


```

HH      HH      000000      MM      MM      000000      GGGGGGGG
HH      HH      000000      MM      MM      000000      GGGGGGGG
HH      HH      00      00      MMMM      MMMM      00      00      GG
HH      HH      00      00      MMMM      MMMM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG
HH      HH      000000      MM      MM      000000      GGGGGG
HH      HH      000000      MM      MM      000000      GGGGGG

```

```

....
....
....
....

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS

```



```
0000 82          .SBTTL  DECLARATIONS
0000 83          .PSECT  MONDATA,QUAD,NOEXE
0000 84          :
0000 85          : INCLUDE FILES:
0000 86          :
0000 87          :
0000 88          $CDBDEF      ; Define Class Descriptor Block
0000 89          $CDXDEF      ; Define CDB Extension
0000 90          $IDBDEF      ; define item descriptor block offsets
0000 91          $MBPDEF      ; Define Monitor Buffer Pointers
0000 92          $MONDEF      ; Monitor Recording File Definitions
0000 93          $SCBDEF      ; Define STATS Control Block
0000 94          $TM1DEF      ; Define temporary storage offsets
```

```

0000 0000 96 .SBTTL FILL HOMOG STATS - Fill STATS buffs for homogs
0000 0000 97 .PSECT $$MORCODE,ROWRT,EXE
0000 0000 98
0000 0000 99 :++
0000 0000 100 :
0000 0000 101 : FUNCTIONAL DESCRIPTION:
0000 0000 102 :
0000 0000 103 : FILL_HOMOG_STATS
0000 0000 104 :
0000 0000 105 : This routine fills all the STATS buffers for the
0000 0000 106 : class indicated by CDBPTR. The SCB (STATS Control
0000 0000 107 : Block) Table and the Element ID Table are also
0000 0000 108 : updated. These tables maintain information about
0000 0000 109 : the elements of this homogeneous class. An "element"
0000 0000 110 : is, for example, a particular disk in the DISK class.
0000 0000 111 : There is a STATS buffer for each item defined for the
0000 0000 112 : class. An "item" is, for example, operation count
0000 0000 113 : for the DISK class.
0000 0000 114 :
0000 0000 115 : INPUTS:
0000 0000 116 :
0000 0000 117 : 4(AP) - address of CURRENT collection buffer
0000 0000 118 :
0000 0000 119 : 8(AP) - address of PREVIOUS collection buffer
0000 0000 120 :
0000 0000 121 : IMPLICIT INPUTS:
0000 0000 122 :
0000 0000 123 : OUTPUTS:
0000 0000 124 :
0000 0000 125 : ALL STATS buffers for this homogeneous class filled.
0000 0000 126 :
0000 0000 127 : IMPLICIT OUTPUTS:
0000 0000 128 :
0000 0000 129 : CDB$$_ECOUNT and CDX$$_CUMELCT established for the current interval.
0000 0000 130 : Element ID Table and SCB (STATS Control Block) updated.
0000 0000 131 :
0000 0000 132 : ROUTINE VALUE:
0000 0000 133 :
0000 0000 134 : RO = $$$_NORMAL
0000 0000 135 :
0000 0000 136 : SIDE EFFECTS:
0000 0000 137 :
0000 0000 138 : none
0000 0000 139 :
0000 0000 140 : --
0000 0000 141 :

