

J

$JBREL subroutine, 1–79
  .CALLK with, 1–80
JREAD function
  reads entire DU device, 2–12
JWRITE function
  writes to entire DU device, 2–12

K

KDA50 controller
  bad-block replacement for, 2–11
KLESI-QA controller
  bad-block replacement for, 2–10
KLESI-UA controller
  bad-block replacement for, 2–11

L

LD.ACT, 2–30
LD.FLG, 2–30
LD.ID, 2–30

LD.NAM, 2–31
LD.NDX, 2–30
LD.NUM, 2–30
LD.OFS, 2–30
LD.RDO, 2–30
LD.SIZ, 2–31
LD.UNT, 2–30
LD.UNX, 2–30
LD.UOF, 2–30
LD handler
  LDREL$ with, 2–32
  number of logical disks supported, 2–30
  programming information for, 2–29
  translation table
    physical device name, 2–31
  translation tables, 2–29
    accessing, 2–31
    flags, 2–30
    header, 2–30
    number of entries, 2–30
    size, 2–31
    size of, 2–29
    structure of, 2–30
    unit numbers, 2–30
  when handler is loaded or unloaded, 2–32
LDREL$
  CONFG2 bit, LD with, 2–32
.LOOKUP programmed request
  and .CLOSE with magtapes, 2–43
  and .PURGE with magtapes, 2–43
  errors defined for magtapes, 2–40
  file-structured on magtapes, 2–39
  non-file-structured with magtapes, 2–43
    error codes for, 2–44
    sequence numbers for, 2–44
  on a special directory device, 1–69
  sequence numbers defined for magtapes, 2–39

M

Magtape hardware handlers
  .CLOSE request with, 2–50
  described, 2–55
  installing, 2–55
  rewinding and going off line, 2–51
  rewinding from, 2–52
  spacing backward, 2–52
  spacing forward from, 2–53
  writing a tape mark, 2–54
Magtape hardware handlers (Cont.)
  writing with extended gap, 2–52
Magtapes
  asynchonous directory operations, 2–44
  bypassing the FSM, 2–43
  compatibility with FSM, 2–34
    summary of operations, 2–34
  emulating .ENTER with SF.USR, 2–44
  enabling 100ips streaming, 2–50
  end-of-file condition reporting, 2–49
  .ENTER request with, 2–38
  exception reporting for special functions, 2–48
  FSM, when incompatible with, 2–51
  FSM described, 2–33
  FSM search by file name, 2–37
  FSM search by sequence number, 2–36
  hard error condition reporting, 2–49
  incompatibility with FSM, 2–35
  invalid programmed requests with, 2–42
  .LOOKUP request (file-structured), 2–39
  non-file-structured .LOOKUP request with, 2–43
  opening a file on, 2–38
  programming information for, 2–33
  .PURGE request with, 2–43
  reading physical blocks, 2–47
  reading tapes from other systems, 2–56
  .READx requests with, 2–40
  repositioning can cause data loss, 2–40
  seven-track, 2–58
  spacing error recovery, 2–36
  TMSCP translation tables support, 2–60
  writing physical blocks, 2–48
  .WRITx requests with, 2–41
Mapped monitors
  .FETCH support, 1–70
Mapping
  $P1EXT routine, 1–75
MMG$T=1
  SYSGEN conditional for extended memory support, 1–69
MMGT$, 1–37
MM handler
  programming information for, 2–33
?MON-F-
  action by RMON at abort by, 1–44
$MPMEM routine
  described, 1–71
  illustrated, 1–24

$MPPHY routine
called by $MPPTR, 1–71
described, 1–71
$MPPTR
calls $MPPHY routine, 1–71
pointer to $MPPHY routine, 1–72
MSCP (Mass Storage Communications Protocol)
See DU handler
MSCP controller port numbers, 2–19
MSCP disk block numbers illustrated, 2–20
MSCP disks
addressing, 2–18
MSCP multiple port handler support, 2–21
MSCP partition numbers, 2–20
MSCP unit numbers, 2–18
MSCP unit numbers, ports, partitions
translation table for, 2–14
MS handler
programming information for, 2–33
MT handler
programming information for, 2–33
MU.ENT, 2–61
MU.ESZ, 2–61
MU.ID, 2–61
MU.JOB, 2–61
MU.NUM, 2–61
MU.POR, 2–61
MU.UNI, 2–61
MU handler
bypassing the translation tables, 2–62
compared to other magtape handlers, 2–60
physical TMSCP unit number, 2–61
programming information for, 2–60
size of translation table entry, 2–61
special function bypass for, 2–62
TMSCP port number, 2–61
TMSCP unit job number, 2–61
translation tables, 2–60
accessing, 2–62
size of, 2–60
structure of, 2–61
unit support, CSR and vectors for, 2–63
Multiport booting, 2–23
Multiterminal handler hooks support
connecting with monitor, 1–92
disconnecting with monitor, 1–96
installation code for, 1–91
interrupt processing, 1–95
Multiterminal handler hooks support (Cont.)
SET code for, 1–91

