

DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUU	UUU	GGGGGGGGGGGG
DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUU	UUU	GGGGGGGGGGGG
DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUU	UUU	GGGGGGGGGGGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDD	DDD	EEE	UUU	UUU	GGG
DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	GGGGGGGGGG
DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	GGGGGGGGGG
DDDDDDDDDDDD	EEEEEEEEEEEEEE	BBBBBBBBBBBBBB	UUUUUUUUUUUUUU	UUUUUUUUUUUUUU	GGGGGGGGGG

```

SSSSSSSS TTTTTTTTTT RRRRRRRR UU UU CCCCCCCC DDDDDDDD EEEEEEEEE EEEEEEEEE FFFFFFFF
SSSSSSSS TTTTTTTTTT RRRRRRRR UU UU CCCCCCCC DDDDDDDD EEEEEEEEE FFFFFFFF
SS TT RR RR UU UU CC DD DD EE EE FF
SS TT RR RR UU UU CC DD DD EE EE FF
SS TT RR RR UU UU CC DD DD EE EE FF
SSSSSS TT RRRRRRR UU UU CC DD DD EEEEEEE FFFFFFF
SSSSSS TT RRRRRRR UU UU CC DD DD EEEEEEE FFFFFFF
SS TT RR RR UU UU CC DD DD EE EE FF
SS TT RR RR UU UU CC DD DD EE EE FF
SS TT RR RR UU UU CC DD DD EE EE FF
SSSSSS TT RRRRRRR UU UU CCCCCC DDDDDDDD EEEEEEEEE FF
SSSSSS TT RRRRRRR UU UU CCCCCC DDDDDDDD EEEEEEEEE FF

```

```

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

```

```

RRRRRRR EEEEEEEEE QQQQQQ
RRRRRRR EEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RRRRRRR EEEEEEE QQ QQ
RRRRRRR EEEEEEE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EEEEEEEEE QQQQ QQ
RR RR EEEEEEEEE QQQQ QQ

```


STRUCDEF -- DECLARATION FILE FOR DATA STRUCTURE DEFINITION
AND ACCESS MACROS USED IN THE VAX DEBUGGER

Version: 'V04-000'

```

*****
*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

WRITTEN BY
Bert Beander August, 1981.

MODULE FUNCTION:
This REQUIRE file contains all macros used in defining and accessing
data structures (BLISS BLOCKs) in the VAX Debugger. These symbolic
names should always be used in BLISS Field-References.

DATA STRUCTURE DEFINITION AND ACCESS

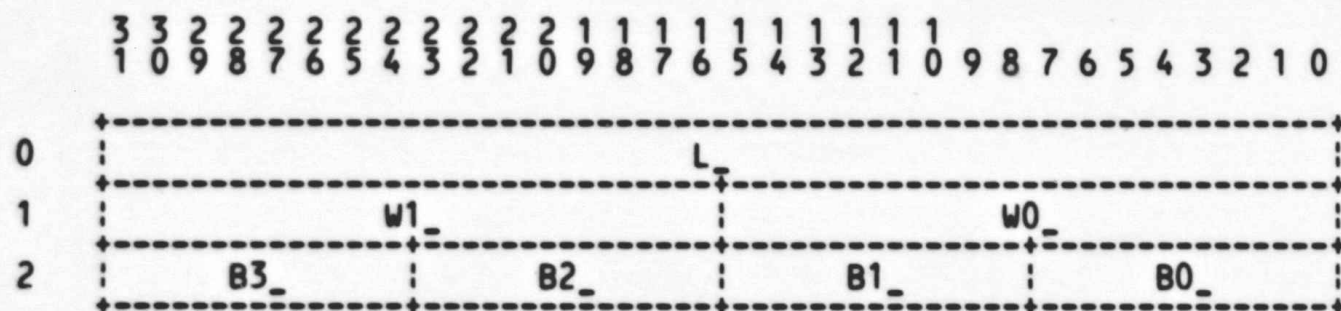
The following macros must be used in defining field names for all data structures in the Debugger. These macros supply the position, size, and sign-extension values when used in FIELD declarations for BLOCK and BLOCKVECTOR data structures. The various generic forms (as specified by the letters in the names) are as follows:

- A Materialized address
- L Longword
- W Zero-extended word
- B Zero-extended byte
- V Zero-extended bit field
- SW Sign-extended word
- SB Sign-extended byte
- SV Sign-extended bit field

The "A" form should be used whenever the field being defined is such that only the address of the field may be materialized in a structure reference; that is, fetch and store operations on the field are not valid. An example of such a field is an ASCII string.

Each of the "V" and "SV" forms take one or two parameters. The first parameter is the bit position within the longword (or byte) and the second is the field size in bits. The second parameter is optional; if omitted, it defaults to 1. Thus V_(5) means bit 5 while V_(5,3) means the 3-bit field starting at bit 5 and ending at bit 7. Bit positions are counted from the low-order (least significant) end of the longword, starting at zero.

The following data structure picture shows the locations of the various fields that can be specified. Note how the bit positions are numbered along the top of the illustration.



MACRO

- A_ = 0, 0, 0 %, ! Address of a longword
- A0_ = 0, 0, 0 %, ! Address of byte 0
- A1_ = 8, 0, 0 %, ! Address of byte 1


