```
RRRRRRRRRRR   MMM       MMM   SSSSSSSSSSSS
RRRRRRRRRRR   MMM       MMM   SSSSSSSSSSSS
RRRRRRRRRRR   MMM       MMM   SSSSSSSSSSSS
RRR       RRR MMMMMM MMMMMM SSS
RRR       RRR MMMMMM MMMMMM SSS
RRR       RRR MMMMMM MMMMMM SSS
RRR       RRR MMM MMM MMM SSS
RRR       RRR MMM MMM MMM SSS
RRR       RRR MMM MMM MMM SSS
RRRRRRRRRRR   MMM       MMM   SSSSSSSSS
RRRRRRRRRRR   MMM       MMM   SSSSSSSSS
RRRRRRRRRRR   MMM       MMM   SSSSSSSSS
RRR   RRR     MMM       MMM          SSS
RRR   RRR     MMM       MMM          SSS
RRR   RRR     MMM       MMM          SSS
RRR     RRR   MMM       MMM          SSS
RRR     RRR   MMM       MMM          SSS
RRR     RRR   MMM       MMM          SSS
RRR       RRR MMM       MMM   SSSSSSSSSSSS
RRR       RRR MMM       MMM   SSSSSSSSSSSS
RRR       RRR MMM       MMM   SSSSSSSSSSSS
```

```
RRRRRRRR    MM      MM   SSSSSSSS    SSSSSSSS   HH      HH   RRRRRRRR
RRRRRRRR    MM      MM   SSSSSSSS    SSSSSSSS   HH      HH   RRRRRRRR
RR      RR  MMMM  MMMM  SS          SS          HH      HH   RR      RR
RR      RR  MMMM  MMMM  SS          SS          HH      HH   RR      RR
RR      RR  MM  MM  MM  SS          SS          HH      HH   RR      RR
RR      RR  MM  MM  MM  SS          SS          HH      HH   RR      RR
RRRRRRRR    MM      MM   SSSSSS      SSSSSS      HHHHHHHHHH   RRRRRRRR
RRRRRRRR    MM      MM   SSSSSS      SSSSSS      HHHHHHHHHH   RRRRRRRR
RR  RR      MM      MM         SS          SS    HH      HH   RR  RR
RR  RR      MM      MM         SS          SS    HH      HH   RR  RR
RR    RR    MM      MM         SS          SS    HH      HH   RR    RR    ....
RR    RR    MM      MM         SS          SS    HH      HH   RR    RR    ....
RR      RR  MM      MM   SSSSSSSS    SSSSSSSS   HH      HH   RR      RR    ....
RR      RR  MM      MM   SSSSSSSS    SSSSSSSS   HH      HH   RR      RR    ....


     SSSSSSSS   DDDDDDD     LL
     SSSSSSSS   DDDDDDD     LL
   SS          DD     DD    LL
   SS          DD     DD    LL
   SS          DD     DD    LL
   SS          DD     DD    LL
     SSSSSS    DD     DD    LL
     SSSSSS    DD     DD    LL
         SS    DD     DD    LL
         SS    DD     DD    LL
         SS    DD     DD    LL
         SS    DD     DD    LL
   SSSSSSSS    DDDDDDD     LLLLLLLLLL
   SSSSSSSS    DDDDDDD     LLLLLLLLLL
```

```
{      Sbegin  rmsshr,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.          *
{*                                                                    *
{*                                                                    *
{***********************************************************************
{
```