```

			0000	143			
		OFFC	0000	144	.ENTRY	FILL_HOMOG_STATS, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>	
			0002	145			
56	00000000	'EF	D0	0002	146	MOVL	CDBPTR,R6 ; Load CDB addr
	57	32 A6	D0	0009	147	MOVL	CDBSA_CD(R6),R7 ; Load CDX addr
	58	04 AC	D0	000D	148	MOVL	4(AP),R8 ; Load CURRENT coll buff addr
				0011	149	ALLOC	TMP\$K_SIZE,R0,R9 ; Allocate local temp storage
18	A6	0A A7	3C	001E	150	MOVZWL	CDX\$W_CUMELCT(R7), - ; Load element count for display
				0023	151		CDB\$E_COUNT(R6)
	58	0D	C0	0023	152	ADDL2	#MNR_CL\$K_HSIZE,R8 ; Point to CURR coll buff prologue
	69	68	D0	0026	153	MOVL	MNR_ROM\$E_LCT(R8), - ; Load current buffer data block count
				0029	154		TMP\$E_DBCT(R9)
		03	12	0029	155	BNEQ	\$S ; Br if have some
		008A	31	002B	156	BRW	UPDATE_SCB_FLAGS ; Else skip past ID Table update
				002E	157		\$S:
	58	08	C0	002E	158	ADDL2	#MNR_HOM\$K_PSIZE,R8 ; Point to first data block
04	A9	20 A6	3C	0031	159	MOVZWL	CDB\$W_BLKLEN(R6), - ; Get data block length
				0036	160		TMP\$E_DBLEN(R9)
08	A9	0A A7	3C	0036	161	MOVZWL	CDX\$W_CUMELCT(R7), - ; Load number of ID Table elements
				003B	162		TMP\$E_ELIDCT(R9)
	5A	09 A7	9A	003B	163	MOVZBL	CDX\$E_ELIDLEN(R7),R10 ; Get element ID length
				003F	164		


```

003F 166 :
003F 167 : Loop through all data blocks in the CURRENT collection buffer.
003F 168 : For each element (represented by a data block), try to find
003F 169 : a match in the element ID table. The ID table represents elements
003F 170 : which have been monitored for this request. On the first time
003F 171 : through this routine, the table will be empty. The element ID table
003F 172 : has several other associated tables, namely the STATS control
003F 173 : block (SCB) table, and all the transformation buffers (STATS, MIN,
003F 174 : MAX, SUM, PCSTATS, PCMIN, PCMAX and PCSUM). Each of these tables/
003F 175 : buffers has one element for each monitored element (i.e., disk for
003F 176 : the DISK class). The current number of elements in each of the tables
003F 177 : is represented by CDX$W_CUMELCT.
003F 178 :
003F 179 :
      OC A9 D4 003F 180          CLRL   TMP$L_DBIDX(R9)          ; Clear data block index
0042 181 10$:
5B   OC A7 D0 0042 182          MOVL   CDX$A_ELIDTABLE(R7),R11 ; Load Element ID Table addr
      10 A9 94 0046 183          CLRFB  TMP$B_FOUND(R9)          ; Clear 'element found' indicator
      54 D4 0049 184          CLRL   R4                      ; Clear element ID table index
55   08 A9 D0 004B 185          MOVL   TMP$L_ELIDCT(R9),R5 ; Load number of elements in ID table
      1E 13 004F 186          BEQL   40$                    ; Br if table is empty
      57 5A D0 0051 187          MOVL   R10,R7                 ; Borrow R7 to hold elt id length
03 4B A6 06 E1 0054 188          BBC    #CDB$V_DISKAC, - ; Branch if no allocation class in name
      00AF 30 0059 189          CDB$L_FLAGS(R6),20$ ;
005C 190          BSBW   SHORTEN_DISKNAM ; Shorten element name for DISK
6B   68 57 29 005C 191 20$:
      06 12 0060 192          CMPC3  R7,(R8),(R11) ; Match current element in table?
10  A9 01 90 0062 193          BNEQU 30$                    ; Br if not
      07 11 0066 194          MOVFB  #1,TMP$B_FOUND(R9) ; Yes -- indicate so
      5B 5A C0 0068 195 30$:
ED 54 55 F2 006B 196          BRB    40$                    ; ... and terminate loop
      55 F2 0068 197          ADDL2  R10,R11 ; Point to next element ID
      55 F2 006B 198          AOBLSS R5,R4,20$ ; Loop through element ID table
006F 199
006F 200 :
006F 201 : At this point the entire element ID table has been scanned for a
006F 202 : match to the current element in the CURRENT collection buffer.
006F 203 :
006F 204 :
006F 205 40$:
57   32 A6 D0 006F 206          MOVL   CDB$A_CDX(R6),R7 ; Re-load CDX addr
      06 10 A9 E8 0073 207          BLBS  TMP$B_FOUND(R9),50$ ; Branch if element found in table
54   0A A7 3C 0077 208          MOVZWL CDX$W_CUMELCT(R7),R4 ; Get next available element index
007B 209
007B 210 :
007B 211 : NOTE -- if R4 is greater than or equal to MAXELTS, issue warning msg
007B 212 : and simply branch to look at next coll buff data block.
007B 213 :
007B 214 :
      68 10 007B 215          BSBB   CHECK_TAB_SPACE ; Check if table space exhausted
007D 216          ; NOTE -- if so, MONITOR request
007D 217          ; ... is terminated
007D 218
007D 219 50$:
55   54 03 C5 007D 220          MULL3  #SCB$K_SIZE,R4,R5 ; Get SCB offset from index
55   10 B745 9E 0081 221          MOVAB  @CDX$A_SCBTABLE(R7)[R5],R5 ; Get SCB address
0086 222

