

```
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFFFFFFFFFFFFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
FFF
```

```
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
11111111
11111111
11111111
```

```
111
111
111
111111
111111
111111
111
111
111
111
111
111
111
111
111
111
111
111
111
111
11111111
11111111
11111111
```

```
AAAAAAAAAA
AAAAAAAAAA
AAAAAAAAAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
AAA        AAA
```



```

CCCCCCCC  HH      HH  KK      KK  SSSSSSSS  UU      UU  MM      MM
CCCCCCCC  HH      HH  KK      KK  SSSSSSSS  UU      UU  MM      MM
CC        HH      HH  KK      KK  SS        UU      UU  MMMM   MMMM
CC        HH      HH  KK      KK  SS        UU      UU  MMMM   MMMM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CC        HHHHHHHHHH KKKKKK  SSSSSS  UU      UU  MM     MM  MM
CC        HHHHHHHHHH KKKKKK  SSSSSS  UU      UU  MM     MM  MM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CC        HH      HH  KK      KK  SS        UU      UU  MM     MM  MM
CCCCCCCC  HH      HH  KK      KK  SSSSSSSS  UUUUUUUUUU MM     MM  MM
CCCCCCCC  HH      HH  KK      KK  SSSSSSSS  UUUUUUUUUU MM     MM  MM

```

```

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

```

```

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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```



```

1 0001 0 MODULE CHKSUM (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1
9 0009 1
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 **
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine computes and checks a file header checksum.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1
41 0041 1 STARLET operating system, including privileged system services
42 0042 1 and internal exec routines.
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1
47 0047 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 22-Nov-1977 22:25
48 0048 1
49 0049 1 MODIFIED BY:
50 0050 1
51 0051 1 A0100 ACG0001 Andrew C. Goldstein, 10-Oct-1978 20:01
52 0052 1 Previous revision history moved to F11A.REV
53 0053 1
54 0054 1 **
55 0055 1
56 0056 1
57 0057 1 LIBRARY 'SYS$LIBRARY:LIB.L32';
    
```

CHKSUM
V04-000

; 58

0058 1 REQUIRE 'SRCS:FCPDEF.B32';

D 9
16-Sep-1984 00:50:35
14-Sep-1984 12:29:21

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]CHKSUM.B32;1 Page (1) 2

CLF
V04

.....

;


```

60 0373 1 GLOBAL ROUTINE CHECKSUM (HEADER) =
61 0374 1
62 0375 1 !++
63 0376 1
64 0377 1 FUNCTIONAL DESCRIPTION:
65 0378 1
66 0379 1 This routine computes, checks, and stores the checksum of a file header.
67 0380 1
68 0381 1 CALLING SEQUENCE:
69 0382 1 CHECKSUM (ARG1)
70 0383 1
71 0384 1 INPUT PARAMETERS:
72 0385 1 ARG1: address of file header buffer
73 0386 1
74 0387 1 IMPLICIT INPUTS:
75 0388 1 NONE
76 0389 1
77 0390 1 OUTPUT PARAMETERS:
78 0391 1 NONE
79 0392 1
80 0393 1 IMPLICIT OUTPUTS:
81 0394 1 NONE
82 0395 1
83 0396 1 ROUTINE VALUE:
84 0397 1 1 if checksum was correct
85 0398 1 0 if checksum was wrong
86 0399 1
87 0400 1 SIDE EFFECTS:
88 0401 1 Correct checksum stored in header
89 0402 1
90 0403 1 !--
91 0404 1
92 0405 2 BEGIN
93 0406 2
94 0407 2 MAP
95 0408 2 HEADER : REF VECTOR [,WORD];
96 0409 2
97 0410 2 LOCAL
98 0411 2 P : REF VECTOR [,WORD], ! pointer to scan header
99 0412 2 SUM : WORD, ! checksum value
100 0413 2 MATCH; ! success flag
101 0414 2
102 0415 2
103 0416 2 ! We simply sum the first 255 words of the file header. Then we compare the
104 0417 2 ! checksum in the header with the one computed, and store the new one.
105 0418 2 !
106 0419 2
107 0420 2 SUM = 0;
108 0421 2 P = .HEADER;
109 0422 2 DECR I FROM 255 TO 1
110 0423 2 DO
111 0424 2 BEGIN
112 0425 2 SUM = .SUM + .P[0];
113 0426 2 P = .P + 2;
114 0427 2 END;
115 0428 2
116 0429 2 MATCH = .SUM EQL .P[0];

```

CHKSUM
V04-000

F 9
16-Sep-1984 00:50:35
14-Sep-1984 12:29:21

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]CHKSUM.B32;1 Page 4
(2)

```
: 117      0430 2 P[0] = .SUM;
: 118      0431 2
: 119      0432 2 RETURN .MATCH;
: 120      0433 2
: 121      0434 1 END;
```

! end of routine CHECKSUM

.TITLE CHKSUM
.IDENT \V04-000\

.PSECT \$CODE\$,NOWRT,2

		0004 00000	.ENTRY CHECKSUM, Save R2	: 0373
		52 B4 00002	CLRW SUM	: 0420
51	04	AC D0 00004	MOVL HEADER, P	: 0421
50	FF	8F 9A 00008	MOVZBL #255, I	: 0422
52		81 A0 0000C 1\$:	ADDW2 (P)+, SUM	: 0425
FA		50 F5 0000F	SOBGR I, 1\$: 0422
		50 D4 00012	CLRL R0	: 0429
61		52 B1 00014	CMPW SUM, (P)	
		02 12 00017	BNEQ 2\$	
		50 D6 00019	INCL R0	
61		52 B0 0001B 2\$:	MOVW SUM, (P)	: 0430
		04 0001E	RET	: 0434

; Routine Size: 31 bytes, Routine Base: \$CODE\$ + 0000

```
: 122      0435 1
: 123      0436 1 END
: 124      0437 0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	31	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	1 0	1000	00:01.9

COMMAND QUALIFIERS

```
:  
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CHKSUM/OBJ=OBJ$:CHKSUM MSRC$:CHKSUM/UPDATE=(ENH$:CHKSUM)  
: Size: 31 code + 0 data bytes  
: Run Time: 00:05.1  
: Elapsed Time: 00:19.2  
: Lines/CPU Min: 5161  
: Lexemes/CPU-Min: 12224  
: Memory Used: 62 pages  
: Compilation Complete
```



0164 AH-BT13A-SE VAX/VMS V4.0
This document contains a grid of 100 small, faint images, likely representing individual data pages or reports. The images are arranged in a 10x10 grid. Some of the visible text within the images includes:

- CHKSUM LIS
- ACPCNTR LIS
- CHKPRO LIS
- FCPOEF B32
- DEACCS LIS
- BADSCR LIS
- CLEUP LIS
- CPYAM LIS
- CHKHOR LIS
- COMMON LIS
- CREHOR LIS
- CREWIN LIS
- ACCESS LIS
- ALLOB LIS
- CHKDMO LIS
- CREATE LIS
- CREFCB LIS
- DELETE LIS