/*

```
{
{       rms shared file structure definitions
{
{ Modified By:
{
{       V03-009 SHZ0005         Stephen H. Zalewski      05-Aug-1983
{               Add a one byte flag field to GBSB.
{
{       V03-008 SHZ0004         Stephen H. Zalewski      28-Jun-1983
{               Change size of RMS facility code in SFSB.
{
{       V03-007 KBT0489         Keith B. Thompson         9-Feb-1983
{               Add FAB and SHR field to sfsb
{
{       V03-006 KBT0325         Keith B. Thompson        10-Sep-1982
{               Remove all SO sharing structure definitions
{
{       V03-005 SHZ0003         Stephen H. Zalewski,     12-Aug-1982  14:09
{               Added further definitions to GBSB.
{
{       V03-004 SHZ0002         Stephen H. Zalewski,     11-Aug-1982  21:10
{               Changed name of $GBLB to $GBSB.
{
{       V03-003 SHZ0001         Stephen H. Zalewski      10-Aug-1982
{               Add GBLB definitions.
{
{       V03-002 KBT0081         Keith B. Thompson        12-Jul-1982
{               Add b_curlck, l_ebk and l_hbk to the sfsb.  Also fix
{               revision number in jwh0001
{
{       V03-001 JWH0001         Jeffrey W. Horn          24-Mar-1982
{               Change SIFB to reflect journaling changes to the IFB.
{
```
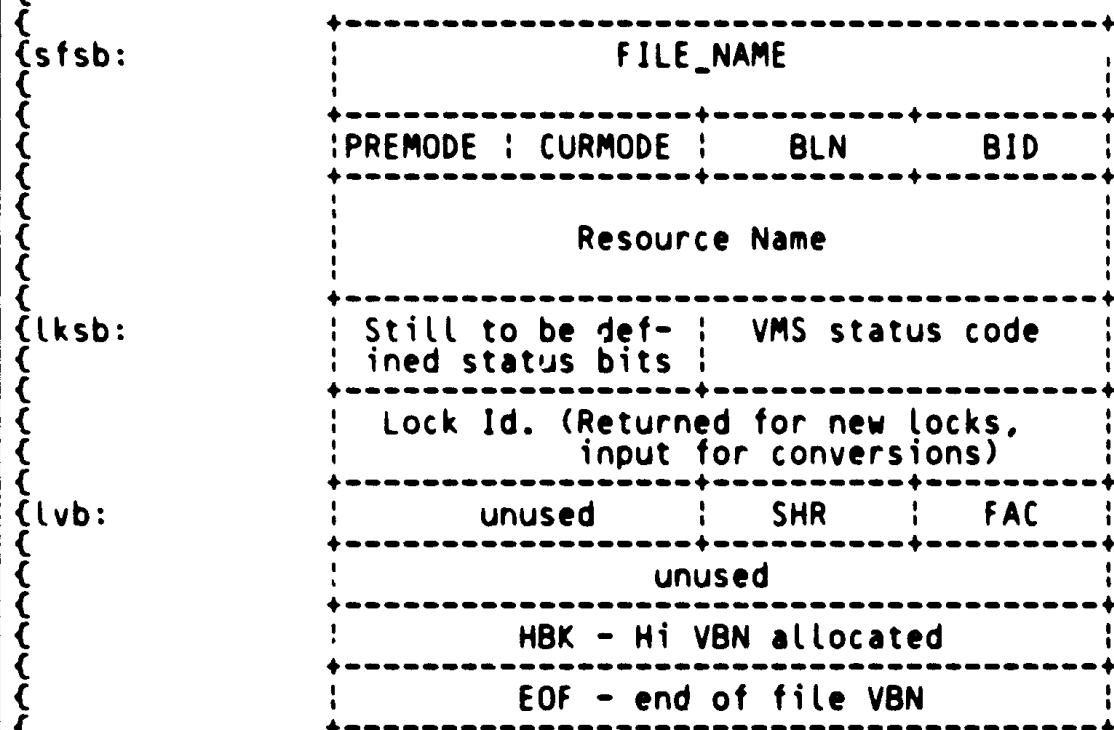
```
{
{        SFSB field definitions - shared file synchronization block
{
{        The SFSB contains the information necessary to lock a file for
{        RMS synchronization via the VAX/VMS Lock Manager.
{
{                    +------------------------------------------+
{                    :                 FILE_NAME                :
{sfsb:               :                                          :
{                    +-----------------+-----------+------------+
{                    :PREMODE : CURMODE :    BLN        BID     :
{                    +-----------------+-----------+------------+
{                    :                                          :
{                    :              Resource Name               :
{                    :                                          :
{                    +-----------------+------------------------+
{lksb:               : Still to be def-:   VMS status code      :
{                    : ined status bits:                        :
{                    +-----------------+------------------------+
{                    : Lock Id. (Returned for new locks,        :
{                    :        input for conversions)            :
{                    +--------------------+---------+-----------+
{lvb:                :      unused        :   SHR   :   FAC     :
{                    +--------------------+---------+-----------+
{                    :                  unused                  :
{                    +------------------------------------------+
{                    :          HBK - Hi VBN allocated          :
{                    +------------------------------------------+
{                    :          EOF - end of file VBN           :
{                    +------------------------------------------+
{
{
module $SFSBDEF;


aggregate SFSBDEF structure fill prefix SFSB$;
    FILENAME_OVERLAY union fill;
        FILENAME quadword unsigned;                         /* descriptor of shared file resource name.
                                                            /*  resource name is NODE, DEVICE, FILE_ID
                                                            /*  points to RESNAM, below

        FILENAME_FIELDS structure fill;
            NAME_LEN word unsigned;                         /* subfield to address descriptor length field