```

HOMOG
V04-000

L 2

- MONITOR Homogeneous Class STATS Rtn 16-SEP-1984 02:05:50 VAX/VMS Macro V04-00 Page 7
FILL_HOMOG_STATS - Fill STATS buffs for 5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1 (6)

00	02	A5	00	E2	0086	223	BBSS	#SCBSV_CURRENT, -	; Set 'current' bit indicating this
					0088	224		SCBSB_FLAGS(R5),60\$; element in ID table was in CURR buff
					0088	225			
					0088	226			
05	10	A9		E8	008B	227	BLBS	TMP\$B_FOUND(R9),70\$; Branch if element found in table

MF
V0

```

008F 229 :
008F 230 : Element in CURRENT buffer was NOT found in the element ID table.
008F 231 : Add a new element to the table.
008F 232 :
008F 233 :
0068 30 008F 234 BSBW ADD_NEW_ELT ; Add elt to table
1B 11 0092 235 ; NOTE -- several registers altered
0092 236 BRB 80$ ; Go look at next coll buff data block
0094 237
0094 238 :
0094 239 : Element in CURRENT was found in the element ID table.
0094 240 :
0094 241 :
0094 242 70$:
0094 243
0094 244 MOVZWL SCB$W_DBIDX(R5),R11 ; Get data block index for prev buff
65 5B 65 3C 0097 245 MOVW TMP$L_DBIDX(R9),SCB$W_DBIDX(R5) ; Save curr index for next int
65 OC A9 B0
0F 02 A5 01 E1 009B 246 BBC #SCB$V_ACTIVE, - ; Done with this elt if not active
009B 247 SCB$B_FLAGS(R5),80$
00A0 248
00A0 249
00A0 250 :
00A0 251 : This element is active. Call routine to actually fill the STATS buffers,
00A0 252 : given the element ID table index and the addresses of this element's
00A0 253 : data blocks for both CURRENT and PREVIOUS collection buffers.
00A0 254 :
00A0 255 :
5B 04 A9 C4 00A0 256 MULL2 TMP$L_DBLEN(R9),R11 ; Get data block offset from index
52 08 AC D0 00A4 257 MOVL 8(AP),R2 ; Get ptr to PREVIOUS coll buff
5B 15 A24B 9E 00A8 258 MOVAB <MNR_CLS$K_HSIZE+MNR_HOM$K_PSIZE>(R2)[R11],R11 ; Compute PREVIOUS data block addr
00AD 259
00AD 260
6C 10 00AD 261 BSBW HOMOG_STATS ; Fill STATS buffs for all req'd items
00AF 262 ; NOTE -- this subrtn destroys
00AF 263 ; ... R0-R3 an. R5
00AF 264
00AF 265 80$:
00AF 266
5B 04 A9 C0 00AF 267 ADDL2 TMP$L_DBLEN(R9),R8 ; Point to next data block
8A OC A9 69 F2 00B3 268 AOBLS TMP$L_DBCT(R9), - ; Loop once for each elt in CURR buff
00B3 269 TMP$L_DBIDX(R9),10$
00B8 270
00B8 271
00B8 272