N

NC handler
See also Ethernet handlers
programming information for, 2–65
NL handler
programming information for, 2–64
Non-file-structured .LOOKUP request with magtapes, 2–43
NQ handler
See also Ethernet handlers
programming information for, 2–65
NU handler
See also Ethernet handlers
programming information for, 2–65

O

ODT
debug a handler, 1–111
restrictions with mapped monitors, 1–112

P

P1$EXT
RMON fixed offset 432
pointer to $P1EXT, 1–75
$P1EXT routine
borrows PAR1, 1–75
described, 1–75
restrictions, 1–76
PAR1
borrowed by $P1EXT to map user buffer, 1–75
PI handler
keyboard support, 2–76
function key mode, 2–77
normal mode, 2–76
programming information for, 2–76
sending command packet to GIDIS, 2–76
setting status from GIDIS, 2–76
video terminal support, 2–78
advanced video option, 2–78
device attributes for, 2–78
text cursor mode, 2–78
$PNAME table
referenced by bootstrap, 1–104
Primary driver

Primary driver (Cont.)
discussion, 1–83
entry routine, 1–84
set up by .DRBOT, 1–83
software bootstrap, 1–85
PS
See PSW
Pseudodevices
writing handlers for, 1–29
$PTBYT
pointer to $PUTBYT routine, 1–73
$PTBYT routine
illustrated, 1–22
$PTWRD
pointer to $PUTWRD routine, 1–74
$PTWRD routine
illustrated, 1–23
.PURGE programmed request
after .ENTER with magtapes, 2–43
after .LOOKUP with magtapes, 2–43
with magtapes, 2–43
$PUTBYT routine
described, 1–73
$PUTWRD routine
described, 1–74

Q

Q$BLKN, 1–8
Q$BUFF, 1–8
Q$COMP, 1–8
Q$CSW, 1–8
Q$FUNC, 1–8
Q$JNUM, 1–8
Q$LINK, 1–8
Q$MEM, 1–9
Q$PAR, 1–9
Q$UNIT, 1–8
Q$WCNT, 1–8
Q.BLKN, 1–8
Q.BUFF, 1–8
referenced by $MPMEM, 1–71
referenced by $MPPHY, 1–72
special directory device, for a, 1–68
updated by $GETBYT, 1–73
updated by $PUTBYT, 1–73, 1–74
Q.COMP, 1–8
Q.CSW, 1–8
Q.ELGH, 1–8
Q.FUNC, 1–8
Q.FUNC (Cont.)
check for .SPFUN request, 1–67
for a special directory device, 1–68
with extended device-unit handlers, 1–101
Q.JNUM, 1–8
Q.LINK, 1–7
Q.MEM, 1–8
referenced by $MPMEM, 1–71
Q.PAR, 1–8
referenced by $MPPHY, 1–72
used by $P1EXT, 1–76
Q.UNIT, 1–8
Q.WCNT, 1–8
with special directory devices, 1–69
.QELDF macro
called by .DRDEF, 1–5, 1–7
defines queue element offsets, 1–7
Queue element offsets
defined by .QELDF, 1–7
for mapped monitors, 1–7

R

.READC programmed request
with magtapes, 2–40
.READ programmed request
with magtapes, 2–40
.READW programmed request
with magtapes, 2–40
Registers available
error logger, 1–64
REMOVE keyboard command
device slots, makes available, 1–104
.RENAME programmed request
invalid with magtapes, 2–42
RGT.BS, 1–36
RGT.FG, 1–36
RGT.PD, 1–36
RGT.SB, 1–36
RGT.ST, 1–36
RGT.SZ, 1–36
RGT.TC, 1–36
RL01/RL02 disk
See also DL handler
RLSUMR routine, 2–87
$H2UB fixed offset with, 2–88
mirrors ALLUMR, 2–87
RMON
with abort processing, 1–43
action defined for, 1–45