            FILL_3 byte dimension 2 fill prefix SFSBDEF tag $$;
            "ADDRESS" longword unsigned;                    /* subfield to address descriptor address field
        end FILENAME_FIELDS;
    end FILENAME_OVERLAY;
    BID byte unsigned;                                      /* block id
    constant BID          equals 16  prefix SFSB tag $C;    /* sfsb code
    BLN byte unsigned;                                      /* block length in longwords
    CURMODE byte unsigned;                                  /* Mode of the current lock
    PREMODE byte unsigned;                                  /* Mode of the previous lock
    RESNAM_OVERLAY union fill;
        RESNAM character length 32;                         /* 32 bytes for name of shared resource
        RESNAM_FIELDS structure fill;
            FAC_CODE longword unsigned;                     /* RMS facility code (RMS$)
            FID_NUM word unsigned;                          /* file id word one
```

```
            FID_SEQ word unsigned;                           /* file id word two
            FID_RVN word unsigned;                           /* file id word three
            constant FIX_LEN    equals 10  prefix SFSB tag $C;/* 10 bytes of fixed size data
            DEV_NAM character length 22;                     /* 22 bytes remain to hold device id (node$device_name)
        end RESNAM_FIELDS;
    end RESNAM_OVERLAY;
    LKSB_OVERLAY union fill;
        LKSB longword unsigned;                              /* lock status block
        LKSB_FIELDS structure fill;
            STATUS word unsigned;                            /* VMS status code
            S_BITS word unsigned;                            /* various status bits
        end LKSB_FIELDS;
    end LKSB_OVERLAY;
    LOCK_ID longword unsigned;                               /* second longword of LKSB is the lock id
    LVB_OVERLAY union fill;
        LVB longword unsigned dimension 4;                   /* lock value block
        constant BLN equals . prefix SFSB$ tag K;            /* length of sfsb
        constant BLN equals . prefix SFSB$ tag C;            /* length of sfsb
/*
/*      keep the next two fields in same order as they are in FAB
/*
        LVB_FIELDS structure fill;
            FAC byte unsigned;                               /* fac bits from FAB
            SHR byte unsigned;                               /* sharing bits (from FAB SHR field)
            FILL_1 word fill prefix SFSBDEF tag $$;          /* spare
            FILL_2 longword fill prefix SFSBDEF tag $$;      /* spare
            HBK longword unsigned;                           /* high block
            EBK longword unsigned;                           /* end of file
        end LVB_FIELDS;
    end LVB_OVERLAY;
end SFSBDEF;

end_module $SFSBDEF;

module $GBSBDEF;
```
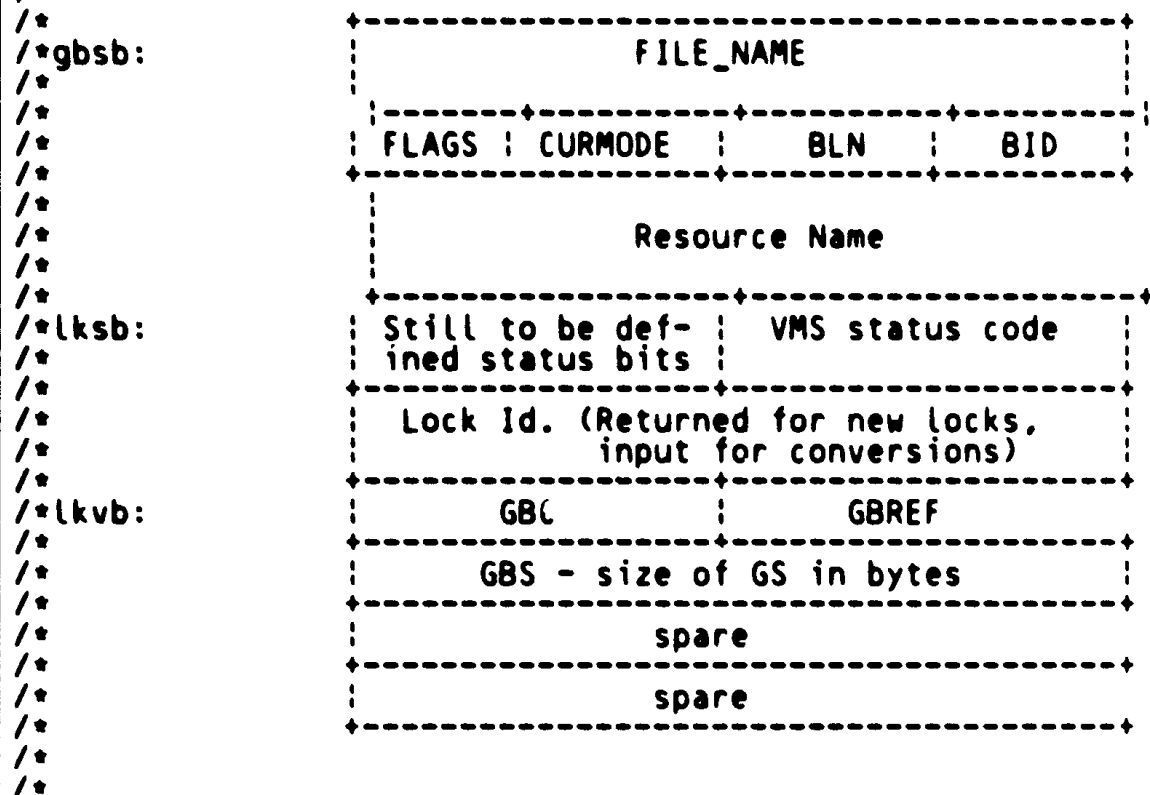
```
/*
/*          GBSB field definitions - global buffer synchronization block                  /*--
/*
/*          The GBSB contains the information necessary to determine if a
/*          global section is already open for a file on a given node, and
/*          is used for synchronizing access to the global section.
/*
/*          +--------------------------------------------------+
/*gbsb:      :                  FILE_NAME                       :
/*           :                                                  :
/*           :--------+---------+----------+----------:
/*           : FLAGS  : CURMODE :   BLN    :   BID    :
/*           +--------+---------+----------+----------+
/*           :                                        :    :
/*           :              Resource Name             :    :