```



```

011B 350 :
011B 351 : HOMOG_STATS - Fill the STATS buffer for each requested data item
011B 352 : (e.g., disk operation count) for the current element
011B 353 : (e.g., DBA0).
011B 354 :
011B 355 : Register Inputs:
011B 356 :
011B 357 : R4 = element index of current element
011B 358 : R6 = CDB address
011B 359 : R7 = CDX address
011B 360 : R8 = address of data block for CURRENT coll buffer,
011B 361 : OR 0, if STATS buffers are to be cleared.
011B 362 : R10 = element ID length (unused if R8 = 0)
011B 363 : R11 = address of data block for PREVIOUS coll buffer
011B 364 : (unused if R8 = 0)
011B 365 :
011B 366 : Implicit Inputs:
011B 367 :
011B 368 : Volatile registers: R0, R1, R2, R3, R5
011B 369 :
011B 370 : Implicit outputs:
011B 371 :
011B 372 : ALL STATS buffers updated.
011B 373 :
011B 374 :
011B 375 : HOMOG_STATS:
011B 376 :
0600 8F B2 011B 377 : PUSHR #*M<R9,R10> : Save regs
011F 378 : TSTL R8 : Data block ptr present?
14 12 0121 379 : BNEQ 20$ : Br if yes
0123 380 :
0123 381 : Special case: clear STATS buffers for this element
0123 382 :
0123 383 :
50 06 A7 9A 0123 384 : MOVZBL CDX$B_IDISCT(R7),R0 : Get number of STATS buffers
51 2E A6 D0 0127 385 : MOVL CDB$A_BUFFERS(R6),R1 : Get addr of first MBP ptr
012B 386 :
10$: 012B 387 : MOVL (R1)+,R2 : Get MBP pointer
08 B244 D4 012E 388 : CLRL @MBP$A_STATS(R2)[R4] : Clear STATS for this item & element
F6 50 F5 0132 389 : SOBGTR R0,10$ : Loop back for next STATS buffer
24 11 0135 390 : BRB 40$ : Go to common return

```


HOMOG
Symbol table

F 3
- MONITOR Homogeneous Class STATS Rtn

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 14
(12)

IDBSM_PCNT = 00000001
IDBSS_FIL_L_R = 00000007
IDBSS_FLAGS = 00000001
IDBSS_IDB = 00000011
IDBSV_FILLER = 00000001
IDBSV_PCNT = 00000000
IDBSW_ISIZE = 00000008
IDBSW_TYPE = 0000000A
MAXELTS *****
MAX_STAT = 00000004
MBP = 00000000
MBPSA_ADDR = 00000018
MBPSA_B1ST = 00000004
MBPSA_BA = 00000000
MBPSA_BUFF1ST = 00000004
MBPSA_BUFFERA = 00000000
MBPSA_BUFFERB = 00000004
MBPSA_DATA = 00000008
MBPSA_DIFF = 0000000C
MBPSA_MAX = 00000010
MBPSA_MIN = 0000000C
MBPSA_ORDER = 00000010
MBPSA_PCMAX = 00000020
MBPSA_PCMIN = 0000001C
MBPSA_PCSTATS = 00000018
MBPSA_PCSUM = 00000024
MBPSA_PID = 00000014
MBPSA_PR_FAOSTK = 00000008
MBPSA_STATS = 00000008
MBPSA_SUM = 00000014
MBPSK_SIZE = 00000028
MBPSS_MBP = 00000028
MBPSS_MBP2 = 0000001C
MBPSS_MBP3 = 0000000C
MBP2 = 00000000
MBP3 = 00000000
MIN_STAT = 00000003
MNR\$ TABLEFULL *****
MNR_CLSSB_TYPE = 00000000
MNR_CLSSK_HSIZE = 0000000D
MNR_CLSSQ_STAMP = 00000003
MNR_CLSSS_CLASS_HDR = 0000000D
MNR_CLSSS_FILLER = 0000000F
MNR_CLSSS_FLAGS = 00000002
MNR_CLSSS_STAMP = 00000008
MNR_CLSSV_CONT = 00000000
MNR_CLSSV_FILLER = 00000001
MNR_CLSSW_FLAGS = 00000001
MNR_CLSSW_RESERVED = 0000000B
MNR_HDR\$B_TYPE = 00000000
MNR_HDR\$K_CLASSBITS = 00000073
MNR_HDR\$K_MAXCOMLEN = 0000003C
MNR_HDR\$K_REVLEVELS = 00000083
MNR_HDR\$K_SIZE = 00000103
MNR_HDR\$L_FLAGS = 00000001
MNR_HDR\$L_INTERVAL = 00000015