RMON
with abort processing (Cont.)
order of action by, 1–46
RONLY$
bit in device status word, 1–11
defined by .DRDEF, 1–10
RQDX1 controller
bad-block replacement for, 2–10
RQDX2 controller
bad-block replacement for, 2–10
RQDX3 controller
bad-block replacement for, 2–10
RSTS/E
reading tapes from, 2–56
RSX–11D (IAS)
reading tapes from, 2–57
RSX–11M
reading tapes from, 2–56
RTEM$, 1–37
RX01 diskette
See DX handler
RX02 diskette
See DY handler
RX50 diskette
See DU or DZ handler

S

SDE.PA, 1–51
Second handler status word, 1–36
SET keyboard command
enabling options for, 1–49
enabling options with .DRSET, 1–51
how it works, 1–50
option table contents, 1–51
option table format, 1–50
SETOVR PSECT
with extended device-unit support, 1–98
Seven-track magtape, 2–58
SF.ARD
for DL, 2–3
for DM, 2–5
for DU, 2–8
for DU, described, 2–12
for DW, 2–25
for DX, DY, 2–26
for DZ, 2–28
SF.AWR
for DL, 2–3
for DM, 2–5
SF.AWR (Cont.)
for DU, 2–8
for DU, described, 2–12
for DW, 2–25
for DX, DY, 2–26
for DZ, 2–28
SF.BBR
for DL, 2–3
for DM, 2–5
SF.BRK
for communications handlers, 2–95
SF.BYP
for DU, 2–8
for DU, control block for, 2–18
for DU, described, 2–17
for MU, 2–62
SF.CLR
for communications handlers, 2–95
SF.DTR
for communications handlers, A–1
SF.MBS
with magtapes, 2–52
SF.MFS
with magtapes, 2–53
SF.MOR
with magtapes, 2–51
SF.MRD
for magtapes, 2–47
requires an non-file-structured .LOOKUP, 2–47
SF.MRE
with magtapes, 2–52
SF.MST
for magtapes, 2–50
SF.MTB
for MU, 2–60
SF.MTM
with magtapes, 2–54
SF.MWE
with magtapes, 2–52
SF.MWR
for magtapes, 2–48
requires an non-file-structured .LOOKUP, 2–48
SF.NAL
for Ethernet handlers, 2–66
SF.NMU
for Ethernet handlers, 2–69
SF.NPR
for Ethernet handlers, 2–67

SF.NRD
for Ethernet handlers, 2–73
SF.NWR
for Ethernet handlers, 2–71
SF.OBY
for DU, 2–8
SF.OFF
for communications handlers, 2–96
SF.PRD
for PI, 2–76
SF.PWR
for PI, 2–76
SF.R32
for DU, 2–8
for DU, described, 2–12
SF.S16
for DU, 2–8
SF.S32
for DU, 2–8
SF.SIZ
for DL, 2–3
for DM, 2–5
for DU, 2–8
for DU, described, 2–8
for DW, 2–25
for DZ, DY, 2–26
for LD, 2–29
SF.SRD
for communications handlers, 2–95
SF.STS
for communications handlers, 2–96
SF.TAB
for DU, 2–8
for DU, described, 2–14
for LD, 2–29
SF.TRD
with SF.TAB, 2–16
SF.TWR
with SF.TAB, 2–16
SF.USR
for magtapes, 2–44
requires non-file-structured .LOOKUP, 2–44
SF.W32
for DU, 2–8
for DU, CAUTION when using, 2–13
for DU, described, 2–12
SF.WDD
for DX, DY, 2–26
.SFDDF system definition macro, 2–2
.SFMDF system definition macro, 2–2
.SFNDF system definition macro, 2–2
.SFODF system definition macro, 2–2
.SFXDF system definition macro, 2–2
.SFxxx programmed requests
invalid with magtapes, 2–42
Software bad-block replacement
for DL handler, 2–3
for DM handler, 2–6
table illustrated, 2–4
Software bad-block replacemen table, 1–36
Software bootstrap, 1–85
Source listings, annotated
DL handler, A–23
DX handler, A–1
XL handler, A–56
Spacing error recovery, 2–36
Special function bypass
for DU, 2–17
for MU, 2–62
Special function codes
defining with .DRSPF, 1–13
Special function code support table, 1–34
Special function names
finding distributed in SYSTEM.MLB, 2–2
Special functions
See also .SPFUN programmed request
See also Special function codes
See also Special function names
exception reporting for magtapes, 2–48
finding names for distributed, 2–2
for DL, 2–3
for DM, 2–5
for DU, 2–8
for DW, 2–25
for DX, 2–26
for DY, 2–26
for DZ, 2–28
for LD, 2–29
for MU, 2–60
for NC, NQ, NU, 2–65
for PI, 2–76
for UB
See also Special functions, nonstandard
See also Special functions, standard
for XC, XL, 2–94
Special functions, nonstandard
defined, 2–84