/*           :                                        :
/*           +--------------------------+-------------+
/*lksb:      : Still to be def- : VMS status code     :
/*           : ined status bits :                     :
/*           +------------------+--------------------+
/*           : Lock Id. (Returned for new locks,      :
/*           :          input for conversions)        :
/*           +--------------------------+-------------+
/*lkvb:      :      GBC        :       GBREF          :    :
/*           +----------------+---------------------+
/*           :        GBS - size of GS in bytes       :
/*           +----------------------------------------+
/*           :                 spare                  :    :
/*           +----------------------------------------+
/*           :                 spare                  :    :
/*           +----------------------------------------+
/*
/*
```

```
aggregate GBSBDEF structure fill prefix GBSB$;
    FILENAME_OVERLAY union fill;                                                             end
        FILENAME quadword unsigned;                         /* descriptor of shared file resource name.
                                                            /*  resource name is NODE, DEVICE, FILE_ID
                                                            /*  points to RESNAM, below

        FILENAME_FIELDS structure fill;
            NAME_LEN word unsigned;                         /* subfield to address descriptor length field
            FILL_3 byte dimension 2 fill prefix GBSBDEF tag $$;
            "ADDRESS" longword unsigned;                    /* subfield to address descriptor address field
        end FILENAME_FIELDS;
    end FILENAME_OVERLAY;
    BID byte unsigned;                                      /* block id
    constant BID        equals 9  prefix GBSB tag $C;       /* gbsb code
    BLN byte unsigned;                                      /* block length in longwords
    CURMODE byte unsigned;                                  /* Mode of the current lock
    FLAGS_OVERLAY union fill;
        FLAGS byte unsigned;                                /* spare
        FLAGS_BITS structure fill;
            NOTACCESSED bitfield mask;                      /* Process has already decremeted access count for GBS.
        end FLAGS_BITS;
    end FLAGS_OVERLAY;
```

```
    RESNAM character length 32;                      /* 32 bytes for name of shared resource
    LKSB_OVERLAY union fill;                          /* lock status block
        LKSB longword unsigned;                       /* lock status block
        LKSB_FIELDS structure fill;
            STATUS word unsigned;                     /* VMS status code
            S_BITS word unsigned;                     /* various status bits
        end LKSB_FIELDS;
    end LKSB_OVERLAY;
    LOCK_ID longword unsigned;                        /* second longword of LKSB is the lock id
    GBC word unsigned;                                /* Number of global buffers in section.
    GBREF word unsigned;                              /* Number of accessors to global section.
    GS_SIZE longword unsigned;                        /* Size of global section in bytes.
    FILL_1 longword fill prefix GBSBDEF tag $$;       /* spare
    FILL_  longword fill prefix GBSBDEF tag $$;       /* spare
    constant BLN equals . prefix GBSB$ tag K;         /* length of gbsb
    constant BLN equals . prefix GBSB$ tag C;         /* length of gbsb
end GBSBDEF;

end_module $GBSBDEF;
```

RMSFILSTR
SDL

RMSMAC
REQ

RMSINTSTR
SDL

RMSUSR
SDL

NWADEF
MDL

RMSFWADEF
SDL

RMSSHR
SDL