X 02

X 02

MNR_HDR\$L_RECCT = 00000029
MNR_HDR\$O_CLASSBITS = 00000073
MNR_HDR\$O_REVOCLSBITS = 00000019
MNR_HDR\$Q_BEGINNING = 00000005
MNR_HDR\$Q_ENDING = 0000000D
MNR_HDR\$S_BEGINNING = 00000008
MNR_HDR\$S_CLASSBITS = 00000010
MNR_HDR\$S_COMMENT = 0000003C
MNR_HDR\$S_ENDING = 00000008
MNR_HDR\$S_FILE_HDR = 00000103
MNR_HDR\$S_FILLER = 00000020
MNR_HDR\$S_FLAGS = 00000004
MNR_HDR\$S_LEVEL = 00000008
MNR_HDR\$S_REVOCLSBITS = 0000001C
MNR_HDR\$S_REVLEVELS = 00000080
MNR_HDR\$S_TYPE = 00000008
MNR_HDR\$T_COMMENT = 00000035
MNR_HDR\$T_LEVEL = 0000002D
MNR_HDR\$T_REVLEVELS = 00000083
MNR_HDR\$V_FILLER = 00000000
MNR_HDR\$W_COMLEN = 00000071
MNR_HOM\$K_PSIZE = 00000008
MNR_HOM\$L_ELCT = 00000000
MNR_HOM\$L_RESERVED = 00000004
MNR_HOM\$S_HOM_CLASS_PRE = 00000008
MNR_PRO\$B_PRI = 0000000A
MNR_PRO\$K_DSIZE = 0000003B
MNR_PRO\$K_FSIZE = 00000040
MNR_PRO\$K_PSIZE = 00000008
MNR_PRO\$K_REVODSIZE = 00000033
MNR_PRO\$K_REV1DSIZE = 0000003B
MNR_PRO\$L_BIOCNT = 0000002F
MNR_PRO\$L_CPUTIM = 0000002B
MNR_PRO\$L_DIOCNT = 00000023
MNR_PRO\$L_EFWM = 00000037
MNR_PRO\$L_EPID = 00000033
MNR_PRO\$L_IPID = 00000000
MNR_PRO\$L_PAGEFLTS = 00000027
MNR_PRO\$L_PCTINT = 00000004
MNR_PRO\$L_PCTREC = 00000000
MNR_PRO\$L_STS = 0000001F
MNR_PRO\$L_UIC = 00000004
MNR_PRO\$O_LNAME = 0000000B
MNR_PRO\$S_LNAME = 00000010
MNR_PRO\$S_PROCESS_CLASS = 0000003B
MNR_PRO\$S_PRO_CLASS_PRE = 00000008
MNR_PRO\$W_GPGCNT = 0000001B
MNR_PRO\$W_PPGCNT = 0000001D
MNR_PRO\$W_STATE = 00000008
MNR_SYISB_MPCPUS = 0000000D
MNR_SYISB_TYPE = 00000000
MNR_SYISK_BALSETMEM = 0000001E
MNR_SYISK_CPUTYPE = 00000026
MNR_SYISK_MPWHILIM = 00000022
MNR_SYISK_NODENAME = 0000000E
MNR_SYISK_SIZE = 0000002A
MNR_SYISL_BALSETMEM = 0000001E

MF
V0
1
1

HOMOG
Symbol table

G 3
- MONITOR Homogeneous Class STATS Rtn

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 15
(12)

MNR_SYISL_CPUTYPE = 00000026
MNR_SYISL_MPWHILIM = 00000022
MNR_SYISQ_BOOTTIME = 00000003
MNR_SYISS_BOOTTIME = 00000008
MNR_SYISS_FILLER = 0000000E
MNR_SYISS_FLAGS = 00000002
MNR_SYISS_NODENAME = 00000010
MNR_SYISS_SYS_INFO = 0000002A
MNR_SYISS_TYPE = 00000008
MNR_SYIST_NODENAME = 0000000E
MNR_SYISV_CLUSMEM = 00000000
MNR_SYISV_FILLER = 00000002
MNR_SYISV_RESERVED1 = 00000001
MNR_SYISW_FLAGS = 00000001
MNR_SYISW_MAXPRCCT = 0000000B
PERFTABLE = *****
PROCDISPS = 00000005
PROCESS_CLASS = 00000000
PRO_CLASS_PRE = 00000000
QUALSA_ALC = 00000064
QUALSA_AVE = 00000074
QUALSA_BEG = 00000004
QUALSA_BY_NODE = 00000054
QUALSA_CLASS = 0000005C
QUALSA_COMM = 0000004C
QUALSA_CPU = 000000AC
QUALSA_CUR = 0000006C
QUALSA_DISP = 00000034
QUALSA_END = 0000000C
QUALSA_FLUSH = 0000001C
QUALSA_INP = 0000002C
QUALSA_INT = 00000014
QUALSA_ITEM = 000000BC
QUALSA_MAX = 00000084
QUALSA_MIN = 0000007C
QUALSA_PCEN = 000000B4
QUALSA_REC = 0000003C
QUALSA_SUMM = 00000044
QUALSA_TOPB = 0000009C
QUALSA_TOPC = 0000008C
QUALSA_TOPD = 00000094
QUALSA_TOPF = 000000A4
QUALSA_VIEW = 00000024
QUALSL_ALL = 00000060
QUALSL_AVE = 00000070
QUALSL_BEG = 00000000
QUALSL_BY_NODE = 00000050
QUALSL_CLASS = 00000058
QUALSL_COMM = 00000048
QUALSL_CPU = 000000A8
QUALSL_CUR = 00000068
QUALSL_DISP = 00000030
QUALSL_END = 00000008
QUALSL_FLUSH = 00000018
QUALSL_INP = 00000028
QUALSL_INT = 00000010
QUALSL_ITEM = 000000B8