```

A2_ = 16, 0, 0 %, ! Address of byte 2
A3_ = 24, 0, 0 %, ! Address of byte 3

L_ = 0, 32, 0 %, ! Longword
W_ = 0, 16, 0 %, ! Word, zero-extended
B_ = 0, 8, 0 %, ! Byte, zero-extended

W0_ = 0, 16, 0 %, ! Word 0 zero-extended
W1_ = 16, 16, 0 %, ! Word 1 zero-extended

B0_ = 0, 8, 0 %, ! Byte 0 zero-extended
B1_ = 8, 8, 0 %, ! Byte 1 zero-extended
B2_ = 16, 8, 0 %, ! Byte 2 zero-extended
B3_ = 24, 8, 0 %, ! Byte 3 zero-extended

V_(P,S) = P, %IF %NULL(S) %THEN 1 %ELSE S %FI, 0 %, ! Unsigned bit field
V0_(P,S) = P, %IF %NULL(S) %THEN 1 %ELSE S %FI, 0 %, ! Bits in B0_
V1_(P,S) = (P+8), %IF %NULL(S) %THEN 1 %ELSE S %FI, 0 %, ! Bits in B1_
V2_(P,S) = (P+16), %IF %NULL(S) %THEN 1 %ELSE S %FI, 0 %, ! Bits in B2_
V3_(P,S) = (P+24), %IF %NULL(S) %THEN 1 %ELSE S %FI, 0 %, ! Bits in B3_

SW_ = 0, 16, 1 %, ! Word, sign-extended
SB_ = 0, 8, 1 %, ! Byte, sign-extended

SW0_ = 0, 16, 1 %, ! Word 0 sign-extended
SW1_ = 16, 16, 1 %, ! Word 1 sign-extended

SB0_ = 0, 8, 1 %, ! Byte 0 sign-extended
SB1_ = 8, 8, 1 %, ! Byte 1 sign-extended
SB2_ = 16, 8, 1 %, ! Byte 2 sign-extended
SB3_ = 24, 8, 1 %, ! Byte 3 sign-extended

SV_(P,S) = P, %IF %NULL(S) %THEN 1 %ELSE S %FI, 1 %, ! Signed bit field
SV0_(P,S) = P, %IF %NULL(S) %THEN 1 %ELSE S %FI, 1 %, ! Bits in B0_
SV1_(P,S) = (P+8), %IF %NULL(S) %THEN 1 %ELSE S %FI, 1 %, ! Bits in B1_
SV2_(P,S) = (P+16), %IF %NULL(S) %THEN 1 %ELSE S %FI, 1 %, ! Bits in B2_
SV3_(P,S) = (P+24), %IF %NULL(S) %THEN 1 %ELSE S %FI, 1 %; ! Bits in B3_

```

```

! END OF STRUCDEF.REQ

```


This page contains a grid of 100 small, faint diagrams or code snippets, arranged in 10 rows and 10 columns. The diagrams are mostly illegible due to low contrast and small size. Several larger, more legible text blocks are interspersed within the grid:

- STRUCDEF REQ**: Located in the upper-middle section.
- TEMPREQ REQ**: Located in the middle section.
- DBGADDEXP LIS**: Located in the lower-middle section.
- DBGATSGN LIS**: Located in the lower-middle section.
- SSIDEF REQ**: Located in the lower section.

The diagrams appear to be technical drawings or code listings related to the VAX/VMS V4.0 system.