


```

1 0001 0 MODULE PATCON (
2 L 0002 0 %IF %VARIANT EQL 1
3 0003 0 %THEN
4 0004 0 ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE, NONEXTERNAL = LONG_RELATIVE),
5 0005 0 %FI
6 0006 0 IDENT = 'V04-000') =
7 0007 1 BEGIN
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 FACILITY: PATCH
33 0033 1
34 0034 1 **
35 0035 1 FUNCTIONAL DESCRIPTION:
36 0036 1
37 0037 1 CONVERSION ROUTINES.
38 0038 1
39 0039 1 Version: V02-010
40 0040 1
41 0041 1 History:
42 0042 1 Author:
43 0043 1 Carol Peters, 18 May 1976: Version 01
44 0044 1
45 0045 1 Modified by:
46 0046 1
47 0047 1 V02-010 PCG0001 Peter George 02-FEB-1981
48 0048 1 Add require statement for LIB$PATDEF.REQ
49 0049 1
50 0050 1 01.09 CNH0013 Chris Hume 27-Aug-1979 13:30
51 0051 1 Added double byte OPcode support. Changed use of PAT$CONV_R_50
52 0052 1 to the RTL routine R50ASC. Removed the former from this module.
53 0053 1
54 0054 1 MODIFICATIONS:
55 0055 1
56 0056 1 NO DATE PROGRAMMER PURPOSE
57 0057 1 -- ---- -

```

58	0058	1	:				
59	0059	1	:	00	19-OCT-77	K.D. MORSE	
60	0060	1	:	01	4-JAN-78	K.D. MORSE	
61	0061	1	:	02	21-FEB-78	K.D. MORSE	
62	0062	1	:	03	24-MAR-78	K.D. MORSE	
63	0063	1	:				
64	0064	1	:				
65	0065	1	:	04	04-APR-78	K.D. MORSE	
66	0066	1	:	05	25-APR-78	K.D. MORSE	
67	0067	1	:	06	18-MAY-78	K.D. MORSE	
68	0068	1	:	07	13-JUN-78	K.D. MORSE	
69	0069	1	:	08	19-JUN-78	K.D. MORSE	
70	0070	1	:				
71	0071	1	:	--			

CONVERT VERSION 7 FOR PATCH.
NO CHANGES FOR VERS 8.
USE EMUL FOR OVERFLOW CHECK.
REPLACE SELECT WITH IF...THEN
AS IN DEBUG OVERFLOW CHECK AS
THIS SAVES BYTES. (9)
NO CHANGES FOR 10.
CONVERT TO NATIVE COMPILER.
NO CHANGES FC# VERS 11.
ADD FAO COUNTS TO SIGNALS.
NO CHANGES FOR VERS 12-13.

```

: 73      0072 1 FORWARD ROUTINE
: 74      0073 1          PAT$RADX_CONVRT : NOVALUE;          ! Converts a string to a value
: 75      0074 1
: 76      0075 1 LIBRARY 'SYSSLIBRARY:STARLET.L32';
: 77      0076 1 REQUIRE 'SRC$:VXSMAC.REQ';
: 78      0141 1 REQUIRE 'SRC$:PATPCT.REQ';
: 79      0181 1 REQUIRE 'SRC$:PATGEN.REQ';
: 80      0403 1 REQUIRE 'LIB$:PATDEF.REQ';          ! Defines literals
: 81      0457 1 REQUIRE 'LIB$:PATMSG.REQ';
: 82      0631 1
: 83      0632 1 EXTERNAL
: 84      0633 1          PAT$GB_MOD_PTR: REF VECTOR [, BYTE];    ! Pointer to current modes