X 02

QUALSL_MAX = 00000080
QUALSL_MIN = 00000078
QUALSL_PCEN = 00000060
QUALSL_REC = 00000038
QUALSL_SUMM = 00000040
QUALSL_TOPB = 00000098
QUALSL_TOPC = 00000088
QUALSL_TOPD = 00000090
QUALSL_TOPF = 000000A0
QUALSL_VIEW = 00000020
QUALSS_QUALIFIER_DESC = 000000C0
QUALIFIER_DESC = 00000000
REG_PROC = 00000000
SCBSB_FLAGS = 00000002
SCBSK_SIZE = 00000003
SCBSS_FILLER = 00000006
SCBSS_FLAGS = 00000001
SCBSS_STATS_BLOCK = 00000003
SCBSV_ACTIVE = 00000001
SCBSV_CURRENT = 00000000
SCBSV_FILLER = 00000002
SCBSW_DBIDX = 00000000
SHORTEN_DISKNAM = 000010B R 02
SS\$NORMAL = ***** X 02
STATS = 00000005
STATS_BLOCK = 00000000
STORE_STATS = 00000160 R 02
SYS_INFO = 00000000
TEMP_1_BLOCK = 00000000
TMP\$B_FOUND = 00000010
TMP\$K_SIZE = 00000011
TMP\$L_DBCT = 00000000
TMP\$L_DBIDX = 0000000C
TMP\$L_DBLEN = 00000004
TMP\$L_ELIDCT = 00000008
TMP\$S_TEMP_1_BLOCK = 00000011
TOPB_PROC = 00000003
TOPC_PROC = 00000001
TOPD_PROC = 00000002
TOPF_PROC = 00000004
UPDATE_SCB_FLAGS = 000000B8 R 02

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
MONDATA	00000000 (0.)	01 (1.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
\$\$MONCODE	00000195 (405.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU time	Elapsed Time
Initialization	29	00:00:00.09	00:00:01.43
Command processing	111	00:00:00.75	00:00:06.13
Pass 1	161	00:00:02.44	00:00:09.57
Symbol table sort	0	00:00:00.48	00:00:00.66
Pass 2	88	00:00:01.08	00:00:03.43
Symbol table output	39	00:00:00.34	00:00:01.12
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	432	00:00:05.22	00:00:22.36

The working set limit was 1200 pages.
15442 bytes (31 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 326 non-local and 22 local symbols.
442 source lines were read in Pass 1, producing 17 object records in Pass 2.
16 pages of virtual memory were used to define 8 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[MONITOR.OBJ]MONLIB.MLB;1	7
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	7

327 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:HOMOG/OBJ=OBJ\$:HOMOG MSRC\$:HOMOG/UPDATE=(ENH\$:HOMOG)+EXE/ML\$/LIB+LIB\$:MONLIB/LIB

0240 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 144 terminal windows arranged in 12 rows and 12 columns. Each window shows a different screen from a VAX/VMS system, likely a menu-driven interface for system management or data processing. The screens contain various types of information, including lists of files, system status, and data tables. Four specific windows are highlighted with larger, bold text labels:

- HOMOG LIS**: Located in the second row, first column.
- MONITOR LIS**: Located in the second row, eighth column.
- MFSUMM LIS**: Located in the fifth row, second column.
- MONDAT LIS**: Located in the fifth row, fifth column.

The other windows in the grid show similar but less legible content, including what appears to be a directory listing, system logs, and data reports. The overall appearance is that of a multi-user terminal session on a mainframe or minicomputer system.