Special functions, nonstandard (Cont.)
.DRBEG macro with, 2–86
obtaining temporary UMRs for, 2–86
Special functions, standard
defined, 2–84
.DRBEG macro with, 2–84
implicit UMR allocation for, 2–84
SPECL$
bit in device status word, 1–11
defined by .DRDEF, 1–10
SPFUN$
bit in device status word, 1–11
defined by .DRDEF, 1–10
.SPFUN programmed request
check Q.FUNC, 1–67
described, 1–65
support in device handlers, 1–67
variable-size device, for a, 1–68
with mapped monitors, 1–66
.SRESET programmed request
action by RMON at abort by, 1–44
STATU$
described, 2–9
uses .DRTAB, 2–9
SYSGEN conditionals
in a device handler, 1–7
SYSGEN options byte (H.GEN), 1–37
SYSTBL.MAC
edit to add a new handler, 1–106
SYSTEM.MLB
finding special function names in, 2–2
System device handlers
creating, 1–83
discussion, 1–83

T

$THKPT
with handler hooks support, 1–93
THOOKS
accessed from $THKPT, 1–93
Timeout
See Device timeout
Timer queue element
applied to device I/O timeout, 1–56
.TIMIO macro
argument range, 1–57
described, 1–55
TIMIT$, 1–37
TMSCP unit numbers and ports
translation table for, 2–61
Transferring data, 1–71
address translation for, 1–71
byte transfers, 1–72
without routine support, 1–75
word transfers, 1–74
Translation tables
for DU, 2–14
accessing, 2–16
bypassing, 2–17
for LD
accessing, 2–31
size of, 2–29
structure of, 2–30
for MU, 2–60
accessing, 2–62
bypassing, 2–62
size of, 2–60
structure of, 2–61

U

UB handler
See also UNIBUS DMA handlers
See also UNIBUS non-DMA handlers
if serialized I/O satisfaction needed, 2–81
installation requirements for, 2–80
permanent UMR allocation for distributed handlers, 2–80
programming information for, 2–80
UMR support for distributed handlers, 2–82
when permanent UMR list is fragmented, 2–81
UDA50 controller
bad-block replacement for, 2–11
UMR
device handler support structure for, 1–38
See also UB handler, 2–80
UMR allocation, permanent, 2–80
UMR allocation, temporary, 2–81
UMR support for distributed handlers, 2–82
UNIBUS DMA handlers
See also UB handler
ALLUMR routine with, 2–84
example of supporting UMRs for, 2–88
explicitly allocating permanent UMRs for, 2–84
releasing permanent UMRs, 2–87

UNIBUS DMA handlers (Cont.)
with various macros and requests, 2–83
UNIBUS Mapping Registers
See UMR and UB handler
UNIBUS non-DMA handlers
See also UB handler
.DRDEF macro with, 2–82
explicitly obtaining temporary UMRs for, 2–86
GETUMR routine with, 2–86
USR
magtape directory operations without, 2–44
$USRPH subroutine
$JBREL instead of, 1–82

V

Variable-size volumes, 1–67
VARSZ$
bit in device status word, 1–11
defined by .DRDEF, 1–10
Virtual memory handler
See VM handler
Virtual to physical address translation, 1–71
VM handler
description of base address for, 2–92
if doesn't install, 2–92
programming information for, 2–92

W

WONLY$
bit in device status word, 1–11
defined by .DRDEF, 1–10
.WRITC programmed request
with magtapes, 2–41
.WRITE programmed request
with magtapes, 2–41
.WRITW programmed request
with magtapes, 2–41

X

XALLOC subroutine, 1–81
XC handler
See also Communications handlers
programming information for, 2–94
XDEALC subroutine, 1–81
XL handler
See also Communications handlers
XL handler (Cont.)
annotated source listing of, A–56
programming information for, 2–94