```

```

86 0634 1 GLOBAL ROUTINE PAT$RADX_CONVRT (STRING_ADDR, VALUE_ADDR) : NOVALUE =
87 0635 1
88 0636 1 !++
89 0637 1 FUNCTIONAL DESCRIPTION:
90 0638 1
91 0639 1     Converts an ascii string to a longword value in the current radix.
92 0640 1
93 0641 1 CALLING SEQUENCE:
94 0642 1
95 0643 1     PAT$RADX_CONVRT ( )
96 0644 1
97 0645 1 INPUTS:
98 0646 1
99 0647 1     STRING_ADDR   - Address of ascii string
100 0648 1     VALUE_ADDR    - Address in which to put converted value
101 0649 1
102 0650 1 IMPLICIT INPUTS:
103 0651 1
104 0652 1     Contents of PAT$gb_mod_ptr [mode_radix], which is the current
105 0653 1     radix.
106 0654 1
107 0655 1 OUTPUTS:
108 0656 1
109 0657 1     none
110 0658 1
111 0659 1 IMPLICIT OUTPUTS:
112 0660 1
113 0661 1     A signal and unwind occurs if the conversion fails.
114 0662 1     The converted value is placed in the address passed as the
115 0663 1     second argument.
116 0664 1
117 0665 1 ROUTINE value:
118 0666 1
119 0667 1     novalue
120 0668 1
121 0669 1 SIDE EFFECTS:
122 0670 1
123 0671 1     none
124 0672 1
125 0673 1 --
126 0674 1
127 0675 2 BEGIN
128 0676 2
129 0677 2 BUILTIN
130 0678 2     EMUL;                                ! Longword mul and add to get quadword
131 0679 2
132 0680 2 MAP
133 0681 2     STRING_ADDR: REF VECTOR [, BYTE],
134 0682 2     VALUE_ADDR: REF VECTOR;
135 0683 2
136 0684 2 LOCAL
137 0685 2     GIVE_MESSAGE,                        ! Error flag
138 0686 2     value : VECTOR[2, LONG],            ! Converted value
139 0687 2     NEGATE,                               ! Negative number flag
140 0688 2     CHAR;                                ! Character-holding variable
141 0689 2
142 0690 2 VALUE[0] = 0;

```

```

143 0691 2 VALUE[1] = 0;
144 0692 2 GIVE_MESSAGE = FALSE;
145 0693 2 NEGATE = FALSE;
146 0694 2 INCR N FROM 0 TO (NUM_MAX_LENGTH - 1) DO
147 0695 2 BEGIN
148 0696 3 IF (.CHAR = .STRING_ADDR [.N]) EQL 0 THEN EXITLOOP;
149 0697 4 IF (.N EQL 0)
150 0698 3 THEN
151 0699 4 BEGIN
152 0700 5 IF (.CHAR EQL '-')
153 0701 4 THEN
154 0702 5 BEGIN
155 0703 5 NEGATE = TRUE;
156 0704 5 CHAR = '0'
157 0705 5 END
158 0706 4 ELSE
159 0707 5 IF (.CHAR EQL '+')
160 0708 4 THEN CHAR = '0';
161 0709 3 END;
162 0710 4 IF ((.CHAR GEQ '0') AND (.CHAR LEQ '9'))
163 0711 3 THEN
164 0712 3 CHAR = .CHAR - '0'
165 0713 3 ELSE
166 0714 4 IF ((.CHAR GEQ 'A') AND (.CHAR LEQ 'F'))
167 0715 3 THEN
168 0716 3 CHAR = .CHAR - 'A' + 10
169 0717 3 ELSE
170 0718 3 CHAR = 256;
171 0719 3 IF .CHAR GEQ .PAT$GB_MOD_PTR [MODE_RADIX]
172 0720 2 THEN
173 0721 2 GIVE_MESSAGE = PAT$_INVNUMBER
174 0722 2 ELSE
175 0723 2 EMUL(VALUE[0], %REF(.PAT$GB_MOD_PTR[MODE_RADIX]), CHAR, VALUE);
176 0724 2 IF .VALUE[1] NEQ 0
177 0725 2 THEN
178 0726 2 GIVE_MESSAGE = PAT$_NUMTRUNC; ! Numeric overflow
179 0727 2 END;
180 0728 2 IF (.GIVE_MESSAGE NEQ 0)
181 0729 2 THEN
182 0730 2 SIGNAL(.GIVE_MESSAGE);
183 0731 2 IF .NEGATE
184 0732 2 THEN
185 0733 2 VALUE[0] = - .VALUE[0];
186 0734 2 VALUE_ADDR [0] = .VALUE[0];
187 0735 2
188 0736 1 END;

```

```

.TITLE PATCON
.IDENT \V04-000\
.EXTRN PAT$GB_MOD_PTR
.PSECT _PAT$CODE,NOWRT,2
.ENTRY PAT$RADX_CONVRT, Save R2,R3,R4
SUBL2 #4, SP

```

```

SE 001C 0000
04 C2 0002

```

```

: 0634
:
```

		7E	D4	00005	CLRL	VALUE	: 0690
	04	AE	D4	00007	CLRL	VALUE+4	: 0691
		53	7C	0000A	CLRQ	GIVE_MESSAGE	: 0692
		52	D4	0000C	CLRL	N	: 0719
	50	04 BC	9A	0000E 1\$:	MOVZBL	@STRING_ADDR[N], CHAR	: 0696
		71	13	00013	BEQL	11\$	
		52	D5	00015	TSTL	N	: 0697
		12	12	00017	BNEQ	4\$	
	2D	50	D1	00019	CMPL	CHAR, #45	: 0700
		05	12	0001C	BNEQ	2\$	
	54	01	D0	0001E	MOVL	#1, NEGATE	: 0703
		05	11	00021	BRB	3\$: 0704
	2B	50	D1	00023 2\$:	CMPL	CHAR, #43	: 0707
		03	12	00026	BNEQ	4\$	
	50	30	D0	00028 3\$:	MOVL	#48, CHAR	: 0708
	30	50	D1	0002B 4\$:	CMPL	CHAR, #48	: 0710
		0A	19	0002E	BLSS	5\$	
	39	50	D1	00030	CMPL	CHAR, #57	
		05	14	00033	BGTR	5\$	
	50	30	C2	00035	SUBL2	#48, CHAR	: 0712
		1C	11	00038	BRB	7\$	
	00000041	8F	50	D1 0003A 5\$:	CMPL	CHAR, #65	: 0714
		0E	19	00041	BLSS	6\$	
	00000046	8F	50	D1 00043	CMPL	CHAR, #70	
		05	14	0004A	BGTR	6\$	
	50	37	C2	0004C	SUBL2	#55, CHAR	: 0716
		05	11	0004F	BRB	7\$	
	50	0100	8F	3C 00051 6\$:	MOVZWL	#256, CHAR	: 0718
	50 0000000G	FF	08	00 ED 00056 7\$:	CMPZV	#0, #8, @PAT\$GB_MOD_PTR, CHAR	: 0719
			09	14 0005F	BGTR	8\$	
	53 006D80EA	8F	D0	00061	MOVL	#7176426, GIVE_MESSAGE	: 0721
		0C	11	00068	BRB	9\$	
	51 0000000G	FF	9A	0006A 8\$:	MOVZBL	@PAT\$GB MOD_PTR, R1	: 0724
	6E	50	51	6E 7A 00071	EMUL	VALUE, R1, CHAR, VALUE	
			04	AE D5 00076 9\$:	TSTL	VALUE+4	: 0725
			07	13 00079	BEQL	10\$	
	53 006D8023	8F	D0	0007B	MOVL	#7176227, GIVE_MESSAGE	: 0727
		88	52	13 F3 00082 10\$:	AOBLEQ	#19, N, 1\$: 0694
			53	D5 00086 11\$:	TSTL	GIVE_MESSAGE	: 0729
			09	13 00088	BEQL	12\$	
			53	DD 0008A	PUSHL	GIVE_MESSAGE	: 0731
	0000000G	00	01	FB 0008C	CALLS	#1, [IB\$SIGNAL	
		03	54	E9 00093 12\$:	BLBC	NEGATE, 13\$: 0732
		6E	6E	CE 00096	MNEGL	VALUE, VALUE	: 0734
	08	BC	6E	D0 00099 13\$:	MOVL	VALUE, @VALUE_ADDR	: 0735
			04	0009D	RET		: 0736

: Routine Size: 158 bytes, Routine Base: _PAT\$CODE + 0000

: 190 0737 1 END
: 191 0738 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
_PAT\$CODE	158	NOVEC,NOWRT, RD, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	4 0	581	00:01.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/VARIANT:1/LIS=LISS:PATCON/OBJ=OBJ\$:PATCON MSRC\$:PATCON/UPDATE=(ENH\$:PATCON)

: Size: 158 code + 0 data bytes
: Run Time: 00:11.5
: Elapsed Time: 00:49.4
: Lines/CPU Min: 3857
: Loxemes/CPU-Min: 55426
: Memory Used: 107 pages
: Compilation Complete

0300 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY