```
                                       JMT$DISPATCHING_PRIORITY          Type is subrange  0 ..   15
READYING_TASK_PRIORITY                Type is field of a record  Length =  1(bytes)  Offset =  17(16) : 0
                                       JMT$DISPATCHING_PRIORITY          Type is subrange  0 ..   15
IJL_THREAD                            Type is field of a record  Length =  2(bytes)  Offset =  18(16) : 0
                                       OST$TASK_INDEX                    Type is subrange  0 ..   4095
END_OF_WAIT_TIME                      Type is field of a record  Length =  6(bytes)  Offset =  1A(16) : 0
                                        Type is subrange  0 ..  281474976710655
```



```
OFFSET BIT     LINE BINARY        SOURCE STATEMENT                                              CPU ASSEMBLY        LEVEL
    0                             mtm$monitor_interrupt_handler IDENT
    0                             .
    0                             ...............................................................................
    0                             .                          MONITOR INTERRUPT HANDLER
    0                             .  This module is the top-level control module in NOSVE monitor. It contains
    0                             .  the procedures that do the following:
    0                             .        - EXCHANGE to job mode
    0                             .        - decode the job mode MCR when it exchanges back to monitor and
    0                             .          call the CYBIL procedures to process the request or condition.
    0                             .        - Process EXCHANGE (170 PP) requests by giving control to the 170
    0                             .          partner
    0                             .        - Process TRAPS that occur in monitor mode.
    0                             .        - Call the CPU dispatcher to change the current task
    0                             .        - Handle (most) dual CPU interlocking.
    0                             .        - This module is the first OS module to begin executing at deadstart. It
    0                             .          performs some basic system initialization functions before exchanging
    0                             .          to job mode to continue deadstart.
    0                             .        - The monitor stack and exchange packages are defined in this module.
    0                             .        - The monitor request table is defined in this module.
    0                             .
    0                             .
    0                             .  NOTE: This module must be the first module on the OSF$MONITOR library and
    0                             .        must be the first module loaded into the monitor address space.
    0                             ...............................................................................
    0                             .
    0                             .
    0                                     page
```

0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890

```
OFFSET BIT   BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY        LEVEL
   o                           .
   o                           ................ begin common deck SYA$CONSTANTS ........................
   o                           .
   o                           .
   o                           .
   o                           .  Define constants for sizes of CYBIL tables - OS tables whose
   o                           .       sizes must be known by assembly language modules. Unless
   o                           .       otherwise stated, the sizes given here can be larger than
   o                           .       the actual size.
   o                           .
   o                           .    **** WARNING - (xcbsize + jstlen*8 + jrootsiz) must be <= 2048
   o                           .           It must fix in a page else the ST may not be in contiguous memory.
   o                           .
   o         0000000000000400  xcbsize  equ      1024              .Size of XCB.
   o         0000000000000030  sdtxsize equ      48                .Size of SDTX entry.
   o         0000000000000118  statsize equ      280               .Size of OST$STATUS.
   o         00000000000000FF  ajllen   equ      255               .Max number of AJL entries - this constant
   o                           .                                    is used to set the size of the monitor
   o                           .                                    seg tbl. The actual size of the AJL can
   o                           .                                    be less than or equal to this value.
   o         0000000000000100  jrootsiz   equ      256             .Length of JMT$JOB_CONTROL_BLOCK
   o                           .
   o                           .
   o                           .
   o                           .
   o                           .   Define monitor constants
   o                           .
   o         0000000000001A2C  mstksize  equ      6700             .Length of monitor stack
   o         0000000000000020  mstkfram  equ      32               .Length of monitor stack frame
   o         0000000000000400  jstksiz1  equ      1024              .Length of job stack for ring 1
   o         0000000000000800  jstksiz2  equ      2048             .Length of job stack for ring 2
   o         0000000000000200  jstksiz3  equ      512             .Length of job stack for ring 3
   o         0000000000000020  jstkfram  equ      32               .Length of job stack frame
   o         000000000000005E  jstlen    equ      94               .Number of segments in Job Segtll
   o         0000000000000014  mstlen    equ      20               .Number of segments in Monitor Segtll
   o         0000000000000013  a170_stl  equ      19               . Number of segments in a170 seg table
   o                           .
   o                           .
   o                           .   Define 'magic' segment numbers. These equates MUST agree with
   o                           .   the actual segment numbers assigned during system generation.
   o                           .   WARNING: in most cases, no run time checks are made to see if
   o                           .   the constants defined here are correct.
   o                           .
   o         0000000000000000  snptmtr   equ      0                .Page table seg num in monitor.
   o         0000000000000002  sn170mcb  equ      2                .170 segment number with cache bypass attribute.
   o         0000000000000003  snnosmtr  equ      3                .NOS segment number in MTR mode.
   o         0000000000000004  snsfmtr   equ      4                .NOS stack segment number in MTR mode.
   o         0000000000000005  snnthmtr  equ      5                .Nos trap handler segment number in MTR mode.
   o                           .
   o         0000000000000003  snjfjob   equ      3                .Job fixed in job mode.
   o                           .
   o                           .   Define a170 segment numbers for NOS, EI and EIE.
   o                           .
   o         0000000000000003  snnos170  equ    3            .Nos segment number

             0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT   BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY        LEVEL
   o         0000000000000004  snsf170   equ    4            .NOS stack segment number
   o         0000000000000005  snnth170  equ    5            .Nos trap handler segment number
   o                           .
   o                           .
   o                           .   Define operating system constants
   o                           .
   o                           .
   o                           .
   o         000000000000FFFC  m_mtrmsk   equ      0fffc(16)      .Monitor mode MM
   o         000000000000FFFC  j_mtrmsk   equ      0fffc(16)      .Job mode MM
   o         000000000000FF7F  m_usrmsk   equ      0ff7f(16)      .Monitor mode UM
   o         000000000000FF77  j_usrmsk   equ      0ff77(16)      .Job mode UM
   o         000000000000EDFF  m_usrabt   equ      0edff(16)      .Fatal UCR conditions, monitor
   o         000000000000CC00  j_usrabt   equ      0cc00(16)      .Fatal UCR conditions, job
   o         0000000000005B2C  m_mcrhlt   equ      05b2c(16)      .MCR conditions that cause halt, monitor.
   o         000000000000E000  j_mcrhlt   equ      0e000(16)      .MCR conditions that cause halt, job.
   o         0000000000000490  m_mcrasy   equ      00490(16)      .MCR conditions that are asynchronous.
   o         0000000000001B0C  j_mcrusr   equ      01b0c(16)      .MCR conditions that are normally
   o                           .                                   processed by the job trap handler.
   o         0000000000002000  m_mcrsw    equ      02000(16)      .MCR condition: short_warning
   o                           .
   o                           .
   o                           .   Define offsets for referencing fields in the job table segments
   o                           .
   o         0000000000000100  jr_mxcb    equ       jrootsiz        .XCB for Job Monitor.
   o                           .
   o                           .
   o                           .     PROC Definitions for initializing exchange packages
   o                           .
   o                                     PROC                                                          PDEF
   o                           xpareg    pname                                                         PDEF
   o                                     do  sn:(f:(2,2))=sn:(nil)                                     PDEF
   o                                       org  f:(2,0)+f:(2,1)*8+10                                   PDEF
   o                                       vfd,16,32  0ffff(16),080000000(16)                          PDEF
   o                                     else                                                          PDEF
   o                                       org   f:(2,0)+f:(2,1)*8+10                                  PDEF
   o                                       address  r,f:(2,2)+f:(2,3)                                  PDEF
   o                                     dend                                                          PDEF
   o                                     PEND                                                          PDEF
   o                                     PROC                                                          PDEF
   o                           xpa       pname                                                         PDEF
   o                                     do  sn:(f:(2,2))=sn:(nil)                                     PDEF
   o                                       org  f:(2,0)+f:(2,1)                                        PDEF
   o                                       vfd,16,32  0ffff(16),080000000(16)                          PDEF
   o                                     else                                                          PDEF
   o                                       org   f:(2,0)+f:(2,1)                                       PDEF
   o                                       address  r,f:(2,2)+f:(2,3)                                  PDEF
   o                                     dend                                                          PDEF
   o                                     PEND                                                          PDEF
   o                                     PROC                                                          PDEF
   o                           xpv       pname                                                         PDEF
   o                                     org   f:(2,0)+f:(2,1)                                         PDEF
   o                                     vfd,f:(2,3) f:(2,2)                                           PDEF
   o                                     PEND                                                          PDEF

             0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT   LINE BINARY   SOURCE STATEMENT                                              CPU ASSEMBLY   LEVEL
     0                     .
     0                     ................ end common deck SYA$CONSTANTS .........................
     0                     .
     0                     .  Deck SYA$CYBIL_INTERFACE_PROCEDURES follows but is not listed.
```



```
OFFSET BIT   LINE BINARY          SOURCE STATEMENT                                       CPU ASSEMBLY   LEVEL
     0                            ..................begin common deck MTA$CPU_STATE_TABLE................
     0                            . WARNING! WARNING! WARNING! WARNING! WARNING! WARNING! WARNING!
     0                            .      If this type is changed the type OST$CPU_STATE_TABLE must reflect a
     0                            .      corresponding change!
     0                            .
     0        0000000000000002 maxcst    equ    2          .maximum cst entries.
     0        0000000000000110 cstsize   equ    34*8       .Size of CST
     0                            .
     0                            .. Define offsets for the various fields in CST.
     0                            .
     0        0000000000000000 fill      equ    0
     0        0000000000000001 prior180  equ    1          .CPU priority (180)
     0        0000000000000002 dsprior   equ    2          .dual-state priority and subpriority
     0        0000000000000004 memport   equ    4          .processor memory port.
     0        0000000000000005 lpid      equ    5          .logical processor id
     0        0000000000000006 cpu_stat  equ    6          .cpu state (on/off/down)
     0        0000000000000007 nextstat  equ    7          .next cpu state (on/off/down)
     0        0000000000000008 cpwell    equ    1*8        .Changing value = CPU alive.
     0        0000000000000010 taskid    equ    2*8        .Taskid of current task.
     0        0000000000000013 ajlo      equ    2*8+3      .AJLO of current task,
     0        000000000000001E cp_state  equ    3*8+6      .current/requested processor states: (2 bytes)
     0        0000000000000014 dualstat  equ    2*8+4      .NOS JPS if dual state, 0 if not dual state.
     0        0000000000000018 jcbp      equ    3*8        .JCB_P to current task's JCB.
     0        0000000000000020 xcbp      equ    4*8        .Pointer to XCB of current task.
     0        0000000000000028 xcbrma    equ    5*8        .RMA of current task XP.
     0        0000000000000030 discntl   equ    6*8        .Dispatch control info.
     0        0000000000000038 cptime    equ    7*8        .Max cptime for current task.
     0        0000000000000040 jtime     equ    8*8        .Time in job mode for current task.
     0        0000000000000048 mtime     equ    9*8        .Time in monitor mode for current task.
     0        0000000000000050 ext_int   equ    10*8       .type of external interrupt request
     0        0000000000000051 idlecode  equ    10*8+1     .system idle code
     0        0000000000000052 lpid8     equ    10*8+2     .LPID times 8.
     0        0000000000000058 cachtim   equ    11*8       .time cache purged
     0        0000000000000060 maptim    equ    12*8       .time map purged
     0        0000000000000068 mps       equ    13*8       .Monitor MPS.
     0        0000000000000070 elem_id   equ    14*8       .processor element id
     0        0000000000000078 ijlo      equ    15*8       .IJL ordinal of current task.
     0        000000000000007A ijlep     equ    15*8+2     .Pointer to IJL entry.
     0        0000000000000080 idlstats  equ    16*8       .cpu idle statistics
     0        00000000000000A0 tracectl  equ    20*8       .trace control
     0        00000000000000A8 termmess  equ    21*8       .termination message record
     0        00000000000000F8 cpstreas  equ    31*8       .reason for current state of CPU
     0        00000000000000F9 cpprepro  equ    31*8+1     .CPU preprocessing state
     0        00000000000000FA cpu_spin  equ    31*8+2     .TRUE = CPU should spin without doing any useful work
     0        00000000000000FB prevstat  equ    31*8+3     .previous CPU state
     0        00000000000000FC log_stat  equ    31*8+4     .TRUE = Log CPU state change
     0        0000000000000100 dpint     equ    32*8       .integer value of the dispatching priority
     0        0000000000000108 dummy4    equ    33*8       .dummy field for later use
     0                            .
     0        0000000000000030 caldisp   equ    discntl    .This byte can be set to '1' to force a call to
     0                                                    . the CPU dispatcher.
     0        0000000000000034 asyncp    equ    discntl+4  .this byte is set true when async events
     0                                                    or external interrupts are pending.
     0                            .  define offsets into ext_int
```

```
OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                                CPU ASSEMBLY                LEVEL
      0          403 0000000000000000 tsk_sw     equ    0           .switch task
      0          404 0000000000000001 pur_ca     equ    1           .purge_cache
      0          405 0000000000000002 pur_map    equ    2           .purge_map
      0          406 0000000000000003 step_pr    equ    3           .step processor
      0          407                  .
      0          408                  .  define offsets into idlstats
      0          409 0000000000000000 idl_noio   equ    0           .cumulative idle time with no IO active
      0          410 0000000000000008 idl_w_io   equ    1*8         .cumulative idle time with IO active
      0          411 0000000000000010 idlstart   equ    2*8         .starting time for current idle
      0          412 0000000000000018 idletype   equ    3*8         .type of cpu idle: with or without IO active
      0          413 0000000000000019 idle_cnt   equ    3*8+1       .number of times the cpu goes idle
      0          414                  .
      0          415                  .  define offsets into termmess
      0          416 0000000000000000 tm_size    equ    0           .size of termination message
      0          417 0000000000000001 tm_un_id   equ    0*8+1       .unique identifier
      0          418 0000000000000002 tm_text    equ    0*8+2       .text of termination message
      0          419                  .
      0          420                  .  value assigned cpu state and next cpu state (cp_stat, nextstat)
      0          421                  .  also known as CYBIL ordinal type CMT$ELEMENT_STATE
      0          422 0000000000000000 on         equ    0
      0          423 0000000000000001 off        equ    1
      0          424 0000000000000002 down       equ    2
      0          425                  .
      0          426                  .  define offsets into cpu_state
      0          427 0000000000000000 cp_curst   equ    0*8         .current cpu state
      0          428 0000000000000001 cp_nxtst   equ    0*8+1       .next cpu state
      0          429                  .
      0          430                  .  value assigned current, next cpu_state
      0          431 0000000000000000 running    equ    0
      0          432 0000000000000001 stepped    equ    1
      0          433                  ..................end common deck MTA$CPU_STATE_TABLE.....................
      0          434                  .
      0          435                  .  Define equates used to reference fields in the SMU Communications Block (SCB)
      0          436                  .   Note - only fields referenced from assembly language are defined. See
      0          437                  .   the deck MTDSCB for the complete definition.
      0          438                  .
      0          439 0000000000000190 scbsize    equ    50*8        .Length of SCB.
      0          440 0000000000000009 scb_cpus   equ    1*8+1       .Set of CPUs which are logically ON.
      0          441 000000000000000B scbvecsim  equ    1*8+3       .Vector simulation control (3 bytes)
      0          442 0000000000000010 scbidler   equ    2*8         .Non-zero if IDLE is requested.
      0          443 0000000000000012 scbstepr   equ    2*8+2       .Non-zero if STEP is requested.
      0          444 0000000000000018 scbnsrv    equ    3*8         .Flag to indicate 180 is alive and well.
      0          445                  .
```

```
                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                                CPU ASSEMBLY                LEVEL
      0          447                  ..
      0          448                  .
      0          449                  .  Common Deck osa$dual_state_control_block
      0          450                  .  Defines the Dual State Control Block offsets.
      0          451                  .
      0          452
      0          453
      0          454                  ..
      0          455                  .         DSCBW - Macro used to define offsets to the EI control block (EICB).
      0          456                  .
      0          457                  .         CALLING_SEQUENCE:
      0          458                 .symbol  dscbw     length
      0          459                  .
      0          460                  .         PARAMETERS:
      0          461                  .                   OFFSET_NAME = defines symbol that defines location in EICB of
      0          462                  .                                 that value.
      0          463                  .                   LENGTH = length in words associated with symbol value.
      0          464                  .
      0          465                  .         NOTE: Offsets are defined in bytes.
      0          466                  .
      0          467
      0          468                  proc                                                                                    PDEF
      0          469                 dscbw    pname                                                                           PDEF
      0          470                 f:(0)    equ       dscb_nxt*8                                                            PDEF
      0          471                 dscb_nxt set       dscb_nxt+f:(2)                                                        PDEF
      0          472                  pend                                                                                    PDEF
      0          473 0000000000000000 dscb_nxt set       0
      0          474                  .
      0          475                  .
      0          476                  d7ty     dscbw     1                   .C170 OS type
      0          482                  d7jp     dscbw     2                   .C170 job information for cpu 0
      0          488                  d7st     dscbw     1                   .C170 operating system status
      0          494                  d7rs     dscbw     3                   .C170 SCD/MDD communication
      0          500                  d7cm     dscbw     2                   .central memory allocation
      0          506                  d7sv     dscbw     6                   .C170 save area
      0          512                  .
      0          513                  d8ty     dscbw     1                   .C180 operating system type
      0          519                  d8tm     dscbw     2                   .Time spend in C180 OS
      0          525                  d8jp     dscbw     2                   .C180 job parameters for cpu 0
      0          531                  d8st     dscbw     1                   .C180 operating system status
      0          537                  d8ds     dscbw     3                   .deadstart parameters
      0          543                  d8sv     dscbw     6                   .C180 OS scratch area
      0          549                  .
      0          550                  dscm     dscbw     5                   .control information block
      0          556                  dfcm     dscbw     11                  .fatal error message buffer displayed by SCD.
      0          562
      0          563 0000000000000170 DSCBL    equ       dscb_nxt*8
      0          564
      0          565                  .
      0          566                  .  Offsets for dual state deadstart.
      0          567                  .
      0          568 00000000000000A8 ds_stat  equ       d8ds                .deadstart status
      0          569 00000000000000AC ds_flag  equ       d8ds+4              .deadstart flag
      0          570 00000000000000B0 os_sfsa  equ       d8ds+8              .stack frame save area rma
```

```
                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
 OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY                LEVEL
      0           571 00000000000000B4 os_jps    equu      d8ds+12                .OS JPS rma
      0           572 00000000000000B8 ve_sfsa   equ       d8ds+16                .stack frame save area rma for VE
      0           573 00000000000000BC ve_jps    equ       d8ds+20                .VE JPS rma
      0           574                  .
      0           575                  .  Byte offsets for use by MTAMTR.
      0           576                  .
      0           577 0000000000000000 np170ty   equ       d7ty                   .date/time pointers, os type
      0           578 000000000000000E np170pr   equ       d7jp+6                 .Current 170 priority
      0           579 0000000000000096 np180pr   equ       d8jp+6                 .Current 180 priority
      0           580 0000000000000080 npxtime   equ       d8tm                   .Time not spent in NOS
      0           581 0000000000000170 dscbln    equ       dscbl                  .dscb block length
      0           582                  .
      0           583                  .  Symbol definitions for the system type and interface level.
      0           584                  .
      0           585                  .  This constant defines the psr level of the operating system with respect to DFT.
      0           586                  .  This has to be changed when the levels change.
      0           587                  .
      0           588 00000000000002E3 dft_psr   equ       739
      0           589
      0           590                  .
      0           591                  .***  End common deck SYC$COMPASS_OS_LEVELS
      0           592 0000000000000001 ost$ei    equ       1
      0           593 0000000000000001 ost$nos   equ       1
      0           594 0000000000000002 ost$nbe   equ       2
      0           595 0000000000000002 ost$nve   equ       2
      0           596 00000000000002E3 ost$psr   equ       dft_psr
      0           597
      0           598                  .  Define EICB interface version number.  I do not understand exactly what this
      0           599                  .  means but if its value is changed check the code in osm$os_environment_monitor
      0           600                  .  that references it.  Instruction retry is not attempted if less than this value.
      0           601                  .  Assume that it has something to to with the host system.
      0           602
      0           603 0000000000000002 if_versn equ        2
      0           604 0000000000000001 if_level equ        1
      0           605                  .
      0           606                  .  symbol definitions for the dscm words.
      0           607                  .
      0           608 0000000000000021 c170_due equ        21(16)                 .c170 due and due with no retry
      0           609 0000000000000024 c180_due equ        24(16)                 .c180 due and due with no retry
      0           610 0000000000000001 retry_failed equ    01(16)                 .c170=22, c180=25
      0           611 0000000000000010 retry_due equ       10(16)                 .c170=32, c180=34
      0           612                  .
      0           613                  .***  End common deck OSA$DUAL_STATE_CONTROL_BLOCK
      0           614                  ......................  begin common deck OSA$BASIC_REGISTER_EQUATES ................
      0           615                  .
      0           616                  .
      0           617                  .  Define Macros for definning X and A register equates.
      0           618                  .
      0           619                            PROC                                                                  PDEF
      0           620                  xreg      pname                                                                 PDEF
      0           621                  f:(0,0)   equ    f:(2,0)                                                        PDEF
      0           622                  f:(0,0)   atrib  #regtyp,#xreg                                                  PDEF
      0           623                            PEND                                                                  PDEF
      0           624                            PROC                                                                  PDEF

                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890
```



```
 OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY                LEVEL
      0           625                  areg      pname                                                                 PDEF
      0           626                  f:(0,0)   equ    f:(2,0)                                                        PDEF
      0           627                  f:(0,0)   atrib  #regtyp,#areg                                                  PDEF
      0           628                            PEND                                                                  PDEF
      0           629                  .
      0           630                  .  Define A and X register usage
      0           631                  .
      0           632                  a_tos     areg     0
      0           638                  a_dsp     areg     0
      0           644                  a_csf     areg     1
      0           650                  a_psa     areg     2
      0           656                  a_bindin  areg     3
      0           662                  a_plist   areg     4
      0           668                  .
      0           669                  .
      0           670                  .
      0           671                  ......................  end common deck OSA$BASIC_REGISTER_EQUATES ...................
      0           672                  .
      0           673                  .  MTA$DFT_BLOCK - This deck defines used by assembly language decks to reference      ...
                                       fields in the
      0           674                  .                     DFT. Only fields that are referenced are defined.
      0           675                  .
      0           676 0000000000000000 dftcw       equ      0                   .DFT control word.
      0           677                  .
      0           678                  .
      0           679                  .  Define keypoint class codes
      0           680                  .
      0           681 0000000000000000 oscdata     equ    0               .Data keypoint
      0           682 0000000000000001 oscunus     equ    1               .Unusual keypoint.
      0           683 0000000000000002 oscent      equ    2               .Entry keypoint.
      0           684 0000000000000003 oscexit     equ    3               .Exit keypoint.
      0           685 0000000000000004 oscdbug     equ    4               .Debug keypoint.
      0           686 0000000000000005 oscmtr      equ    5               .monitor entry/exit.
      0           687                  .
      0           688                  .           define bias for monitor exit keypoint.
      0           689                  .
      0           690 0000000000001000 oskxbias    equ    4096
      0           691                  .
      0           692                  .
      0           693                  .
      0           694                  .  Define keypoint codes used in OS assembly language decks.
      0           695                  .
      0           696                  .
      0           697 0000000000000FA0 oskpurg     equ    4000            .Used in OSAINX wil cache/map purge.
      0           698 0000000000000FA1 oskexc8     equ    4001            .Exchange to/from 180 job mode.
      0           699 0000000000000FA2 oskexc7     equ    4002            .Exchange to/from 170 job mode.
      0           700 0000000000000FA3 osktrpm     equ    4003            .Monitor mode trap.
      0           701 0000000000000FA4 osktrpj     equ    4004            .Job mode trap.
      0           702 0000000000001FA1 oskexc8x    equ    oskexc8+oskxbias
      0           703 0000000000001FA2 oskexc7x    equ    oskexc7+oskxbias
      0           704 0000000000001FA3 osktrpmx    equ    osktrpm+oskxbias
      0           705 0000000000001FA4 osktrpjx    equ    osktrpj+oskxbias
      0           706                  .
      0           707                  .

                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT   LINE BINARY           SOURCE STATEMENT                                          CPU ASSEMBLY      LEVEL
  0
  0                                .
  0                                .
  0                                .   Hardware defined constants for indexing and referencing an exchange
  0                                .   package or stack frame.
  0                                .
  0
  0          0000000000000030 sfsa_mcr  equu     48            .Offset to MCR in Stk Frame Save Area
  0          0000000000000028 sfsa_ucr  equ      40            .Offset to UCR in Stk Frame Save Area
  0          00000000000001A0 xpsize    equ      416           .Exchange package size (bytes)
  0          000000000000011A xptp      equ      282           .XP offset to Trap Pointer
  0          0000000000000122 xpdlp     equ      290           .XP offset to Debug list pointer.
  0          0000000000000110 xpstau    equ      272           .XP offset to seg Table Adr upper
  0          0000000000000118 xpstal    equ      280           .XP offset to seg Table Adr lower
  0          0000000000000120 xpdebugi  equ      288           .XP offset to debug index.
  0          0000000000000121 xpdebugm  equ      289           .XP offset to debug mask.
  0          0000000000000112 xputp     equ      274           .XP offset to UTP
  0          0000000000000010 xpflgte   equ      16            .XP offset to FLAGS and TE
  0          0000000000000030 xpmcr     equ      48            .XP offset to MCR field
  0          0000000000000028 xpucr     equ      40            .XP offset to UCR field
  0          000000000000002B xp170mf   equ      43            .XP offset to byte containing 170 mtr flag
  0          0000000000000008 xpvmid    equ      8             .XP offset to VMID field
  0          0000000000000010 xpfdesc   equ      16            .XP offset to SFSA frame descriptor.
  0          0000000000000010 xpcff     equ      16            .XP offset to CFF flag.
  0          0000000000000080 xpstl     equ      128           .XP offset to Seg Table Len
  0          0000000000000018 xpum      equ      24            .XP offset to User Mask
  0          0000000000000020 xpmm      equ      32            .XP offset to Monitor Mask
  0          0000000000000040 xpkm      equ      64            .XP offset to Keypoint Mask
  0          0000000000000058 xppit     equ      88            .XP offset to PIT (upper)
  0          0000000000000068 xpbc1     equ      104           .XP offset to Base Constant (upper)
  0          0000000000000070 xpbc2     equ      112           .XP offset to Base Constant (lower)
  0          0000000000000128 xplrn     equ      296           .XP offset to LRN
  0          0000000000000088 xpxregs   equ      136           .XP offset to first X register
  0          000000000000012A xptos     equ      298           .XP offset to Top of Stack
  0
  0                               .  Define constants for MCR and UCR mask bits.
  0
  0          0000000000000010 m_mcrsit  equ      00010(16)     .MCR masks
  0          0000000000000090 m_mcrexs  equ      00090(16)     .EXT INT and SIT
  0          000000000000A000 m_mcrhdw  equ      0a000(16)     .DUE and SHORT WARNING
  0          0000000000002000 m_mcrsrw  equ      02000(16)
  0          0000000000008000 m_mcrdue  equ      08000(16)
  0          0000000000000080 m_mcrei   equ      00080(16)
  0          0000000000000400 m_mcrexc  equ      00400(16)
  0          0000000000000040 m_mcrpf   equ      00040(16)
  0          0000000000000020 m_mcrmc1  equ      00020(16)
  0          0000000000000002 m_mcrsel  equ      00002(16)
  0          0000000000000001 m_mcrtrx  equ      00001(16)
  0          0000000000000003 m_mcrelt  equ      00003(16)     .SOFT ERROR LOG and TRAP EXCEPTION.
  0
  0          0000000000002000 m_ucrff   equ      02000(16)
  0          0000000000000400 m_ucrcff  equ      00400(16)
  0          0000000000000200 m_ucrkp   equ      00200(16)
  0          0000000000000080 m_ucrdb   equ      00080(16)

                 0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT   LINE BINARY           SOURCE STATEMENT                                          CPU ASSEMBLY      LEVEL
  0
  0                               .  Define constants for accessing processor state registers.
  0
  0          0000000000000010 r_eid     equ      010(16)       .Element id
  0          0000000000000011 r_pid     equ      011(16)       .Processor id
  0          0000000000000048 r_pta     equ      048(16)       .Page table address.
  0          0000000000000049 r_ptl     equ      049(16)       .Page table length.
  0          000000000000004A r_psm     equ      04a(16)       .Page size mask.
  0          00000000000000C2 r_te      equ      0c2(16)       .Trap enable
  0          00000000000000C0 r_td      equ      0c0(16)       .Trap disabled
  0          00000000000000C3 r_ted     equ      0c3(16)       .Trap enable delay
  0          00000000000000E0 r_cff_c   equ      0e0(16)       .Critical frame flag
  0          0000000000000061 r_jps     equ      061(16)       .Job Process State
  0          0000000000000062 r_sit     equ      062(16)       .System Interval Timer
  0          00000000000000C9 r_pit     equ      0c9(16)       .Process interval timer.
  0          0000000000000045 r_stl     equ      045(16)       .Segment Table Length
  0          0000000000000042 r_mcr     equ      042(16)       .Monitor condition register
  0          0000000000000047 r_bc      equ      047(16)       .Base constant.
  0          00000000000000CA r_kef0    equ      0ca(16)       .Keypoint enable flag - clear.
  0          00000000000000CB r_kef1    equ      0cb(16)       .Keypoint enable flag - set.
  0          00000000000000E4 r_di      equ      0e4(16)       .Debug index.
  0          00000000000000E5 r_dmr     equ      0e5(16)       .Debug mask register.
  0          00000000000000E6 r_um      equ      0e6(16)       .User mask.
  0          0000000000000060 r_mm      equ      060(16)       .Monitor mask.
  0          00000000000000C4 r_tp      equ      0c4(16)       .trap pointer.
  0          00000000000000C5 r_dlp     equ      0c5(16)       .Debug list pointer.
  0                               .
  0          0000000000000003 donthing equ      3        .Define symbol from osa$ei_constant_definitions
  0          0000000000000008 issuekpt equ      8        .Debug issue keypoint request for 170 trap handler
  0          0000000000000FA5 osktrap  equ  4005         .Trap handler keypoint request

                 0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT   LINE BINARY   SOURCE STATEMENT                                                        CPU ASSEMBLY          LEVEL
0                          ...........................................................................
0                          . XTRACE - This macro is used to keep trace information about what happens.
0                          .     The TRACE buffer is a circular buffer containing a list of the last
0                          .     256 items of interest. Items currently maintained are:
0                          .         0. exchange to job mode. (timestamp)
0                          .         1. exchange from job mode. (timestamp, MCR)
0                          .         2. trap in monitor mode. (timestamp, MCR)
0                          .         3. EXCHANGE to NOS for EXCHREQ trap. (timestamp)
0                          .         4. EXCHANGE back from NOS for EXCHREQ trap. (timestamp, MCR)
0                          .         5. Taskswitch. (timestamp, new task XP RMA)
0                          .     An entry in the trace buffer is 1 word long and contains:
0                          .         bit 0 - 3,  trace id. Same as item number in above list
0                          .         bit 4-31, data dependant on id. Usually MCR or XP RMA.
0                          .         bit 32-63, lower 32 bits of FREE RUNNING CLOCK.
0                          .
0                          .     calling sequence to macro....
0                          .          xtrace  p0,p1,p2,p3
0                          .              p0 - contains trace id (0 .. 5)
0                          .              p1 - contains data to be saved
0                          .              p2, p3 - 2 X-registers that can be used for scratch
0                          .                 WARNING - X0 cannot be used for p2.
0                          .              p4 - scratch A register
0                          .
0                          .     NOTE: While system is stepped, it uses a different trace buffer to prevent
0                          .            destroying the primary buffer that may contain useful info.
0                          ...........................................................................
0                          .
0                                     PROC                                                                               PDEF
0                          xtrace     pname                                                                              PDEF
0                          f:(0)      bss     0                                                                          PDEF
0                                     local   t1,t2                                                                      PDEF
0                                     la      f:(2,4),a_cst,tracect1+2                                                   PDEF
0                                     entp    f:(2,3),0                                                                  PDEF
0                                     cpytx   f:(2,3),f:(2,3)                                                            PDEF
0                                     sx      f:(2,3),f:(2,4),0                                                          PDEF
0                                     entz    f:(2,3)                                                                    PDEF
0                                     entp    f:(2,2),f:(2,0)                                                            PDEF
0                                     do      sn:(f:(2,1))=sn:(0)                                                        PDEF
0                                       shfx  f:(2,2),f:(2,2),x0,60                                                      PDEF
0                                       iorx  f:(2,3),f:(2,2)                                                            PDEF
0                                     dend                                                                               PDEF
0                                     do      sn:(f:(2,1))/=sn:(0)                                                       PDEF
0                                       shfx  f:(2,2),f:(2,2),x0,28                                                      PDEF
0                                       iorx  f:(2,2),f:(2,1)                                                            PDEF
0                                       shfx  f:(2,2),f:(2,2),x0,32                                                      PDEF
0                                       iorx  f:(2,3),f:(2,2)                                                            PDEF
0                                     dend                                                                               PDEF
0                                     lx      f:(2,2),f:(2,4),8                                                          PDEF
0                                     isob    f:(2,2),f:(2,2),x0,7007(8)    .WARNING - <tracesiz> dependent.            PDEF
0                                     incx    f:(2,2),1                                                                  PDEF
0                                     sx      f:(2,2),f:(2,4),8                                                          PDEF
0                                     sxi     f:(2,3),f:(2,4),f:(2,2),8                                                  PDEF
0                          t2         bss     0                                                                          PDEF
0                                     pend                                                                               PDEF
```



```
OFFSET BIT   LINE BINARY   SOURCE STATEMENT                                                        CPU ASSEMBLY          LEVEL
0                          ...........................................................................
0                          . ERRSTOP - This macro generates a call to the error stop routine to
0                          .           terminate 180 operation after an unrecoverable error has
0                          .           occurred.
0                          .
0                          .         errstop p1
0                          .              where p1 = label on a string that defines the error halt message
0                          ...........................................................................
0                          .
0                                     PROC                                                                               PDEF
0                          errstop    pname                                                                              PDEF
0                          f:(0)      addaq   a0,a0,16                                                                   PDEF
0                                     sa      af,a0,-16                                                                  PDEF
0                                     addaq   af,a_root,f:(2,0)                                                          PDEF
0                                     cpyax   x0,af                                                                      PDEF
0                                     shfx    x0,x0,x0,16                                                                PDEF
0                                     addxq   x0,x0,31                                                                   PDEF
0                                     sx      x0,a0,-8                                                                   PDEF
0                                     ente    x0,00ff(16)                                                                PDEF
0                                     addaq   af,a0,-8                                                                   PDEF
0                                     callseg bs_errst,a_bindin,af                                                       PDEF
0                                     la      af,a0,-16                                                                  PDEF
0                                     addaq   a0,a0,-16                                                                  PDEF
0                                     PEND                                                                               PDEF
```

```
OFFSET BIT   BINARY            SOURCE STATEMENT                                             CPU ASSEMBLY      LEVEL
   o                           ..........................................................................
   o                           .  Define A and X register usage
   o                           .    Note...
   o                           .        1. X0, X1, X2, XD, XE and XF are scratch registers
   o                           .        2. AE and AF are scratch registers
   o                           .        3. AF contains a pointer to 'xpinitv' at deadstart time. It's
   o                           .           used only for system initialization
   o                           ..........................................................................
   o                           .
   o                           a_root   areg    4                .Pointer to beginning of mainframe wired.
   o                           .                                 .  (If not A4, MXP must be changed.)
   o                           a_cst    areg    5                .Pointer to CST.
   o                           a_xcb    areg    6                .Pointer to XCB of current task. NIL if idle.
   o                           a_dscb   areg    7                .NOS170 DSCB.
   o                           .
   o                           x_mcr    xreg    3                .Scratch reg for MCR
   o                           x_clock  xreg    4                .Contains PIT/FRC values.
   o                           x_kef    xreg    5                .Contains KEF while processing traps and RUNNOS. ...

   o                           x_resume xreg    8                .Contains RESUME flag while in IDLE 180.
   o                           .
   o                           .
   o                           .  Equates for RUNNOS routine.
   o                           .
   o                           a_innosx areg    8                .Pointer to NOS XCB.
   o                           a_inret  areg    9                .Return address.
   o                           .
   o                           x_inmcr  xreg    6                .NOS170 MCR.
   o                           x_infrc  xreg    7                .Save for free running clock.
   o                           .
   o                           .
   o                           .  Equates for RQPROC routine.
   o                           .
   o                           a_rqtbl  areg    10               .Contains the pointer to REQTBL entry
   o                           .
   o                           a_rq_ret areg    11               .Return from RQPROC routine.
   o                           a_extret areg    12               .Return from EXTINT routine.
   o                           a_sitret areg    13               .Return address for PRSIT routine.
   o                           .
   o                           ..
   o                           .  Equates for REGISTER SAVE values (X0 for CALLSEG instructions)
   o                           .
   o         00000000000000C7  x_envirl equ     00c7(16)         .Environment for CALL.
```

1000 xxxx MCR [Δt]

0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890


```
OFFSET BIT   BINARY            SOURCE STATEMENT                                             CPU ASSEMBLY      LEVEL
   o                           ..........................................................................
   o                           .  DEFCST  - This PROC is used to define and initialize the CST.
   o                           .           (See deck that defines equates to see definition of fields).
   o                           ..........................................................................
   o                           .
   o                                    PROC                                                              PDEF
   o                           defcst   pname                                                             PDEF
   o                           f:(0,0)  bss     0                                                         PDEF
   o                           lpidz    set     0                                                         PDEF
   o                                    while   lpidz<f:(2,0)                                             PDEF
   o                                    local   cst1                                                      PDEF
   o                           cst1     bssz    cstsize                                                   PDEF
   o                                                                                                      PDEF
   o                                    org     cst1+memport            .Best guess for memory port mask. PDEF
   o                                    vfd,8   1**(lpidz*2)            .  (may be changed in BEGIN).     PDEF
   o                                    org     cst1+lpid                                                 PDEF
   o                                    vfd,8   lpidz                                                     PDEF
   o                                    org     cst1+lpid8                                                PDEF
   o                                    vfd,8   lpidz*8                                                   PDEF
   o                                    org     cst1+cpu_stat                                             PDEF
   o                                    vfd,8   2                                                         PDEF
   o                                    org     cst1+tracect1                                             PDEF
   o                                    vfd,16  0                                                         PDEF
   o                                    address r,trace+lpidz*(tracesiz+2)*8                              PDEF
   o                                    org     cst1+taskid                                               PDEF
   o                                    vfd,16,8 1,1                                                      PDEF
   o                                    org     cst1+prior180      .Initial 180 priority and 170 equivalent). PDEF
   o                                    vfd,24  070308(16)                                                PDEF
   o                                    org     cst1+jcbp                                                 PDEF
   o                                    vfd,4,12,32 1,mstlen,0                                            PDEF
   o                                    org     cst1+cp_state+cp_curst                                    PDEF
   o                                    vfd,8   running                                                   PDEF
   o                                    org     cst1+cp_state+cp_nxtst                                    PDEF
   o                                    vfd,8   running                                                   PDEF
   o                                    org     cst1+xcbp                                                 PDEF
   o                                    vfd,48  0ffff80000000(16)                                         PDEF
   o                                    org     cst1+cptime                                               PDEF
   o                                    vfd,64  0ffffffffffff(16)                                         PDEF
   o                                    org     cst1+jtime                                                PDEF
   o                                    vfd,64  07fffffff(16)                                             PDEF
   o                                    org     cst1+cachtim                                              PDEF
   o                                    vfd,64  07fffffffffffffff(16)                                     PDEF
   o                                    org     cst1+maptim                                               PDEF
   o                                    vfd,64  07fffffffffffffff(16)                                     PDEF
   o                                    org     cst1+ijlep                                                PDEF
   o                                    vfd,48  0ffff80000000(16)                                         PDEF
   o                                    org     cst1+dpint                                                PDEF
   o                                    vfd,64  07fffffffffffffff(16)                                     PDEF
   o                                    org     cst1+idlstats+idle_cnt                                    PDEF
   o                                    vfd,56  1                         . initialize the cpu idle count PDEF
   o                                    org     cst1+cstsize                                              PDEF
   o                           lpidz    set     lpidz+1                                                   PDEF
   o                                    dend                                                              PDEF
   o                                    pend                                                              PDEF
```

0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890

```
OFFSET BIT  BINARY            SOURCE STATEMENT                                      CPU ASSEMBLY            LEVEL
   0                          ......................................................................
   0                          .                         MAINFRAME WIRED
   0                          .   Define oss$mainframe_wired data.
   0                          .   This data must be at the beginning of the Mainframe-wired segment.
   0                          .      !!!! THIS DATA MUST START AT BYTE 0 OF SEGMENT 1 !!!!
   0                          .
   0                          ......................................................................
   0                          oss$mainframe_wired   SECTION working,read+write
   0                                  USE     oss$mainframe_wired
   0                                  def     root
   0        0000000000000000  root    vfd,64  0
   8        0000000000000000          vfd,64  0
   0                          .
   0                          ...............................................................
   0                          .
   0                          .       NOS/VE memory limits.  Defines the upper and lower bounds of NOS/VE
   0                          .       memory, the bounds are RMAs.  During deadstart the memory upper bound
   0                          .       is determined by the size of the memory image.
   0                          .
   0                          .       NOTE:  The memlimit variable is referenced from Cybil, definition is
   0                          .       defined by the variable 'osv$180_memory_limits'.
   0                          .
   0                          ...............................................................
  10        00000000          memlimit vfd,32 0                        .Lower bound.
  14        00000000                  vfd,32  0                        .Upper bound during deadstart.
  18        00000000                  vfd,32  0                        .Upper bound after system initialized.
  1C        00000000                  vfd,32  0                        .?????
   0                          .
  20        0000000000000190  scb     bss     scbsize                  .SCB communication area.
   0        000000000000002B  scbvec  equ     scb+scbvecsim            .Vector simulation option.
   0                          .
 1B0        0000000000000000  mtv$idle_message_line bss 0              .Message written to line 1 of console
 1B0        0001                      vfd,8,8 0,1                      .  y position on console
 1B2        0000                      vfd,8,8 0,0                      .  length
 1B4        00000000                  vfd,32  0                        .  rma field
 1B8        0000000000000050          bss     80                       .  text of message
 208        0000000000000006          bss     6                        .  space for pointer
   0                          .
 220                                  align   0,32
   0                          cst0    defcst  maxcst                   .Define CPU STATE TABLE (CST).
   0                          .
 440        00                os_type vfd,8   0                        .Operating mode (standalone, NOS, or NOSBE)
 441        00                os_terms vfd,8  0                        .170 termination status (0=running,
 442        000000000000              vfd,48  0                        .  1=mode error, 2=fatal due)
 448        0000000000000000  kcb_rma vfd,64  0                        .RMA pointer to keypoint buffer
 450        01                manddlst vfd,8  1                        .TRUE if dualstate is mandatory at this site.
 451        00                cpusposs vfd,8  0                        .TRUE if multiple cpus could EVER
   0                                                                   . be run on this mainframe
 458                                  align   0,8
 458        0000000000000000  multpro vfd,64  0                        .Non-zero if more than one cpu is running.
 460        0000000000000000  nosjps  vfd,64  0                        .JPS of NOS170 if Dual State active.
 468        0000000000000000  nosexit vfd,64  0                        .Time of last exit from NOS170.
 470        100300000000      nostab  vfd,48  0100300000000(16)        .If dual state, contains PVA of
   0                          .                                         NOS table containing priorities, etc.

            0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT  BINARY            SOURCE STATEMENT                                      CPU ASSEMBLY            LEVEL
 476                          nosxp   address r,a170_xp                .If dual state, contains PVA of NOS XP
 47C                          nossegt address r,a170_st                .Pointer to NOS segment table ..
 482        0000009800000000          vfd,32,32,32  a170_stl*8,0,8     .  ... rest of adaptable pointer to seg table.
 482        00000008
   0                          .
 492                          frc_p   address p,xfrc_p                 .Pointer to free running clock time for
   0                          .                                        dispatcher to run.
   0                          .                                         180 idle routine too early)
 498        0100              mlist   vfd,16  00100(16)                .Memory_link_status.
 4A0                                  align   0,8
 4A0        000002E300002081  ve_vrsn vfd,32,14,6,6,6  ost$psr,0,ost$nve,if_versn,if_level  .PSR lvl, OS type,
   0                                                                   .   i/f version and level. This field is set
   0                                                                   .  by LINOS but may be changed by the CHAOSV
   0                                                                   .  command.
 4A8        000FFFFFFFFFFFFF  eiflag  vfd,64  0fffffffffffff(16)       .EXTERNAL INTERRUPT flag. Contains FRC value
   0                                                                   .  of when to poll for next IO completion.
   0                                                                   .  If a IOU sends an external interrupt, the
   0                                                                   .  value of this word is set to one.
 4B0        00000000000F4240  eiinc   vfd,64  1000000                  .Rate to poll for lost external interrupts.
   0                                                                   .  Polling is immediate if EXT INT received
   0                                                                   .  and EIFLAG <> 0.
   0                                                                   .  NOTE: because of the algorithm used,
   0                                                                   .  asyninc must not be larger than this number.
 4B8        0000000000000000  asyntime vfd,64 0                        .FRC time to next check async activities.
 4C0        0000000000030D40  asyninc vfd,64  200000                   .Rate at which asyn activities are checked.
 4C8        000000000000C350  sitvalue vfd,64 50000                    .Default SIT value.
 4D0        000000000000246C  mstacklx vfd,64 mstackl                  .Length of monitor stack.
 4D8        0000000000000002  num_cst vfd,64  maxcst                   .Number of cst tables.
 4E0        0000000000000000  lockwait vfd,64,64 0,0                   .Total time/count waiting for dual CPU
 4E0        0000000000000000
   0                                                                   .  interlock.
 4F0        101400000100      sjmtrxcb vfd,4,12,32 1,mstlen,jrootsiz .Pointer to system job monitor execution
   0                          .                                         control block.
   0
   0                          .       Define interrupt ports for IOU external interrupts.  This is a mask with bit
   0                          .       7 being port 0, bit 6 being port 1, bit 5 being port 2, etc.  Currently all
   0                          .       non S0 machines interrupt on port 1 (value of 1) and the S0 interrupts on
   0                          .       port 2 (value of 4).  The value of this variable is set early in
   0                          .       initialization, it is set to the same value as memport.
   0
 4F6        01                intport vfd,8   1                        .Interrupt port mask for IOU external
   0                                                                   . interrupts.
 4F7        00                num_proc vfd,8  0                        .Number of processors physically configured.
 4F8        0708              mtrprior vfd,16 708(16)                  .Priority of 180 if control is
   0                                                                   . given to 170 via trap in 180 monitor.
 4FA        00                cpus_on vfd,8   0                        .Number of cpus logically on.
 500                                  align   0,8
 500        0000000000000020  osv_bl  bssz    32                       .osv$build_level
 520        0000000000000000  nostime vfd,64,64 0,0                    .Total time spent in NOS(total, ve_idle).
 520        0000000000000000
 530        0000FFFFFFFFFFFF  mmtime  vfd,64  0ffffffffffff(16)        .FRC time to next call Memory Manager.
 538        0000FFFFFFFFFFFF  swaptime vfd,64 0ffffffffffff(16)        .FRC time to next call job swapper.
 540        0000000000000000  scbtime vfd,64  0                        .FRC time to next check SCB status.
 548        00                haltring vfd,8  0                        .Halt CP on MCR fault <= this number.

            0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT    LINE BINARY              SOURCE STATEMENT                                    CPU ASSEMBLY                         LEVEL
   549        1297 00                  systemhr vfd,8    0                          .Same as above but for system job only.
   54A        1298 00                  asylock  vfd,8    0                          .Asynchronous interrupt lock.
   54B        1299 00                  asylocki vfd,8    0                          .Asynchronous interrupt lock for idle loop.
   54C        1300 00                  heap_tr  vfd,8    0                          .Enable_heap_trace system attribute.
   54D        1301 00                  heap_ver vfd,8    0                          .Verify_heap_linkage system attribute.
   54E        1302 00                  fltinj   vfd,8    0                          .Enable fault injection utility.
   54F        1303 FFFF80000000        mtvdftb  vfd,48   0ffff80000000(16)          .Pointer to DFT block control word
   555        1304 100300000000        nossegp  vfd,4,12,32 1,snnosmtr,0            .Pointer to NOS segment
   55B        1305                     mtrstp   address r,mst                       .Pointer to MTR SEG TABLE
   561        1306                     mtrxpp   address r,mxp                       .pointer to mtr xp
   567        1307                     dpv$scd_block_p address r,asciiblk
   56D        1308                     pextiou  address r,extiou
   573        1309                     pdpv$scd_time address r,dpv$scd_time
   579        1310 FFFF800000000000 endtbls  vfd,16,32,64 0ffff(16),080000000(16),0 .Pointer to mainframe wired heap.
   579        1311 000000000000
   588        1312                              align    0,8
   588        1313 0000000000000080 debug0   bssz     16*8                        .Array of debug values.
     0        1314                     .
     0        1315                     .
     0        1316                     .         The following is a cybil record.  Immediately after deadstart
     0        1317                     .         the NOS system time and date and the free running clock value
     0        1318                     .         are saved in this record.  During deadstart initialization these
     0        1319                     .         values are converted to NOS/VE base system time.
     0        1320                     .
   608        1321                              align    0,8
   608        1322 0B5B6EF6DBBDB6EF nos_tod  vfd,64   55333357333357333357(8) .NOS time of day (60 bits of display code)
   610        1323 0B637286E3A1C7AF nos_date vfd,64   55433450334350343657(8) .NOS date (60 bits of display code)
   618        1324 000000000000        cor_frc  vfd,48   0                          .Free running clock corresponding to 'nos_tod'
   61E        1325 0000000000000000 nosve_bt vfd,8,8,8,8,8,16,48 0,0,0,0,0,0,0    .NOS/VE base time (sec,min,hr,d,m,y,FRC)
   61E        1326 0000000000
     0        1327                     .
     0        1328                     .         Define symbols to reference NOS date and time in NOS's field length.
     0        1329                     .
     0        1330 0000000000000711 nostod   equu     3421(8)                     .NOS time of day address mask
     0        1331 0000000000000291 nosdate  equ      1221(8)                     .NOS date address mask
     0        1332                     .
     0        1333                     .        End of base system time record.
     0        1334                     .
     0        1335                     . ---------- Declaration code was omitted at compilation time ----------
     0        1336                     .
     0        1337                     .  Standalone deadstart data
     0        1338                     .
   62B        1339 100400000000        nossf    vfd,4,12,32    1,snsfmtr,0     .PVA of nos stack frame in mtr mode
     0        1340                              defg           osv$boot_sdte
     0        1341                              defg           dsv$ssr_sdte
     0        1342                              defg           osv$boot
     0        1343                              defg           osv$boot_is_executing
   63F        1344                              align          7,8
   63F        1345 00                  osv$boot       vfd,8     0
   640        1346 0000000000000000 osv$boot_sdte vfd,64    0
   648        1347 0000000000000000 dsv$ssr_sdte  vfd,64    0
   650        1348 00                  osv$boot_is_executing vfd,8 0
     0        1349                     .
   658        1350                              align    0,8

                 0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT    LINE BINARY              SOURCE STATEMENT                                    CPU ASSEMBLY                         LEVEL
     0        1351 0000000000000100 tracesiz equ      256                         .Number of trace entries per processor
     0        1352                                                                 . (must be power of 2.
     0        1353                                                                 . WARNING - TRACE macro must be changed
     0        1354                                                                 . if TRACESIZ is changed.
   658        1355 0000000000001020 trace    bssz     8*maxcst*(2+tracesiz) .Array to keep trace information
     0        1356                                                                 .  of what happens in monitor. See
     0        1357                                                                 .  the XTRACE macro.
  1678        1358 0000000000000810 dtrace   bssz     8*(tracesiz+2)              .Array for recording trace info while system
     0        1359                                                                 .  is stepped or idle.
  1E88        1360 00000000000001A0 xpinitv  bss      xpsize                      .Initial value for all job mode
     0        1361                     .                                             exchange packages.
  2028        1362 00000000000001A0 initmxp  bss      xpsize                      .initial value of mtr xp.
     0        1363                     .
     0        1364                     .
     0        1365                     .    Error mesasages displayed on error stop.
     0        1366                     .
  21C8        1367 48414C5445442056 csthalt    vfd,248    c'HALTED VIA CST REQUEST         '
  21C8        1368 4941204353542052
  21C8        1369 4551554553542020
  21C8        1370 20202020202020
  21E7        1371 5354455050454420 stepmes    vfd,248    c'STEPPED VIA CST REQUEST        '
  21E7        1372 5649412043535420
  21E7        1373 5245515545535420
  21E7        1374 20202020202020
  2206        1375 435055204641494C cpudown    vfd,248    c'CPU FAILED WITH INTERLOCK SET  '
  2206        1376 4544205749544820
  2206        1377 494E5445524C4F43
  2206        1378 4B205345542020

                 0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT  BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY           LEVEL
   0                          ..........................................................................
   0                          .  MONREQ  -  This proc is used to call a monitor request processor.
   0                          .                monreq   rc, ring, returnadr
   0                          .                        rc          - request code, either constant or x-register
   0                          .                        ring        - ring number for request validation. Zero implies
   0                          .                                      no checking. (rc must be constant if ring = 0)
   0                          .                        returnadr   - label to return to. If not supplied, returns
   0                          .                                      to next instruction.
   0                          .          A pointer to the beginning of the stack frame is passed as the parameter
   0                          .          list pointer. NOTE: most procedures called with this macro expect the
   0                          .          second parameter to be a pointer to the current CST.
   0                          .
   0                          ...........................................................................
   0                          .
   0                                   PROC                                                                    PDEF
   0                          monreq   pname                                                                   PDEF
   0                                   local    ex                                                             PDEF
   0                          f:(0)    bss      0                                                              PDEF
   0                                   do       sn:(f:(2,2))=0                                                 PDEF
   0                                     addpxq  a_rq_ret,x0,ex                                                PDEF
   0                                   dend                                                                    PDEF
   0                                   do       sn:(f:(2,2))/=0                                                PDEF
   0                                     addpxq  a_rq_ret,x0,f:(2,2)                                           PDEF
   0                                   dend                                                                    PDEF
   0                                   do       sn:(f:(2,1))=sn:(0)                                            PDEF
   0                                     addaq   a_rqtbl,a_root,reqtbl+rqtbles*f:(2,0)                         PDEF
   0                                     addaq   ae,a_bindin,16*f:(2,0)                                        PDEF
   0                                   dend                                                                    PDEF
   0                                   do       sn:(f:(2,1))/=sn:(0)                                           PDEF
   0                                     shfx    f:(2,0),f:(2,0),x0,4                                          PDEF
   0                                     cpyaa   ae,a_bindin                                                   PDEF
   0                                     addax   ae,f:(2,0)                                                    PDEF
   0                                     addaq   a_rqtbl,a_root,reqtbl                                         PDEF
   0                                     addax   a_rqtbl,f:(2,0)                                               PDEF
   0                                     shfx    f:(2,0),f:(2,0),x0,-1                                         PDEF
   0                                     addax   a_rqtbl,f:(2,0)                                               PDEF
   0                                     lbyts,1 f:(2,0),a_rqtbl,x0,rn                                         PDEF
   0                                     brxge   f:(2,0),f:(2,1),rqproc                                        PDEF
   0                                     addaq   a_rqtbl,a_root,reqtbl                                         PDEF
   0                                     addaq   ae,a_bindin,16*rqunim                                         PDEF
   0                                   dend                                                                    PDEF
   0                                   brxeq    x0,x0,rqproc                                                   PDEF
   0                          ex      bss      0                                                               PDEF
   0                                   pend                                                                    PDEF
```

0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890


```
OFFSET BIT  BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY           LEVEL
   0                          ..........................................................................
   0                          .  RQTABLE -  This macro generates monitor request table entries,
   0                          .             and binding section pointers. it also increments *MTRQMAX*
   0                          .             to indicate the maximum number of requests.
   0                          .               (see next page for definition of fields in macro)
   0                          ...........................................................................
   0                          .
   0                          .        Define offsets into a request table entry.
   0                          .
   0        0000000000000018 rqtbles  equ      3*8                    .Size of request table entry.
   0        0000000000000000 rn       equ      0                      .Highest RN for the request
   0        0000000000000001 il       equ      1                      .Interlock ordinal
   0        0000000000000002 rc       equ      2                      .Request code
   0        0000000000000008 totalt   equ      1*8                    .Total time for the request
   0        0000000000000010 rqcntmax equ      2*8                    .Word with both max and count.
   0                          .                                          (max time = left, count = right)
   0                                   proc                                                                    PDEF
   0                          rqtable  pname                                                                   PDEF
   0                                   org      reqtbl+rqtbles*f:(2,0)                                         PDEF
   0                                   vfd,8    f:(2,1)          .Highest ring number for the call             PDEF
   0                                   vfd,8    f:(2,2)          .Interlock ordinal                            PDEF
   0                                   vfd,8    f:(2,0)          .Request code                                 PDEF
   0                                   vfd,40   0                                                              PDEF
   0                                   bssz     24                                                             PDEF
   0                          .                                                                                PDEF
   0                                   do       f:(2,0)>mtrqmax                                                PDEF
   0                          mtrqmax  set      f:(2,0)                                                        PDEF
   0                                   dend                                                                    PDEF
   0                                   org      reqtbl+mtrqmax*rqtbles+rqtbles                                 PDEF
   0                                   use      binding                                                        PDEF
   0                                   do       sc:(f:(2,3))/=7                                                PDEF
   0                                   ref      f:(2,3)                                                        PDEF
   0                                   dend                                                                    PDEF
   0                                   address ce,f:(2,3)                                                      PDEF
   0                                   use      #lastsec                                                       PDEF
   0                                   pend                                                                    PDEF
   0                          .
   0                          .     Initialize maximum requests to 0.
   0                          .
   0        0000000000000000 mtrqmax  set      0
   0                          .
   0                          .     Define fwa of binding section and reqtable pointers.
   0                          .
   0                                   use      binding
   0                                   def      bindsec
   0        0000000000000000 bindsec  bss      0
   0        0000000000000000 bs_rqtbl bss      0
   0                                   use      #lastsec
   0                          .
```

0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

```
  OFFSET BIT        LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY          LEVEL
       0            1475                   ...........................................................................
       0            1476                   .                   MONITOR REQUEST TABLE
       0            1477                   .
       0            1478                   .     Each monitor request requires an entry in the following table
       0            1479                   .     Each entry is specified as follows:
       0            1480                   .          RQTABLE    NUM, HIGHRING, INTERLOCK_ORD, PROC
       0            1481                   .               NUM - Request code number
       0            1482                   .               HIGHRING - Highest ring number that can issue the request.
       0            1483                   .                      (0 = request restricted to monitor only).
       0            1484                   .               INTERLOCK_ORD - Specifies which interlock to use to serialize
       0            1485                   .                      monitor requests on a dual CPU machine.
       0            1486                   .                         (0 = no interlock)
       0            1487                   .               PROC - Name of procedure to call to process the request.
       0            1488                   .
       0            1489                   .     NOTE: When making an entry in this table, the following changes also
       0            1490                   .        have to be made:
       0            1491                   .            1) Define the request code in the deck SYC$MONITOR_REQUEST_CODES
       0            1492                   .            2) Add the request name to the table in CLM$DISPLAY_SYSTEM_DATA.
       0            1493                   .
       0            1494                   ...........................................................................
    2240            1495                           align   0,32
    2240            1496 0000000000000000 reqtbl   bssz    0
       0            1497                           rqtable 0,15,1,tmp$process_unknown_req_fault
       0            1518                           rqtable 1,13,0,tmp$cycle
       0            1539                           rqtable 2,13,0,tmp$delay
       0            1560                           rqtable 3,0,0,tmp$process_unknown_req_fault
       0            1581                           rqtable 4,1,1,iop$io_processor
       0            1602                           rqtable 5,13,1,mmp$advise_request_processor
       0            1623                           rqtable 6,13,1,mmp$advise_request_processor
       0            1644                           rqtable 7,13,1,mmp$advise_request_processor
       0            1665                           rqtable 8,2,1,tmp$create_task
       0            1686                           rqtable 9,0,1,pr_pf
       0            1707                           rqtable 10,2,1,tmp$create_job
       0            1728                           rqtable 11,2,1,tmp$exit_job
       0            1749                           rqtable 12,13,1,mmp$free_flush
       0            1770                           rqtable 13,13,1,mmp$free_flush
       0            1791                           rqtable 14,1,1,mmp$mtr_change_segment_table
       0            1812                           rqtable 15,0,0,tmp$process_unknown_req_fault   .FREE
       0            1833                           rqtable 16,0,0,tmp$process_unknown_req_fault   .FREE
       0            1854                           rqtable 17,0,0,tmp$process_unknown_req_fault   .FREE
       0            1875                           rqtable 18,3,1,jsp$mtr_job_swapping_requests
       0            1896                           rqtable 19,3,0,mtp$mtr_step_unstep_system
       0            1917                           rqtable 20,0,1,tmp$process_task_mcr_fault
       0            1938                           rqtable 21,15,1,tmp$mtr_process_system_error
       0            1959                           rqtable 22,3,0,tmp$fetch_task_statistics
       0            1980                           rqtable 23,0,0,tmp$process_unknown_req_fault   .FREE
       0            2001                           rqtable 24,0,0,tmp$process_unknown_req_fault   .FREE
       0            2022                           rqtable 25,13,0,tmp$mtr_ready_task
       0            2043                           rqtable 26,3,0,tmp$mtr_set_system_flag
       0            2064                           rqtable 27,15,1,tmp$mtr_wait
       0            2085                           rqtable 28,1,1,mmp$mtr_lock_ring_1_stack
       0            2106                           rqtable 29,3,1,tmp$mtr_send_signal
       0            2127                           rqtable 30,1,1,mmp$mtr_set_get_segment_length
       0            2148                           rqtable 31,6,1,mmp$mtr_read_write_io

                         0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
  OFFSET BIT        LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY          LEVEL
       0            2169                           rqtable 32,3,1,tmp$job_recovery_requests
       0            2190                           rqtable 33,1,1,mmp$mtr_ring1_segment_request
       0            2211                           rqtable 34,2,1,tmp$task_exit
       0            2232                           rqtable 35,0,0,tmp$process_unknown_req_fault   .FREE
       0            2253                           rqtable 36,3,1,tmp$mtr_update_job_task_enviro
       0            2274                           rqtable 37,0,0,tmp$process_unknown_req_fault   .FREE
       0            2295                           rqtable 38,13,1,mmp$mtr_lock_unlock_pages
       0            2316                           rqtable 39,13,1,mmp$mtr_lock_unlock_pages
       0            2337                           rqtable 40,13,1,mmp$mtr_fetch_pva_unwritten_pgs
       0            2358                           rqtable 41,1,1,dmp$mtr_allocate_front_end
       0            2379                           rqtable 42,1,1,dmp$mtr_deallocate_front_end
       0            2400                           rqtable 43,1,1,dmp$apply_mat_changes
       0            2421                           rqtable 44,1,1,iop$tape_queue_request
       0            2442                           rqtable 45,1,1,iop$translate_byte_address
       0            2463                           rqtable 46,3,1,cmp$monitor_routines
       0            2484                           rqtable 47,3,1,tmp$mtr_ready_system_task
       0            2505                           rqtable 48,13,1,mmp$mtr_lock_unlock_segment
       0            2526                           rqtable 49,3,1,dsp$issue_dft_request
       0            2547                           rqtable 50,13,1,mmp$mtr_wait_io_completion
       0            2568                           rqtable 51,0,0,tmp$switch_task
       0            2589                           rqtable 52,0,0,mtp$process_short_warning
       0            2610                           rqtable 53,0,0,mtp$monitor_system_status
       0            2631                           rqtable 54,0,1,iop$process_io_completions
       0            2652                           rqtable 55,3,0,dpp$display_request
       0            2673                           rqtable 56,0,0,dpp$process_scd_block
       0            2694                           rqtable 57,3,1,osp$process_job_keypoint_req
       0            2715                           rqtable 58,0,1,mmp$periodic_call
       0            2736                           rqtable 59,0,0,mtp$process_due
       0            2757                           rqtable 60,0,0,tmp$process_unknown_req_fault   .FREE
       0            2778                           rqtable 61,0,1,jsp$swap_polling
       0            2799                           rqtable 62,0,0,mtp$process_170_mtr_requests
       0            2820                           rqtable 63,0,0,tmp$process_unknown_req_fault   .FREE
       0            2841                           rqtable 64,1,1,iop$request_processor
       0            2862                           rqtable 65,3,0,dsp$access_logging_data
       0            2883                           rqtable 66,0,0,dsp$process_dft_entry
       0            2904                           rqtable 67,3,1,jmp$mtr_job_scheduler_requests
       0            2925                           rqtable 68,1,1,mmp$mtr_fetch_offset_mod_pages
       0            2946                           rqtable 69,13,1,mmp$process_assign_pages_req
       0            2967                           rqtable 70,13,1,mmp$free_flush
       0            2988                           rqtable 71,1,1,rfp$queue_data_fragments
       0            3009                           rqtable 72,3,1,nap$mtr_request_processor
       0            3030                           rqtable 73,3,1,dfp$mtr_file_server_request
       0            3051                           rqtable 74,6,1,mmp$process_move_pages_request
       0            3072                           rqtable 75,3,1,mmp$process_assign_contig_mem
       0            3093                           rqtable 76,1,1,dmp$mtr_reallocate_front_end
       0            3114                           rqtable 77,1,1,mmp$mtr_r1_server_seg_request
       0            3135                           rqtable 78,1,1,mtp$process_cpu_state_change
       0            3156                           rqtable 79,3,1,sfp$mtr_stats_facility_requests
       0            3177                           rqtable 80,3,1,dsp$mtr_manage_system_ds_status
       0            3198                           rqtable 81,3,1,jmp$update_serv_class_stats_req
       0            3219                           rqtable 82,0,0,tmp$process_unknown_req_fault   .FREE
       0            3240                           rqtable 83,0,0,tmp$process_unknown_req_fault   .FREE
       0            3261                           rqtable 84,0,0,tmp$process_unknown_req_fault   .FREE
       0            3282                           rqtable 85,13,0,syp$mtr_inject_hardware_fault

                         0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                                  CPU ASSEMBLY          LEVEL
      0           3304                    ..........................................................................
      0           3305                    .  Define the interlock array.  Initially only one lock word
      0           3306                    .  is used by the various request processors.
      0           3307                    .      One word per entry, word 0 not used.
      0           3308                    .           bit 0      0 = lock clear, 1 = lock set
      0           3309                    .           bit 32 - 63 = CST offset of CPU that has lock set.
      0           3310                    .
      0           3311                    . NOTE: If this record changes, be sure to make corresponding changes to the
      0           3312                    . CYBIL record declaration MTT$REQUEST_INTERLOCK_TABLE.
      0           3313                    ..........................................................................
      0           3314                    .
      0           3315                    .      Define offsets into interlock table.
      0           3316                    .
      0           3317 0000000000000006 maxilo    equ     6                  .Array size is 0..5
      0           3318 0000000000000008 ilsize    equ     8                  .Size of interlock table entry.
      0           3319 0000000000000000 ilflag    equ     0                  .Interlock flag
      0           3320 0000000000000002 lockcp    equ     2                  .^CST of locking cpu
      0           3321                    .
      0           3322                    .
      0           3323                    .
   2A50           3324                          align   0,8
   2A50           3325 0000000000000030 il_tbl    bssz    maxilo*ilsize      .Interlock array.
```



```
OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                                  CPU ASSEMBLY          LEVEL
      0           3327                    ..........................................................................
      0           3328                    .      Define  request codes for requests issued internally by monitor.
      0           3329                    .          These request codes must match the values defined in
      0           3330                    .          SYC$MONITOR_REQUEST_CODES.
      0           3331                    .        NOTE: only requests actually used by monitor are defined here.
      0           3332                    ..........................................................................
      0           3333                    .
      0           3334 0000000000000000 rqunim    equ     0                  .Unimplemented request code
      0           3335 0000000000000009 rqpf      equ     9                  .Code for PAGE FAULT.
      0           3336 0000000000000014 rqfault   equ     20                 .Code for MCR/UCR faults.
      0           3337 0000000000000033 tsksw     equ     51                 .task switch
      0           3338 0000000000000034 pswarn    equ     52                 .process short warning
      0           3339 0000000000000035 mon_smu   equ     53                 .monitor_smu_status
      0           3340 0000000000000036 proc_io   equ     54                 .process_io_completions
      0           3341 0000000000000038 ascii_kb  equ     56                 .process ascii keyboard
      0           3342 000000000000003A per_call  equ     58                 .periodic_call
      0           3343 000000000000003B proc_due  equ     59                 .process_due
      0           3344 000000000000003D swap_job  equ     61                 .poll job swapping
      0           3345 000000000000003E mm_ei     equ     62                 .process_170_mtr_requests
      0           3346 0000000000000042 proc_dft  equ     66                 .process DFT block
      0           3347 000000000000004E proc_cpu  equ     78                 .process_cpu_state_change
```

```
OFFSET BIT   LINE BINARY   SOURCE STATEMENT                                              CPU ASSEMBLY   LEVEL
                           ...........................................................................
                           . Define entry points into this module and the External names
                           . of the entry points
                           ...........................................................................
                                       defg    haltring,systemhr
                                       defg    mmtime,nostime
                                       def     swaptime,scbtime
                                       defg    sitvalue,fltinj
                                       def     dpv$scd_block_p
                                       def     mtv$idle_message_line
                                       def     dpv$scd_time
                                       defg    extiou
                                       def     mtrprior
                                       def     nosxp
                                       def     nosjps,nostab
                                       def     trace,dtrace
                                       def     nossegp
                                       defg    kcb_rma
                                       defg    nosve_bt,nos_tod,mlist
                                       defg    endtbls,memlimit,cst0
                                       defg    xpinitv,os_type,scb,scbvec
                                       def     os_terms
                                       defg    reqtbl
                                       def     il_tbl
                                       defg    multpro
                                       defg    manddlst
                                       defg    cpusposs
                                       def     eiflag
                                       def     lockwait
                                       def     sjmtrxcb
                                       defg    debug0
                                       defg    intport
                                       def     asyntime,asyntime
                                       defg    num_proc
                                       defg    initmxp
                                       defg    osv_bl
                                       def     idle,async,exchloop,rqproc,run_nos,extrq,traprtn
                                       def     int,nossegt
                                       defg    cpus_on
                                       defg    num_cst
                                       defg    mstacklx
                                       def     mtvdftb
                                       defg    heap_tr
                                       defg    heap_ver
                           heap_tr   ALIAS     syv$enable_heap_trace
                           heap_ver  ALIAS     syv$verify_heap_linkage
                           initmxp   ALIAS     OSV$INITIAL_MONITOR_XP
                           fltinj    ALIAS     syv$enable_fault_injection
                           mstacklx  ALIAS     osv$monitor_stack_length
                           num_cst   ALIAS     osv$maximum_cst_tables
                           num_proc  ALIAS     osv$cpus_physically_configured
                           cpus_on   ALIAS     osv$cpus_logically_on
                           lockwait  ALIAS     osv$monitor_interlock_wait_time
                           sjmtrxcb  ALIAS     mtv$system_job_monitor_xcb_p

                0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT   LINE BINARY   SOURCE STATEMENT                                              CPU ASSEMBLY   LEVEL
                           eiflag    ALIAS     osv$external_interrupt_time
                           intport   ALIAS     osv$external_interrupt_selector
                           asyntime  ALIAS     OSV$TIME_TO_CHECK_ASYN
                           asyninc   ALIAS     OSV$RATE_TO_CHECK_ASYN
                           manddlst  ALIAS     syv$mandatory_dualstate
                           cpusposs  ALIAS     osv$multiple_cpus_possible
                           multpro   ALIAS     OSV$MULTIPROCESSOR_RUNNING
                           debug0    ALIAS     osv$debug
                           mtrprior  ALIAS     OSV$MONITOR_PRIORITY
                           reqtbl    ALIAS     MTV$REQUEST_TABLE
                           il_tbl    ALIAS     mtv$request_interlock_table
                           xfrc_p    ALIAS     TMV$TIME_TO_CALL_DISPATCHER
                                     ref       xfrc_p
                           nosjps    ALIAS     MTV$NOS_JPS
                           sitvalue  ALIAS     OSV$DEFAULT_SIT_VALUE
                           os_type   ALIAS     OSV$170_OS_TYPE
                           os_terms  ALIAS     OSV$170_OS_TERMINATION_STATUS
                           nossegp   ALIAS     MTV$NOS_SEG_P
                           nostime   ALIAS     MTV$TOTAL_NOS_CPU_TIME
                           haltring  ALIAS     MTV$HALT_CPU_RING_NUMBER
                           systemhr  ALIAS     MTV$SYSTEM_HALTRING
                           extiou    ALIAS     OSV$IOU_EXTERNAL_INTERRUPT
                           mtvdftb   ALIAS     mtv$dft_block_p
                           scb       ALIAS     MTV$SCB
                           scbvec    ALIAS     MTV$SCB_VECTOR_SIM_ATTRIBUTE
                           nosxp     ALIAS     MTV$NS_XP_P
                           nostab    ALIAS     MTV$NST_P
                           nossegt   ALIAS     MTV$NOS_SEGMENT_TABLE_P
                           dtrace    ALIAS     MTV$DUMMY_TRACE_BUFFER
                           trace     ALIAS     MTV$TRACE_BUFFER
                           osv_bl    ALIAS     osv$build_level
                           ENDTBLS   ALIAS     OSV$MAINFRAME_WIRED_HEAP
                           memlimit  ALIAS     OSV$180_MEMORY_LIMITS
                           MMTIME    ALIAS     MMV$TIME_TO_CALL_MEM_MGR
                           scbtime   ALIAS     MTV$TIME_TO_CHECK_SCB_STATUS
                           CST0      ALIAS     MTV$CST0
                           XPINITV   ALIAS     MTV$XP_INITIAL_VALUE
                           kcb_rma   ALIAS     syv$pmf_cb_rm_word_address
                           ROOT      ALIAS     MTV$ROOT
                           BEGIN     ALIAS     MTP$BEGIN
                           BINDSEC   ALIAS     MTV$BINDING_SECTION
                           NOSVE_BT  ALIAS     OSV$BASE_SYSTEM_TIME
                           NOS_TOD   ALIAS     SYV$NOS_SYSTEM_TIME
                           MLIST     ALIAS     MTV$MLI_STATUS
                           SWAPTIME  ALIAS     JSV$TIME_TO_CALL_JOB_SWAPPER
                           .
                           .  The following are XDCLed so that the KEYPOINT analyzer can determine which
                           .  part of this module is executing when analyzing KEYPOINT files. If any changes
                           .  are made to these names or to the RELATIVE positions of the routines, the
                           .  KEYPOINT analyzer must be changed.
                           .
                           ASYNC     ALIAS     MTP$CHECK_ASYNC_ACTIVITY
                           IDLE      ALIAS     MTP$MONITOR_IDLE_LOOP
                           EXCHLOOP  ALIAS     MTP$PROCESS_JOB_EXCH_REQ

                0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT      LINE BINARY               SOURCE STATEMENT                                        CPU ASSEMBLY     LEVEL
     0          3457                      TRAPRTN  ALIAS   MTP$TRAP_HANDLER
     0          3458                      RQPROC   ALIAS   MTP$CALL_MONITOR_REQUEST
     0          3459                      RUN_NOS  ALIAS   MTP$RUN_NOS_170_MODE
     0          3460                      EXTRQ    ALIAS   MTP$PROCESS_EXTERNAL_INTERRUPT
```



```
OFFSET BIT      LINE BINARY               SOURCE STATEMENT                                        CPU ASSEMBLY     LEVEL
     0          3462                      ......................................................................
     0          3463                      .
     0          3464                      .   Define Stack Segment and exchange packages for monitor mode.
     0          3465                      .   The monitor exchange packages and segment table are located at
     0          3466                      .   the beginning of the monitor stack.
     0          3467                      .
     0          3468                      ......................................................................
     0          3469 000000000000246C mstackl  equu    xpsize+(mstlen+ajllen+1)*8+mstksize
     0          3470                  mts$monitor_stack SECTION extwork,read+write,,0,8,mstackl
     0          3471                           use     mts$monitor_stack
     0          3472                           def     mtrstak,mst
     0          3473                           defg    mxp
     0          3474 0000000000000000 bgnstak  bss     0
     0          3475 00000000000001A0 mxp      bssz    xpsize
   1A0          3476 00000000000008A0 mst      bssz    mstlen*8+ajllen*8+8
   A40          3477 0000000000001A2C mtrstak  bss     mstksize
  246C          3478 0000000000000000 mtrstake bss     0
     0          3479                           xpa     mxp,2,begin
     0          3490                           xpa     mxp,xptos,mtrstak,0
     0          3501                           xpareg  mxp,a_tos,mtrstak,mstkfram
     0          3512                           xpareg  mxp,a_csf,mtrstak,0
     0          3523                           xpareg  mxp,a_psa,nil
     0          3534                           xpareg  mxp,a_bindin,bindsec
     0          3545                           xpareg  mxp,a_root,root,0
     0          3556                           xpareg  mxp,5,nil
     0          3567                           xpareg  mxp,6,nil
     0          3578                           xpareg  mxp,7,nil
     0          3589                           xpareg  mxp,8,nil
     0          3600                           xpareg  mxp,9,nil
     0          3611                           xpareg  mxp,10,nil
     0          3622                           xpareg  mxp,11,nil
     0          3633                           xpareg  mxp,12,nil
     0          3644                           xpareg  mxp,13,nil
     0          3655                           xpareg  mxp,14,nil
     0          3666                           xpareg  mxp,15,nil
     0          3677                           xpv     mxp,xpstal,mst-bgnstak,16 .Segment table address
     0          3683                           xpv     mxp,xpstl,mstlen+ajllen,16 .Segment table length
     0          3689                           xpv     mxp,xpmm,m_mtrmsk,16  .Monitor mask
     0          3695                           xpv     mxp,xpum,m_usrmsk,16  .User mask
     0          3701                           xpv     mxp,xpkm,0,16
     0          3707                           xpv     mxp,xppit,0ffff(16),16  .Monitor PIT
     0          3713                           xpv     mxp,xppit+8,0ffff(16),16
     0          3719                           xpv     mxp,xplrn,1,16
     0          3725                           xpa     mxp,xptp,bs_trap,0
     0          3736                           xpv     mxp,xpflgte,00000(16),16
     0          3742                           xpv     mxp,xpbc2,cst0,16
     0          3748                           xpv     mxp,248,cst0,32  .Set offset and length of CST0 into XE
     0          3754                           xpv     mxp,252,cstsize,32
  246C          3760 000000000000246C          org     mtrstake
     0          3761                  mxp      alias   MTV$MONITOR_EXCHANGE_PACKAGE
     0          3762                  mst      alias   MTV$MONITOR_SEGMENT_TABLE
```

```
OFFSET BIT   LINE BINARY         SOURCE STATEMENT                                   CPU ASSEMBLY          LEVEL
     0                           ..........................................................................
     0                           .
     0                           .   Define Binding Section
     0                           .       (note - the RQTABLE macro puts entries here also)
     0                           ..........................................................................
     0                                   USE      BINDING
     0                           .
     0                                   def      bs_trap
     0                           BS_TRAP  ALIAS   MTV$TRAP_CBP
   560                           bs_trap  address ce,traprtn              .Used for monitor XP trap ptr.
   572                           bs_root  address p,root
     0                           .
     0                                   ref      MTP$ERROR_STOP
     0                                   ref      MTP$MTR_ERROR_STOP
     0                                   ref      OSP$PROCESS_MTR_PAGE_FAULT
     0                                   ref      TMV$PTL_LOCK
     0                           .
   578                           bs_errst address c,MTP$ERROR_STOP
   588                           bs_merrs address c,MTP$MTR_ERROR_STOP
   598                           bs_pgflt address ce,OSP$PROCESS_MTR_PAGE_FAULT
   5AA                           bs_ptlok address p,TMV$PTL_LOCK
```



```
OFFSET BIT   LINE BINARY         SOURCE STATEMENT                                   CPU ASSEMBLY          LEVEL
     0                           ..........................................................................
     0                           .
     0                           .   BOOT - Execution at deadstart starts here. Save a copy of the job
     0                           .          XP, reset the clock, and jump to the location that exchanges
     0                           .          to job mode.
     0                           ..........................................................................
     0                           .
     0                                   USE      CODE
     0                                   def      begin
     0            0000000000000000 begin bss      0                       .This is where execution begins
     0            8D000063               ente     x0,63(16)               .Initialize KBP register
     4            AC010410               isom     x1,x0,2020(8),x0        .NOT CORRECT FOR MULTIPROCESSOR
     8            0F01                   cpyxs    x1,x0                   .Need to do in every processor
     A            8D000047               ente     x0,r_bc
     E            0E01                   cpysx    x1,x0                   .Get  base constant.
    10            0B42                   cpyax    x2,a_root
    12            2421                   addx     x1,x2                   .Form pointer to cst
    14            0A15                   cpyxa    a_cst,x1
    16            8515000A               sa       a_cst,a_csf,10          .Save CST_P for p-list.
    1A            3F10                   entl     x0,r_eid                .Save EID in CST.
    1C            0E00                   cpysx    x0,x0
    1E            8350000E               sx       x0,a_cst,elem_id
    22            3D00                   entp     x0,0                    .Start cache and map purging
    24            8350000B               sx       x0,a_cst,cachtim
    28            8350000C               sx       x0,a_cst,maptim
    2C            8241008D               lx       x1,a_root,nosexit       .Check if this is first CPU.
    30            951000CC               brxne    x1,x0,begin5            .Jump if not first CPU.
     0                           .
     0                           .   The following is initialization code executed ONLY on the first CPU to start.
     0                           .
    34            834D00C8               sx       xd,a_root,osv$boot_sdte
    38            8D011014               ente     x1,1000(16)+mstlen      .Set up pointer to system jobmonitor
    3C            A9110020               shfx     x1,x1,x0,32             .  XCB.
    40            8B110100               addxq    x1,x1,jr_mxcb
    44            0A16                   cpyxa    a_xcb,x1
    46            3D11                   entp     x1,1
    48            D841063F               sbyts,1  x1,a_root,x0,osv$boot
     0                           .
    4C            84470470               la       a_dscb,a_root,nostab    . Build pointer to the dscb
    50            2AF7                   addax    a_dscb,xf
     0                           .
    52            844E055B               la       ae,a_root,mtrstp        . Update NOS st from mtr st
    56            844F047C               la       af,a_root,nossegt
    5A            D7E10028               lbyts,8  x1,ae,x0,snnthmtr*8
    5E            DFF10028               sbyts,8  x1,af,x0,snnth170*8
    62            D7E10020               lbyts,8  x1,ae,x0,snsfmtr*8
    66            DFF10020               sbyts,8  x1,af,x0,snsf170*8
    6A            DF410648               sbyts,8  x1,a_root,x0,dsv$ssr_sdte  . Tell job mode the STE of SSR
    6E            D7E10090               lbyts,8  x1,ae,x0,12(16)*8
    72            DFF10090               sbyts,8  x1,af,x0,12(16)*8
     0                           .
    76            85560020               sa       a_xcb,a_cst,xcbp        .Store xcb pointer in CST.
    7A            1661                   tpage    x1,a_xcb                .Save RMA of XCB in CST.
    7C            83510005               sx       x1,a_cst,xcbrma
```

```
OFFSET BIT   LINE BINARY            SOURCE STATEMENT                                   CPU ASSEMBLY          LEVEL
   80        3840 3F61                         ent1     x0,r_jps               .Update JPS.
   82        3841 0F01                         cpyxs    x1,x0
   84        3842 1671                         tpage    x1,a7                  .Store MPS into CST.
   86        3843 DB510068                     sbyts,4  x1,a_cst,x0,mps
   8A        3844 844D0470                     la       ad,a_root,nostab       .FWA of NOS field length.
   8E        3845 85470470                     sa       a_dscb,a_root,nostab   .Save dscb pointer
   92        3846 0811                         cpytx    x1,x1                  .Reset time task began execution.
   94        3847 83410007                     sx       x1,a_root,scb+scbnsrv
   98        3848 3F10                         ent1     x0,r_eid
   9A        3849 8341008D                     sx       x1,a_root,nosexit      .Set time when last exited NOS
   9E        3850 0E0E                         cpysx    xe,x0                  .Save element id.
   A0        3851 3D56                         entp     x6,5                   .High order 4 bits of S0 model number.
   A2        3852 ADEE0A03                     isob     xe,xe,x0,(40*64+3)     .High order 4 bits of model number from
    0        3853                                                              . element id.
    0        3854
    0        3855                   .  Set up memory and interrupt port mask based on processor.
    0        3856
   A6        3857 D0510004                     lbyts,1  x1,a_cst,x0,memport    .Memory and interrupt port mask for non S0
   AA        3858 91E60007                     brrne    xe,x6,begin2_5         .Jump if not an S0.
   AE        3859 2831                         incr     x1,3                   .If S0, change port 0 (int sel = 1) to a 4,
   B0        3860 3D42                         entp     x2,4                   .  port 1 (int sel = 4) to an 8.
   B2        3861 94120003                     brxeq    x1,x2,begin2_5         .If cpu 0, then 4 is the right answer,
   B6        3862 3D81                         entp     x1,8                   .  otherwise 8 is the answer.
   B8        3863 0000000000000000 begin2_5 bss      0
   B8        3864 D8510004                     sbyts,1  x1,a_cst,x0,memport    .Set up port number mask for ext interrupts.
   BC        3865 D84104F6                     sbyts,1  x1,a_root,x0,intport
   C0        3866 824E0094                     lx       xe,a_root,ve_vrsn      .ve os type, dscb version/level
   C4        3867 837E000F                     sx       xe,a_dscb,d8ty         .Save in block
    0        3868                   .
    0        3869                   .     Set the NOS/VE memory limits.  Both upper bounds are set to the RMA of
    0        3870                   .     the SSR.  The deadstart upper bound may be reset before first page fault
    0        3871                   .     based on the image size.
    0        3872                   .
   C8        3873 82710008                     lx       x1,a_dscb,d7cm+8       .Fetch memory limits
   CC        3874 AD1E0417                     isob     xe,x1,x0,(64-48)*100(8)+24-1  .Isolate ve fwa DIV 10000(8)
   D0        3875 A9EE000C                     shfx     xe,xe,x0,12
   D4        3876 DB4E0010                     sbyts,4  xe,a_root,x0,memlimit
   D8        3877 844E062B                     la       ae,a_root,nossf               .Set upper bounds to the SSR RMA
   DC        3878 16EE                         tpage    xe,ae
   DE        3879 DB4E0018                     sbyts,4  xe,a_root,x0,memlimit+8       .Upperbound.
   E2        3880 DB4E0014                     sbyts,4  xe,a_root,x0,memlimit+4       .Upperbound during deadstart.
    0        3881                   .
    0        3882                   .    Fetch and store pointer to the DFT block
    0        3883                   .      r_pointer: offset, r_upper, r_lower, size
    0        3884                   .      rma of r_pointer = r_upper*10000000(8) + r_lower*1000(8) + offset*10(8)
    0        3885                   .
   E6        3886 D176010A                     lbyts,2  x6,a_dscb,x0,dscm+3*8+2 .Load r_upper into x6
   EA        3887 A9660015                     shfx     x6,x6,x0,7*3           .Shift: r_upper * 10000000(8)
   EE        3888 D17B010C                     lbyts,2  xb,a_dscb,x0,dscm+3*8+4 .Load r_lower into xb
   F2        3889 A9BB0009                     shfx     xb,xb,x0,3*3           .Shift: r_lower * 1000(8)
   F6        3890 24B6                         addx     x6,xb                  .Add r_lower to r_upper
   F8        3891 D17B0108                     lbyts,2  xb,a_dscb,x0,dscm+3*8+0 .Load offset into xb
   FC        3892 A9BB0003                     shfx     xb,xb,x0,1*3           .Shift: offset * 10(8)
  100        3893 24B6                         addx     x6,xb                  .Add offset to (r_upper + r_lower)

             0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789
```



```
OFFSET BIT   LINE BINARY            SOURCE STATEMENT                                   CPU ASSEMBLY          LEVEL
  102        3894 3D2B                         entp     xb,sn170mcb
  104        3895 8547054F                     sa       a_dscb,a_root,mtvdftb  .Save base ptr: ring and segment
  108        3896 D84B0550                     sbyts,1  xb,a_root,x0,mtvdftb+1 .Set cache bypass segment number for
    0        3897                                                              . DFT buffer.
  10C        3898 DB460551                     sbyts,4  x6,a_root,x0,mtvdftb+2 .Store dft offset in ptr
    0        3899                   .
  110        3900 3D00                         entp     x0,0
  112        3901 827E0000                     lx       xe,a_dscb,d7ty         .Determine STATE
  116        3902 ADE10B85                     isob     x1,xe,x0,5605(8)
  11A        3903 D8410440                     sbyts,1  x1,a_root,x0,os_type
  11E        3904 90100035                     brreq    x1,x0,begin4           .If not dualstate jump
    0        3905                   .
    0        3906                   .  Save NOS base system time and the corresponding value of the free running clock.
    0        3907                   .
  122        3908 ADE20711                     isob     x2,xe,x0,nostod        .Isolate time of day pointer
  126        3909 ADEE0291                     isob     xe,xe,x0,nosdate       .Isolate date pointer
  12A        3910 A2D22000                     lxi      x2,ad,x2,0             .Time of day (display code)
  12E        3911 A2DEE000                     lxi      xe,ad,xe,0             .Date (display code)
  132        3912 0801                         cpytx    x1,x0                  .Free running clock
  134        3913 834200C1                     sx       x2,a_root,nos_tod
  138        3914 834E00C2                     sx       xe,a_root,nos_date
  13C        3915 DD410618                     sbyts,6  x1,a_root,x0,cor_frc
  140        3916 844F0476                     la       af,a_root,nosxp
  144        3917 3D02                         entp     x2,0                   .Clear left half of nosjps
  146        3918 16F2                         tpage    x2,af
  148        3919 8342008C                     sx       x2,a_root,nosjps
  14C        3920 DB520014                     sbyts,4  x2,a_cst,x0,dualstat
  150        3921 844E047C                     la       ae,a_root,nossegt      .Store upper bits of nos seg table adr.
  154        3922 16E2                         tpage    x2,ae
  156        3923 D9F20118                     sbyts,2  x2,af,x0,xpstal
  15A        3924 A9220FF0                     shfx     x2,x2,x0,-16
  15E        3925 D9F20110                     sbyts,2  x2,af,x0,xpstau
  162        3926 844F055B                     la       af,a_root,mtrstp       .Set entry for MNFR WIRED SEG in NOS ST
  166        3927 0B41                         cpyax    x1,a_root              .a_root is mnfr wired segment
  168        3928 AD11050B                     isob     x1,x1,x0,2413(8)       .Isolate segment number
  16C        3929 A9110003                     shfx     x1,x1,x0,3             .make sdt number
  170        3930 D7F21000                     lbyts,8  x2,af,x1,0             .get sdt entry
  174        3931 DFE21000                     sbyts,8  x2,ae,x1,0             .set sdt entry in nos st
  178        3932 82710000                     lx       x1,a_dscb,d7ty         .determine STATE
  17C        3933 AD110B85                     isob     x1,x1,x0,5605(8)
  180        3934 D8410440                     sbyts,1  x1,a_root,x0,os_type
  184        3935 9F510002                     brcr     5,1,begin4             .Force EXCH bit
    0        3936
  188        3937 8E4F1E88          begin4     addaq    af,a_root,xpinitv
  18C        3938 766F09FF000009FF             movb,a_xcb,x0 af,x1 0,9,255,0 0,9,255,0
  18C        3939 0000
  196        3940 766F09A100FF09A1             movb,a_xcb,x0 af,x1 0,9,xpsize-255,255 0,9,xpsize-255,255
  196        3941 00FF
  1A0        3942 844E0561                     la       ae,a_root,mtrxpp       .move original xp to
  1A4        3943 8E4F2028                     addaq    af,a_root,initmxp      .mainframe wired.
  1A8        3944 76EF09FF000009FF             movb,ae,x0 af,x1 0,9,255,0 0,9,255,0
  1A8        3945 0000
  1B2        3946 76EF09A100FF09A1             movb,ae,x0 af,x1 0,9,xpsize-255,255 0,9,xpsize-255,255
  1B2        3947 00FF

             0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789
```

```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY                LEVEL
   1BC          3948 3F61                      entl    x0,r_jps             .Save current JPS in CST.
   1BE          3949 0E00                      cpysx   x0,x0
   1C0          3950 83500005                  sx      x0,a_cst,xcbrma
   1C4          3951 94000007                  brxeq   x0,x0,begin22
     0          3952                   .
     0          3953                   .    The following code is initialization code for all cpus EXCEPT the first.
     0          3954                   .
   1C8          3955 0000000000000000 begin5   bss     0
   1C8          3956 84470470                  la      a_dscb,a_root,nostab  .Pointer to interface block
   1CC          3957 3D11                      entp    x1,1
   1CE          3958 D8510030                  sbyts,1 x1,a_cst,x0,caldisp   .Call dispatcher.
     0          3959                   .
     0          3960                   .    Complete processor initialization for ALL processors.
     0          3961                   .
   1D2          3962 82410099          begin22 lx      x1,a_root,sitvalue    .Reset SIT.
   1D6          3963 3F62                      entl    x0,r_sit
   1D8          3964 0F01                      cpyxs   x1,x0
   1DA          3965 3D00                      entp    x0,0
   1DC          3966 D8500006                  sbyts,1 x0,a_cst,x0,cpu_stat  .Set cpu status running
   1E0          3967 3FC2                      entl    x0,r_te               .Enable traps
   1E2          3968 0F00                      cpyxs   x0,x0
```

0 1 2 3 4 5 6 7 12345678901234567890123456789012345678901234567890123456789012345678901234567890


```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY                LEVEL
     0          3970                   ..........................................................................
     0          3971                   .  Check Interrupt/Dispatch Flags - (TOP OF MAIN LOOP)
     0          3972                   .     Control comes here when the 'dispflag' in the CST is set. This code processes
     0          3973                   .     asynchronous conditions (such as IO completions and periodic conditions).
     0          3974                   .     The task switch routine is called if the 'call dispatcher' flag in the CST is    ...
                                       set.
     0          3975                   ..........................................................................
     0          3976                   .
   1E4          3977 0000000000000000 intdislp bss     0                     .Begin of interrupt-dispatch-loop.
   1E4          3978 3FC9                      entl    x0,r_pit              .Save monitor clock.
   1E6          3979 0E04                      cpysx   x_clock,x0
     0          3980                   .
     0          3981                   .  Process asynchronous interrupts (EXT INT, Console input, Memory manager,
     0          3982                   .           Job swapper, etc.)
     0          3983                   .
   1E8          3984 3F00              async   entl    x0,0                  .Check if time to check async
   1EA          3985 0802                      cpytx   x2,x0                 .  activities.
   1EC          3986 82410097                  lx      x1,a_root,asyntime    .Get time of next async activity.
   1F0          3987 D8500034                  sbyts,1 x0,a_cst,x0,asyncp
   1F4          3988 83520001                  sx      x2,a_cst,cpwell       .Update cpu alive flag.
   1F8          3989 97120075                  brxge   x1,x2,tswit           .Jump if not time for async activity.
   1FC          3990 8E4E054A                  addaq   ae,a_root,asylock
   200          3991 14E1                      lbset   x1,ae,x0              .Test and set lock
   202          3992 92100070                  brrgt   x1,x0,tswit           .Jump if another processor is already
     0          3993                   .                                       processing asynchronous work.
   206          3994 82410098                  lx      x1,a_root,asyninc     .Update time to next check async.
   20A          3995 824E0095                  lx      xe,a_root,eiflag      .Fetch ext interrupt flag.
   20E          3996 83420007                  sx      x2,a_root,scb+scbnsrv .Update '180 alive' flag.
   212          3997 2421                      addx    x1,x2
   214          3998 83410097                  sx      x1,a_root,asyntime
   218          3999 96E20016                  brxgt   xe,x2,async6          .Jump if no ext interrupts to process.
   21C          4000 82410096                  lx      x1,a_root,eiinc
   220          4001 844E056D                  la      ae,a_root,pextiou
   224          4002 3F00                      entl    x0,0
   226          4003 2421                      addx    x1,x2
   228          4004 83410095                  sx      x1,a_root,eiflag
   22C          4005 83E00000                  sx      x0,ae,0
     0          4006                           monreq  proc_io
   240          4037 3D02                      entp    x2,0
   242          4038 0822                      cpytx   x2,x2
     0          4039                   .
   244          4040 844E0573          async6  la      ae,a_root,pdpv$scd_time
   248          4041 82E10000                  lx      x1,ae,0               .Test if time to call keyboard rtn.
   24C          4042 9612000C                  brxgt   x1,x2,async8          .Jump if not time
     0          4043                           monreq  ascii_kb
   260          4074 3D02                      entp    x2,0
   262          4075 0822                      cpytx   x2,x2
     0          4076                   .
   264          4077 844E054F          async8  la      ae,a_root,mtvdftb     .Fetch pointer to DFT block.
   268          4078 82E10000                  lx      x1,ae,dftcw           .Get DFT control word.
   26C          4079 A911003E                  shfx    x1,x1,x0,62           .Check E8 field.
   270          4080 9710000C                  brxge   x1,x0,async12         .Jump if not set.
     0          4081                           monreq  proc_dft              .NOTE!! May exit with E8 still set.
   284          4112 3D02                      entp    x2,0                  .  If so, recall in a few hundred
```

0 1 2 3 4 5 6 7 12345678901234567890123456789012345678901234567890123456789012345678901234567890

```
OFFSET BIT      LINE BINARY          SOURCE STATEMENT                                        CPU ASSEMBLY                LEVEL
   286          4113 0822                    cpytx   x2,x2                .  milliseconds.
     0          4114                 .
   288          4115 824100A8        async12 lx       x1,a_root,scbtime    .Check if time to look at SCB status.
   28C          4116 9712000C                brxge   x1,x2,async15        .Jump if SCB check not required.
     0          4117                         monreq  mon_smu
   2A0          4148 3D02                    entp    x2,0
   2A2          4149 0822                    cpytx   x2,x2
     0          4150                 .
   2A4          4151 824100A7        async15 lx       x1,a_root,swaptime   .Check if time to call job swapper.
   2A8          4152 9712000C                brxge   x1,x2,async20        .Jump if job swapper call not needed.
     0          4153                         monreq  swap_job
   2BC          4184 3D02                    entp    x2,0
   2BE          4185 0822                    cpytx   x2,x2
     0          4186                 .
   2C0          4187 824100A6        async20 lx       x1,a_root,mmtime     .Check if time to call Mem Mgr.
   2C4          4188 9712000A                brxge   x1,x2,async50        .Jump if Mem Mgr call not needed.
     0          4189                         monreq  per_call
     0          4220                 .
   2D8          4221 3F00            async50 entl     x0,0
   2DA          4222 D840054A                sbyts,1 x0,a_root,x0,asylock .Clear lock
   2DE          4223 9000FF85                brreq   x0,x0,async          .Check for more work before exiting
     0          4224                 .
     0          4225                 .  Call the task switch routine if necessary.
     0          4226                 .     (NOTE - the following is similar to MONREQ/RQPROC but is inline for
     0          4227                 .              performance.)
     0          4228                 .
   2E2          4229 82510006        tswit   lx       x1,a_cst,discntl     .Check if task switch required.
   2E6          4230 84560020                la      a_xcb,a_cst,xcbp     .XCB will be NIL if task exited!
   2EA          4231 A9110FE0                shfx    x1,x1,x0,-32
   2EE          4232 0B6E                    cpyax   xe,a_xcb             .XE must have XCB adr if branch to tsckpr.
   2F0          4233 9410006B                brxeq   x1,x0,tsckpr         .Jump if task switch not needed.
   2F4          4234 920E0014                brrgt   x0,xe,tswit4         .Jump if NIL XCB (processor idle).
   2F8          4235 D1610058                lbyts,2 x1,a_xcb,x0,xppit    .Calculate JOB MODE time
   2FC          4236 D1620060                lbyts,2 x2,a_xcb,x0,xppit+8
   300          4237 825F0008                lx      xf,a_cst,jtime
   304          4238 A9110010                shfx    x1,x1,x0,16
   308          4239 2412                    addx    x2,x1
   30A          4240 1F22                    ents    x2                   .Sign extend job mode time
   30C          4241 252F                    subx    xf,x2
   30E          4242 835F0008                sx      xf,a_cst,jtime
   312          4243 AC01081F                isom    x1,x0,4037(8)        .Save monitor mode time in CST.
   316          4244 2541                    subx    x1,x_clock
   318          4245 83510009                sx      x1,a_cst,mtime
   31C          4246 8E4A2708        tswit4  addaq   a_rqtbl,a_root,reqtbl+rqtbles*tsksw
   320          4247 8E3E0330                addaq   ae,a_bindin,16*tsksw
   324          4248 3FC9                    entl    x0,r_pit
   326          4249 0E02                    cpysx   x2,x0                .Get current PIT
   328          4250 091F                    cpyaa   af,a_csf
   32A          4251 8D0000C7                ente    x0,x_envir1          .Process the request
   32E          4252 B5EF0000                callseg bs_rqtbl,ae,af
   332          4253 82AE0001                lx      xe,a_rqtbl,totalt    .Update total and max time
   336          4254 82AD0002                lx      xd,a_rqtbl,rqcntmax
   33A          4255 3FC9                    entl    x0,r_pit
   33C          4256 0E0F                    cpysx   xf,x0                .Calculate time to process the request

                     0 1 2 3 4 5 6 7  12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

6' 330


```
OFFSET BIT      LINE BINARY          SOURCE STATEMENT                                        CPU ASSEMBLY                LEVEL
   33E          4257 25F2                    subx    x2,xf
   340          4258 242E                    addx    xe,x2
   342          4259 83AE0001                sx      xe,a_rqtbl,totalt
   346          4260 281D                    incr    xd,1
   348          4261 A8DE0020                shfc    xe,xd,x0,32          .Check if new maximum time.
   34C          4262 93E20005                brrge   xe,x2,tswit5         .  Jump if not new max.
   350          4263 0C2E                    cpyrr   xe,x2
   352          4264 A8ED0020                shfc    xd,xe,x0,32
   356          4265 82510007        tswit5  lx      x1,a_cst,cptime      .Get tasks timeslice.
   35A          4266 84560020                la      a_xcb,a_cst,xcbp     .Reload pointer to current XCB.
   35E          4267 83AD0002                sx      xd,a_rqtbl,rqcntmax
   362          4268 AC04081F                isom    x_clock,x0,4037(8)   .Reset monitor clock.
   366          4269 3F62                    entl    x0,r_sit             .Reset SIT.
   368          4270 0F01                    cpyxs   x1,x0                .Copy timeslice to SIT.
   36A          4271 3D01                    entp    x1,0                 .Reset CST fields -
   36C          4272 DB510030                sbyts,4 x1,a_cst,x0,discntl  .Clear task switch control flags.
   370          4273 83510009                sx      x1,a_cst,mtime       .  monitor mode time
   374          4274 0B6E                    cpyax   xe,a_xcb             .!! XCB adr must be in XE for TSCKPR.
   376          4275 920E0011                brrgt   x0,xe,tswit8         .Skip next part if XCB is NIL.
     0          4276                 .
   37A          4277 D1610058                lbyts,2 x1,a_xcb,x0,xppit    .Reset JOB MODE time
   37E          4278 D1620060                lbyts,2 x2,a_xcb,x0,xppit+8
   382          4279 A9110010                shfx    x1,x1,x0,16
   386          4280 2412                    addx    x2,x1
   388          4281 1F22                    ents    x2                   .Sign extend
   38A          4282 83520008                sx      x2,a_cst,jtime
   38E          4283 1661                    tpage   x1,a_xcb             .Save RMA of XCB in CST.
   390          4284 83510005                sx      x1,a_cst,xcbrma
   394          4285 3F61                    entl    x0,r_jps             .Update JPS.
   396          4286 0F01                    cpyxs   x1,x0
     0          4287                 tswit8  xtrace  5,x1,x2,xd,ae
     0          4315                 .
     0          4316                 .  Run NOS 170 if it has a priority greater than 180 has.
     0          4317                 .       (XE = XCB.OFFSET)
     0          4318                 .
   3C6          4319 D3520014        tsckpr  lbyts,4 x2,a_cst,x0,dualstat .Fetch dual state flag.
   3CA          4320 920E001A                brrgt   x0,xe,tsckpr3       .Jump if 180 is idle.
   3CE          4321 9420006D                brxeq   x2,x0,async90        .Jump if not dual state.
   3D2          4322 D0510052                lbyts,1 x1,a_cst,x0,1pid8    .Get cpu index
   3D6          4323 D15E0002                lbyts,2 xe,a_cst,x0,dsprior  .Get 180 priority. (!! XE for call to RUNNOS)
   3DA          4324 D171100E                lbyts,2 x1,a_dscb,x1,np170pr .Get 170 priority.
   3DE          4325 A8E20FFC                shfc    x2,xe,x0,-4          .Shift off sub priority.
   3E2          4326 A8110FFC                shfc    x1,x1,x0,-4
   3E6          4327 92210061                brrgt   x2,x1,async90        .Run 180 if 180pr > 170pr.
   3EA          4328 9121000A                brrne   x2,x1,tsckpr3        .Run 170 if 170pr > 180 pr.
   3EE          4329 AD220003                isob    x2,x2,x0,0003(8)     .Isolate 180 subpriority.
   3F2          4330 3F00                    entl    x0,0                 .Read the free running clock to calculate
   3F4          4331 0800                    cpytx   x0,x0                .  the 170 subpriority.
   3F6          4332 AD010B83                isob    x1,x0,x0,5603(8)
   3FA          4333 96210057                brxgt   x2,x1,async90        .Jump if 180 has highest priority.
     0          4334                 .
   3FE          4335 82510006        tsckpr3 lx       x1,a_cst,discntl     .Dont go to NOS if async flags are set.
   402          4336 9510FEF3                brxne   x1,x0,async
   406          4337 D0520007                lbyts,1 x2,a_cst,x0,nextstat .Test if CPU is being turned off/down.

                     0 1 2 3 4 5 6 7  12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                    CPU ASSEMBLY                    LEVEL
   40A          4338 9520000F                brxne   x2,x0,idle              .  Go to idle loop if being turned off/down.
   40E          4339 8F090004                addpxq  a_inret,x0,tsckpr5      .Set up return address and
   412          4340 940002AB                brxeq   x0,x0,run_nos           .  run NOS 170.
   416          4341 3FC2            tsckpr5 ent1    x0,r_te                 .Enable traps (RUNNOS exits with disabled).
   418          4342 0F00                    cpyxs   x0,x0
   41A          4343 0B62                    cpyax   x2,a_xcb                .Test for idle system.
   41C          4344 92200046                brrgt   x2,x0,async90           .Run user task if system not idle.
   420          4345 82510006                lx      x1,a_cst,discnt1        .Cycle the loop if task switch/async.
   424          4346 9510FEE2                brxne   x1,x0,async
     0          4347                 .
     0          4348                 .  Idle if no 180 task was found ready.
     0          4349                 .
   428          4350 3F60            idle    ent1    x0,r_mm                 .Disable asynchronous traps.
   42A          4351 8D01DB6C                ente    x1,m_mtrmsk-m_mcrasy-m_mcrsw . and short_warning
   42E          4352 0F01                    cpyxs   x1,x0
   430          4353 0000000000000000 idle3  bss     0
   430          4354 D0510007                lbyts,1 x1,a_cst,x0,nextstat    .Check the next_state of this CPU
   434          4355 94100009                brxeq   x1,x0,idle4             .If state <> ON THEN CPU state changed
   438          4356 8D0000C7                ente    x0,x_envir1             .Set up call to mtp$process_cpu_state_change
   43C          4357 091F                    cpyaa   af,a_csf
   43E          4358 8E3E04E0                addaq   ae,a_bindin,16*proc_cpu
   442          4359 B5EF0000                callseg bs_rqtbl,ae,af          .Call mtp$process_cpu_state_change
   446          4360 B3040018        idle4   enta    x0,40018(16)            .Kill some time by doing a
   44A          4361 A900002C                shfx    x0,x0,x0,44             . double precision divide
   44E          4362 3D01                    entp    x1,0
   450          4363 3700                    divd    x0,x0
   452          4364 8F0C0000        idle5   addpxq  a_extret,x0,idle5        .Branch if EXT INT is set - return
   456          4365 9F8003CB                brcr    8,0,extrq               .  to retest again - loop til no EXT INT
   45A          4366 82510006                lx      x1,a_cst,discnt1        .Exit when flags are set
   45E          4367 9510001B                brxne   x1,x0,idle10
   462          4368 8F0DFFF8                addpxq  a_sitret,x0,idle5
   466          4369 9FB00273                brcr    11,0,prsit              .Branch if SIT is set - return to idle5.
   46A          4370 9F220015                brcr    2,2,idle10              .Fall out if short_warning is set.
   46E          4371 D0510030                lbyts,1 x1,a_cst,x0,caldisp     .Exit idle loop if call_dispatcher
   472          4372 95100011                brxne   x1,x0,idle10            . is set
     0          4373                 .
   476          4374 3D00                    entp    x0,0                    .Read microsecond clock
   478          4375 0802                    cpytx   x2,x0
   47A          4376 844F0492                la      af,a_root,frc_p         .Get FRC time to call dispatcher
   47E          4377 82F10000                lx      x1,af,0
   482          4378 9712FFD7                brxge   x1,x2,idle3             .Jump if not time to call dispatcher
   486          4379 3D10                    entp    x0,1
   488          4380 14F1                    lbset   x1,af,x0                .Test/set bit 1 of FRC time-if already
   48A          4381 9110FFD3                brrne   x1,x0,idle3             . set stay in idle loop-another processor
     0          4382                 .                                       . is updating the timed wait queue
   48E          4383 3D11            idle9   entp    x1,1                    .Exit idle loop and call dispatcher.
   490          4384 D8510030                sbyts,1 x1,a_cst,x0,caldisp
     0          4385                 .
   494          4386 82410099        idle10  lx      x1,a_root,sitvalue      .Put big number in SIT to reduce
   498          4387 3F62                    ent1    x0,r_sit                .  likelyhood of unnecessary SIT.
   49A          4388 0F01                    cpyxs   x1,x0
   49C          4389 3F60                    ent1    x0,r_mm                 .Restore monitor mask.
   49E          4390 8D01FFFC                ente    x1,m_mtrmsk
   4A2          4391 0F01                    cpyxs   x1,x0

                     0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                    CPU ASSEMBLY                    LEVEL
   4A4          4392 9400FEA2                brxeq   x0,x0,async
     0          4393                 .
     0          4394                 .  Reload PIT for current 180 task.
     0          4395                 .
   4A8          4396 3FC9            async90 ent1    x0,r_pit                .Reload monitor clock (PIT).
   4AA          4397 0F04                    cpyxs   x_clock,x0
     0          4398                 .
     0          4399                 .   End of task switch loop.
```

6- 0

0000 828E 0001

```
 OFFSET BIT     LINE BINARY            SOURCE STATEMENT                                                  CPU ASSEMBLY                 LEVEL
     0          4401                   ...............   beginning of critical region ...........
     0          4402                   .
     0          4403                   .                 CRITICAL REGION - between labels BCRIT1 and ECRIT1
     0          4404                   .
     0          4405                   .  If any changes are made in thie following code,
     0          4406                   .  be sure to look at the code in the trap handler.
     0          4407                   .  Under certain circumstances, P will be reset to the beginning
     0          4408                   .  of the critical region.
     0          4409                   .
     0          4410                   .
   4AC          4411 0000000000000000 BCRIT1   bss      0
   4AC          4412 0000000000000000 exchloop bss      0
   4AC          4413 82510006                   lx       x1,a_cst,discntl        .Get async/taskswitch flags.
   4B0          4414 9510FE9A                   brxne    x1,x0,intdislp          .Jump if async or taskswitch.
   4B4          4415 D1520010                   lbyts,2 x2,a_cst,x0,taskid       .Get taskid of current task.
   4B8          4416 D9610030                   sbyts,2 x1,a_xcb,x0,xpmcr        .Clear user's MCR
     0          4417                            xtrace  0,0,x1,x0,ae
   4E4          4445 A922000D                   shfx     x2,x2,x0,13
   4E8          4446 B1521FA1                   keypoint oscmtr,x2,oskexc8x
   4EC          4447 0200              ECRIT1   exchange
     0          4448                   ...............   end of critical region ...........
     0          4449                   .
     0          4450                   .
     0          4451                   .  Get the MCR from the user XP.
     0          4452                   .
     0          4453                   . ---------- Test code was omitted at compilation time ----------
   4EE          4454 D1630030                   lbyts,2 x_mcr,a_xcb,x0,xpmcr     .Get MCR from user XP
     0          4455                            xtrace  1,x_mcr,x1,x0,ae         .Save MCR in trace buffer.
   520          4483 A931000D                   shfx     x1,x_mcr,x0,13          .Keypoint MCR.
   524          4484 B1510FA1                   keypoint oscmtr,x1,oskexc8
     0          4485                   .
     0          4486                   .
     0          4487                   .  Special case an MCR of EXCH ONLY. This is the most frequent interrupt in
     0          4488                   .  dual state. If EXCH is set and other bits are set as well, the EXCH will
     0          4489                   .  be handled later.
     0          4490                   .
   528          4491 8D010400                   ente     x1,m_mcrexc             .Check for only EXCH set.
   52C          4492 95130014                   brxne    x1,x_mcr,ckhdw          .Jump if not EXCH only.
   530          4493 3FC9                       entl     x0,r_pit                .Stop the clock.
   532          4494 0E04                       cpysx    x_clock,x0
   534          4495 844E0476                   la       ae,a_root,nosxp
   538          4496 D9E10030                   sbyts,2 x1,ae,x0,xpmcr
   53C          4497 8F090006                   addpxq   a_inret,x0,ckexsp5      .Set up return address.
   540          4498 D15E0002                   lbyts,2 xe,a_cst,x0,dsprior      .Get current 180 priority.
   544          4499 94000212                   brxeq    x0,x0,run_nos           .Go run NOS 170.
   548          4500 3FC9              ckexsp5  entl     x0,r_pit                .Start monitor clock.
   54A          4501 0F04                       cpyxs    x_clock,x0
   54C          4502 3FC2                       entl     x0,r_te                 .Enable traps.
   54E          4503 0F00                       cpyxs    x0,x0
   550          4504 9400FFAE                   brxeq    x0,x0,exchloop
     0          4505                   .
     0          4506                   .
     0          4507                   .  Process hardware errors - (DUE, SHORT WARNING).
     0          4508                   .

                0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
 OFFSET BIT     LINE BINARY            SOURCE STATEMENT                                                  CPU ASSEMBLY                 LEVEL
   554          4509 8D01A000          ckhdw    ente     x1,m_mcrhdw             .Check for hardware errors
   558          4510 1A31                       andx     x1,x_mcr
   55A          4511 90100023                   brreq    x1,x0,ckasync
   55E          4512 A8310012                   shfc     x1,x_mcr,x0,18          .Check for short warning.
   562          4513 9310000A                   brrge    x1,x0,ckdue             .Jump if no short warning.
     0          4514                            monreq   pswarn
     0          4545                   .
   576          4546 A8310010          ckdue    shfc     x1,x_mcr,x0,16          .Check for DUE.
   57A          4547 93100013                   brrge    x1,x0,ckasync
   57E          4548 0502                       purge    x0,2                    .Purge cache and map.
   580          4549 050F                       purge    x0,15
   582          4550 3D01                       entp     x1,0                    .Set up plist
   584          4551 83110000                   sx       x1,a_csf,0
   588          4552 85160012                   sa       a_xcb,a_csf,18
     0          4553                            monreq   proc_due
   59C          4584 8D030490                   ente     x_mcr,m_mcrasy          .Force async interrupts since these
     0          4585                   .                                           may be invalid because of DUE.
     0          4586                   .
     0          4587                   .  Process asynchronous interrupts.
     0          4588                   .
   5A0          4589 8D010490          ckasync  ente     x1,m_mcrasy             .Check for asynchronous interrupt
   5A4          4590 1A31                       andx     x1,x_mcr
   5A6          4591 9010002D                   brreq    x1,x0,ckuser            .Jump if no asynchronous interrupt
   5AA          4592 3FC9                       entl     x0,r_pit                .Stop the monitor clock.
   5AC          4593 0E04                       cpysx    x_clock,x0
   5AE          4594 A831001B                   shfc     x1,x_mcr,x0,27          .Check for SIT.
   5B2          4595 93100006                   brrge    x1,x0,ckextint          .Jump if no SIT.
   5B6          4596 8F0D0004                   addpxq   a_sitret,x0,ckextint    .Set up return address.
   5BA          4597 940001C9                   brxeq    x0,x0,prsit             .Go process SIT interrupt.
   5BE          4598 A9310018          ckextint shfx     x1,x_mcr,x0,24          .Check for EXT INT
   5C2          4599 8F0C0004                   addpxq   a_extret,x0,ckexch
   5C6          4600 92010313                   brrgt    x0,x1,extrq             .Jump if  EXT INT.
   5CA          4601 8D010090          ckexch   ente     x1,m_mcrexs             .Clear SIT and EXTINT.
   5CE          4602 1C13                       inhx     x_mcr,x1
   5D0          4603 D9630030                   sbyts,2 x_mcr,a_xcb,x0,xpmcr     .Clear MCR - see trap handler.
   5D4          4604 A9310015                   shfx     x1,x_mcr,x0,21          .Check for EXCH
   5D8          4605 93100010                   brrge    x1,x0,ckasyncx          .Jump if no EXCH
   5DC          4606 8D010400                   ente     x1,m_mcrexc             .Set EXCH bit in NOS XP
   5E0          4607 844E0476                   la       ae,a_root,nosxp
   5E4          4608 D9E10030                   sbyts,2 x1,ae,x0,xpmcr
   5E8          4609 8F090006                   addpxq   a_inret,x0,ckexch5
   5EC          4610 D15E0002                   lbyts,2 xe,a_cst,x0,dsprior      .Get current 180 priority.
   5F0          4611 940001BC                   brxeq    x0,x0,run_nos           .Run NOS
   5F4          4612 3FC2              ckexch5  entl     x0,r_te                 .Enable traps
   5F6          4613 0F00                       cpyxs    x0,x0
   5F8          4614 3FC9              ckasyncx entl     x0,r_pit                .Start monitor clock.
   5FA          4615 0F04                       cpyxs    x_clock,x0
   5FC          4616 9430FF58                   brxeq    x_mcr,x0,exchloop
     0          4617                   .
     0          4618                   .  Process faults normally handled in job mode via trap handler.
     0          4619                   .
   600          4620 8D011B0C          ckuser   ente     x1,j_mcrusr             .Check for condition that will
   604          4621 1A31                       andx     x1,x_mcr                .be processed in job mode
   606          4622 9010000B                   brreq    x1,x0,ckpf              .Jump if no job mode request

                0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                        CPU ASSEMBLY                    LEVEL
   60A          4623 050F                     purge   x0,15                   .Purge required after INV SEG.
     0          4624                          monreq  rqfault,0,exchloop
     0          4655                  .
     0          4656                  .   Check for a Page Fault.
     0          4657                  .
   61C          4658 A9310039         ckpf    shfx    x1,x_mcr,x0,57          .Check for a page fault.
   620          4659 9710000A                 brxge   x1,x0,ckmcall           .Jump if no page fault.
     0          4660                          monreq  rqpf,0,exchloop
     0          4691                  .
     0          4692                  .   Check for a System Call request.
     0          4693                  .
   634          4694 A9310003A        ckmcall shfx    x1,x_mcr,x0,58          .Check for a SYSTEM CALL.
   638          4695 97100027                 brxge   x1,x0,ckucr             .Jump if no SYSTEM CALL.
   63C          4696 D0610088                 1byts,1 x1,a_xcb,x0,xpxregs     .Get request code
   640          4697 D0620002                 1byts,1 x2,a_xcb,x0,2           .Get p.rn from XCB
   644          4698 8E6E0088                 addaq   ae,a_xcb,xpxregs        .Set up plist to point to
   648          4699 851E0000                 sa      ae,a_csf,0              .X_regs of current task
   64C          4700 8D0E0055                 ente    xe,mtrqmax              .Check for max req code.
   650          4701 97E10003                 brxge   xe,x1,ckmcall5          .Jump if ok.
   654          4702 3D01                     entp    x1,0
   656          4703 A9220FFC         ckmcall5 shfx   x2,x2,x0,-4
     0          4704                          monreq  x1,x2,exchloop
     0          4735                  .
     0          4736                  .   If control gets here, there is a chance that the MCR value was zero
     0          4737                  .   (except for possible async/sel interrupts). Check for a UCR fault that
     0          4738                  .   caused a monitor exchange because traps were disabled.
     0          4739                  .
   686          4740 D1610028         ckucr   1byts,2 x1,a_xcb,x0,xpucr       .Check for fatal UCR faults.
   68A          4741 8D00CC00                 ente    x0,j_usrabt
   68E          4742 1A01                     andx    x1,x0
   690          4743 9410FF0E                 brxeq   x1,x0,exchloop          .Jump if no fatal faults.
   694          4744 D1610010                 1byts,2 x1,a_xcb,x0,xpflgte     .Check for traps enabled
   698          4745 AD110F81                 isob    x1,x1,x0,7601(8)
   69C          4746 2921                     decr    x1,2
   69E          4747 9010FF07                 brreq   x1,x0,exchloop          .Jump if traps not disabled
     0          4748                          monreq  rqfault,0,exchloop
```



```
OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                        CPU ASSEMBLY                    LEVEL
     0          4780                  ..........................................................................................
     0          4781                  .
     0          4782                  .   Trap Handling Routine for traps that occur in Monitor Mode
     0          4783                  .
     0          4784                  ..........................................................................................
   6B8          4785                          align   0,8
   6B8          4786 0000000000000000 traprtn bss     0
   6B8          4787 3FC8                     entl    x0,r_pit                .Save PIT
   6BA          4788 0E04                     cpysx   x_clock,x0
   6BC          4789 3FCA                     entl    x0,r_kef0               .Save and clear KEF.
   6BE          4790 0E05                     cpysx   x_kef,x0
   6C0          4791 0F00                     cpyxs   x0,x0
   6C2          4792 84340572                 la      a_root,a_bindin,bs_root
   6C6          4793 84470470                 la      a_dscb,a_root,nostab
   6CA          4794 8D000047                 ente    x0,r_bc
   6CE          4795 0E01                     cpysx   x1,x0                   .get  base constant.
   6D0          4796 0B42                     cpyax   x2,a_root
   6D2          4797 2421                     addx    x1,x2                   .form pointer to cst
   6D4          4798 0A15                     cpyxa   a_cst,x1
   6D6          4799 8E000020                 addaq   a0,a0,mstkfram
   6DA          4800 8515000A                 sa      a_cst,a_csf,10          .Sace CST_P in p-list.
     0          4801                  .
   6DE          4802 D1230030                 1byts,2 x_mcr,a_psa,x0,sfsa_mcr .Get MCR
     0          4803                          xtrace  2,x_mcr,x1,x0,ae        .Save MCR in trace buffer.
     0          4831                  .
     0          4832                  .   DO NOT halt the processor if a DUE or SHORT WARNING occurred.
     0          4833                  .
   710          4834 8D01FB2C                 ente    x1,m_mcrhlt+m_mcrhdw    .Check for fatal errors.
   714          4835 1A31                     andx    x1,x_mcr
   716          4836 9410002D                 brxeq   x1,x0,trhdwx
   71A          4837 A8310032                 shfc    x1,x_mcr,x0,50          .Check short warning.
   71E          4838 9710000A                 brxge   x1,x0,trckdue
     0          4839                          monreq  pswarn
     0          4870                  .
   732          4871 A8310030         trckdue shfc    x1,x_mcr,x0,48          .Check DUE.
   736          4872 97100013                 brxge   x1,x0,trhdw5
   73A          4873 8D030490                 ente    x_mcr,m_mcrasy          .Force all async interrupts.
   73E          4874 0502                     purge   x0,2                    .Purge cache and map.
   740          4875 050F                     purge   x0,15
   742          4876 3D21                     entp    x1,2                    .Set up plist
   744          4877 83110000                 sx      x1,a_csf,0              .Store code to indicate DUE in monitor.
   748          4878 85120012                 sa      a2,a_csf,18             .Store pointer to save area.
     0          4879                          monreq  proc_due
     0          4910                  .
   75C          4911 8D015B2C         trhdw5  ente    x1,m_mcrhlt             .Halt if any fatal
   760          4912 1A31                     andx    x1,x_mcr                .  conditions are set
   762          4913 94100007                 brxeq   x1,x0,trhdwx
   766          4914 8D0000FF         trstop  ente    x0,0ff(16)
   76A          4915 B53E00B1                 callseg bs_merrs,a_bindin,ae    .Call mtp$mtr_error_stop.
   76E          4916 0000                     halt                            .should not return, halt if it does
   770          4917 0000000000000000 trhdwx  bss     0
     0          4918                  .
     0          4919                  . Process page fault in monitor mode
     0          4920                  .
```

```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY                LEVEL
   770          4921 8D010040                  ente    x1,m_mcrpf
   774          4922 1A31                      andx    x1,x_mcr
   776          4923 94100013                  brxeq   x1,x0,nopf          .If no page fault
   77A          4924 85120002                  sa      a2,a_csf,2          .Plist = a2
   77E          4925 8E1E0010                  addaq   ae,a_csf,16         .Plist = VAR halt
   782          4926 851E0008                  sa      ae,a_csf,8
   786          4927 091E                      cpyaa   ae,a_csf
   788          4928 8D0000C7                  ente    x0,x_envir1
   78C          4929 B53E00B3                  callseg bs_pgflt,a_bindin,ae
   790          4930 D0110010                  1byts,1 x1,a_csf,x0,16      .Get returned value of halt
   794          4931 8515000A                  sa      a_cst,a_csf,10      .Re-save CST_P in p-list.
   798          4932 951OFFE7                  brxne   x1,x0,trstop        .Jump if fatal error
   79C          4933 0000000000000000 nopf     bss     0
     0          4934                   .
     0          4935                   .  If the trap occurred between the labels BCRIT1 and ECRIT1, reset the
     0          4936                   .  trapped 'P' address to the label BCRIT1.
     0          4937                   .
   79C          4938 D3210004                  1byts,4 x1,a_psa,x0,4       .Get P from SFSA.
   7A0          4939 8F0EFEA6                  addpxq  ae,x0,ecrit1
   7A4          4940 0BE2                      cpyax   x2,ae
   7A6          4941 92120009                  brrgt   x1,x2,trresex
   7AA          4942 8F0EFE81                  addpxq  ae,x0,bcrit1
   7AE          4943 0BE2                      cpyax   x2,ae
   7B0          4944 92210004                  brrgt   x2,x1,trresex
   7B4          4945 852E0002                  sa      ae,a_psa,2
   7B8          4946 0000000000000000 trresex  bss     0
     0          4947                   .
     0          4948                   .  Protect against the case where 1) a SIT or EXT INT occurred in 180 job
     0          4949                   .  mode to cause an exchange to monitor and 2) prior to processing the
     0          4950                   .  SIT/EXT INT an EXCH occurred to cause a trap.
     0          4951                   .
   7B8          4952 84560020                  la      a_xcb,a_cst,xcbp    .Fetch XCB pointer.
   7BC          4953 0B61                      cpyax   x1,a_xcb            .Skip this check if NIL.
   7BE          4954 9201000A                  brrgt   x0,x1,trnom
   7C2          4955 D1610030                  1byts,2 x1,a_xcb,x0,xpmcr   .Fetch MCR from current XP.
   7C6          4956 1813                      iorx    x_mcr,x1            .Merge with trapped MCR.
   7C8          4957 8D020490                  ente    x2,m_mcrasy
   7CC          4958 1C21                      inhx    x1,x2
   7CE          4959 D9610030                  sbyts,2 x1,a_xcb,x0,xpmcr   .Store MCR less asynch bits.
   7D2          4960 0000000000000000 trnom    bss     0
     0          4961                   .
     0          4962                   .  Process asynchronous interrupts.
     0          4963                   .
   7D2          4964 8D010490                  ente    x1,m_mcrasy         .Check for asynchronous interrupts.
   7D6          4965 1A31                      andx    x1,x_mcr
   7D8          4966 94100028                  brxeq   x1,x0,trasy15       .Jump if no asynchronous interrupts.
   7DC          4967 A831001B                  shfc    x1,x_mcr,x0,27      .Check for SIT.
   7E0          4968 93100006                  brrge   x1,x0,trasy5        .Jump if no SIT.
   7E4          4969 8F0D0004                  addpxq  a_sitret,x0,trasy5  .Set up return address.
   7E8          4970 940000B2                  brxeq   x0,x0,prsit         .Go process SIT interrupt.
   7EC          4971 A9310018          trasy5  shfx    x1,x_mcr,x0,24      .Check for EXT INT
   7F0          4972 8F0C0004                  addpxq  a_extret,x0,trasy8
   7F4          4973 920101FC                  brrgt   x0,x1,extrq         .Jump if  EXT INT
   7F8          4974 A9310015          trasy8  shfx    x1,x_mcr,x0,21      .Check for EXCH

                     0 1 2 3 4 5 6 7  12345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY                LEVEL
   7FC          4975 93100016                  brrge   x1,x0,trasy15       .Jump if no EXCH
   800          4976 3F05                      ent1    x0,5                .Set EXCH bit in MCR.
   802          4977 844E0476                  la      ae,a_root,nosxp
   806          4978 89E00030                  sbit    x0,ae,xpmcr,x0
   80A          4979 D14104F8                  1byts,2 x1,a_root,x0,mtrprior .Get 180 monitor priority.
   80E          4980 D15E0002                  1byts,2 xe,a_cst,x0,dsprior .Dont change if already greater.
   812          4981 96E10003                  brxgt   xe,x1,trasy9
   816          4982 0D1E                      cpyxx   xe,x1
   818          4983 3F61              trasy9  ent1    x0,r_jps            .Check if NOS170 is the current task.
   81A          4984 0E01                      cpysx   x1,x0
   81C          4985 D3520014                  1byts,4 x2,a_cst,x0,dualstat
   820          4986 8F090004                  addpxq  a_inret,x0,trasy15
   824          4987 952100A2                  brxne   x2,x1,run_nos       .Go run NOS 170.
     0          4988                   .
     0          4989                   .   Halt processor if fatal UCR fault occurred.
     0          4990                   .
   828          4991 8D01EDFF          trasy15 ente    x1,m_usrabt         .Check for fatal UCR fault
   82C          4992 D1220028                  1byts,2 x2,a_psa,x0,sfsa_ucr .Get UCR
   830          4993 1A21                      andx    x1,x2
   832          4994 951OFF9A                  brxne   x1,x0,trstop        .Jump if fatal error
     0          4995                   .
     0          4996                   .
     0          4997                   .   Set TRAP ENABLE DELAY and return.
     0          4998                   .
   836          4999 3FC3              trexit  ent1    x0,r_ted            .Set trap enable delay
   838          5000 0F00                      cpyxs   x0,x0
   83A          5001 3FCA                      ent1    x0,r_kef0           .Restore KEF.
   83C          5002 0F05                      cpyxs   x_kef,x0
   83E          5003 3FC9                      ent1    x0,r_pit            .Restore PIT
   840          5004 0F04                      cpyxs   x_clock,x0
   842          5005 0400                      return
```

```
 OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY           LEVEL
     0           5007                   ..........................................................................
     0           5008                   .
     0           5009                   .       This routine updates the request statistics and calls the
     0           5010                   .       appropriate request processor.
     0           5011                   .       Interlocking of most monitor functions is performed by this routine.
     0           5012                   .
     0           5013                   .       Entry condition:
     0           5014                   .            a_rqtbl  - pointer to request table entry for request.
     0           5015                   .            ae        - pointer to binding section entry for request proc.
     0           5016                   .            a_rq_ret - return address
     0           5017                   .
     0           5018                   ..........................................................................
     0           5019                   .
   844           5020 0000000000000000 rqproc   bss     0
   844           5021 8241008B                  lx      x1,a_root,multpro      .Test for multiple processors.
   848           5022 3FC9                      entl    x0,r_pit               .Get current PIT
   84A           5023 0E02                      cpysx   x2,x0                  .!! X2 contains PIT thruout this proc.
   84C           5024 94100060                  brxeq   x1,x0,rqpr14           .Jump if not multi_processor
   850           5025 D0A10001                  lbyts,1 x1,a_rqtbl,x0,il       .Test if request must be interlocked.
   854           5026 9410005C                  brxeq   x1,x0,rqpr14           .  jump if interlock not required
   858           5027 8E4F2A50                  addaq   af,a_root,il_tbl       .Calc pointer to interlock word.
   85C           5028 A9110003                  shfx    x1,x1,x0,3
   860           5029 2A1F                      addax   af,x1                  .PVA of interlock table
   862           5030 3D00                      entp    x0,0                   .Try to set lock
   864           5031 14F1                      lbset   x1,af,x0
   866           5032 90100050                  brreq   x1,x0,rqpr12           .Jump if interlock obtained.
     0           5033                   .
   86A           5034 3D00             rqpr4    entp    x0,0                   .Keep trying to set lock.
   86C           5035 14F1                      lbset   x1,af,x0
   86E           5036 9010003D                  brreq   x1,x0,rqpr6            .Jump if lock obtained.
   872           5037 3D11                      entp    x1,1                   .Kill some time by doing a
   874           5038 2711                      divx    x1,x1                  .  divide
   876           5039 D0510007                  lbyts,1 x1,a_cst,x0,nextstat   .Check for halt cpu request
   87A           5040 9510001F                  brxne   x1,x0,rqpr55
   87E           5041 D051001F                  lbyts,1 x1,a_cst,x0,cp_state+cp_nxtst  .check for step cpu request
   882           5042 9410FFF4                  brxeq   x1,x0,rqpr4
     0           5043                           errstop stepmes
   8B4           5059 9400FFDB                  brxeq   x0,x0,rqpr4            .Continue trying to get the lock.
     0           5060                  rqpr55   errstop cpudown
   8E6           5076 0000                      halt                           .Should not return.
     0           5077                   .
   8E8           5078 3FC9             rqpr6    entl    x0,r_pit               .Restore PIT- don't charge user for wait.
   8EA           5079 0E01                      cpysx   x1,x0                  .Read PIT to calc wait time.
   8EC           5080 824D009C                  lx      xd,a_root,lockwait     .Update lock wait time and count.
   8F0           5081 824E009D                  lx      xe,a_root,lockwait+8
   8F4           5082 2521                      subx    x1,x2
   8F6           5083 1B11                      notx    x1,x1
   8F8           5084 24D1                      addx    x1,xd
   8FA           5085 8341009C                  sx      x1,a_root,lockwait
   8FE           5086 101E                      incx    xe,1
   900           5087 834E009D                  sx      xe,a_root,lockwait+8
   904           5088 0F02                      cpyxs   x2,x0
     0           5089                   .
   906           5090 0BF1             rqpr12   cpyax   x1,af                  .x1 = pva of interlock table

                 0 1 2 3 4 5 6 7  12345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
 OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY           LEVEL
   908           5091 85F50002                  sa      a_cst,af,lockcp        .Store ID of locking CPU
   90C           5092 0000000000000000 rqpr14   bss     0                      .x1 = zero if no interlock.
   90C           5093 8D0000C7                  ente    x0,x_envir1            .Process the request
   910           5094 091F                      cpyaa   af,a_csf
   912           5095 B5EF0000                  callseg bs_rqtbl,ae,af
   916           5096 3FC9                      entl    x0,r_pit
   918           5097 0E0D                      cpysx   xd,x0                  .Calculate time to process the request
   91A           5098 82AE0001                  lx      xe,a_rqtbl,totalt      .Update total and max time
   91E           5099 82AF0002                  lx      xf,a_rqtbl,rqcntmax
   922           5100 25D2                      subx    x2,xd
   924           5101 242E                      addx    xe,x2
   926           5102 83AE0001                  sx      xe,a_rqtbl,totalt
   92A           5103 281F                      incr    xf,1
   92C           5104 A8FE0020                  shfc    xe,xf,x0,32            .Check if new maximum time.
   930           5105 93E20005                  brrge   xe,x2,rqpr20           .  Jump if not new max.
   934           5106 0C2E                      cpyrr   xe,x2
   936           5107 A8EF0020                  shfc    xf,xe,x0,32
   93A           5108 83AF0002         rqpr20   sx      xf,a_rqtbl,rqcntmax
   93E           5109 94100006                  brxeq   x1,x0,rqpr30           .Exit if no lock
   942           5110 3F00                      entl    x0,0
   944           5111 0A1F                      cpyxa   af,x1
   946           5112 83F00000                  sx      x0,af,ilflag           .Clear lock
   94A           5113 2FB0             rqpr30   brdir   a_rq_ret,x0            .Return
```

```
                 0 1 2 3 4 5 6 7  12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT   LINE BINARY            SOURCE STATEMENT                                           CPU ASSEMBLY     LEVEL
    0        5115                   .........................................................................
    0        5116                   .
    0        5117                   .  This routine is called whenever a SIT interrupt occurs.
    0        5118                   .
    0        5119                   .........................................................................
    0        5120                   .
  94C        5121 0000000000000000 prsit     bss      0
  94C        5122 3D1F                       entp     xf,1                       .Set up X15 with 'TRUE'.
  94E        5123 3D00                       entp     x0,0                       .Set up X0 with 'FALSE'.
  950        5124 0802                       cpytx    x2,x0                      .Free running clock ->X2.
  952        5125 83420007                   sx       x2,a_root,scb+scbnsrv      .Update '180 alive' flag.
  956        5126 83520001                   sx       x2,a_cst,cpwell            .Update cpu alive flag.
  95A        5127 D85F0030                   sbyts,1  xf,a_cst,x0,caldisp
    0        5128                   .
  95E        5129 82410099                   lx       x1,a_root,sitvalue         .Reset SIT.
  962        5130 3F62                       entl     x0,r_sit
  964        5131 0F01                       cpyxs    x1,x0
    0        5132                   .
  966        5133 2FD0                       brdir    a_sitret,x0
```



```
OFFSET BIT   LINE BINARY            SOURCE STATEMENT                                           CPU ASSEMBLY     LEVEL
    0        5135                   .........................................................................
    0        5136                   .
    0        5137                   .   The purpose of this routine is to give control to NOS170.
    0        5138                   .
    0        5139                   .   NOTE: This routine may be entered with traps disabled or enabled.
    0        5140                   .         Reentry of the routine from the trap handler is prevented by
    0        5141                   .         setting the JPS register to NOS_JPS.  This routine exits with traps
    0        5142                   .         disabled unless NOS is not present. In this case no change is
    0        5143                   .         made to the TE register.
    0        5144                   .
    0        5145                   .     Enter with 180 priority in XE.
    0        5146                   .
    0        5147                   .........................................................................
    0        5148                   .
  968        5149 0000000000000000 run_nos   bss      0
    0        5150                   .
  968        5151 D3510014                   lbyts,4  x1,a_cst,x0,dualstat   .Exit if no dual state.
  96C        5152 9410013F                   brxeq    x1,x0,runexit
    0        5153                   .
  970        5154 3F61                       entl     x0,r_jps
  972        5155 0F01                       cpyxs    x1,x0                  .Copy NOS_JPS to JPS reg
  974        5156 84480476                   la       a_innosx,a_root,nosxp
    0        5157                   .
  978        5158 D0510052                   lbyts,1  x1,a_cst,x0,lpid8      .Store 180 priority.
  97C        5159 D97E1096                   sbyts,2  xe,a_dscb,x1,np180pr
  980        5160 A91E0FFD                   shfx     xe,x1,x0,-3
    0        5161                   .
  984        5162 3D07                       entp     x_infrc,0              .Get current time
  986        5163 0877                       cpytx    x_infrc,x_infrc
  988        5164 8242008D                   lx       x2,a_root,nosexit      .Update time not spent in NOS
  98C        5165 2572                       subx     x2,x_infrc
  98E        5166 1B22                       notx     x2,x2
  990        5167 A271E010                   lxi      x1,a_dscb,xe,npxtime
  994        5168 2421                       addx     x1,x2
  996        5169 A371E010                   sxi      x1,a_dscb,xe,npxtime
    0        5170                   .
    0        5171                   .  (BEGIN - EXCH loop).   Exchange to NOS170.
    0        5172                   .
  99A        5173 0000000000000000 runnos6   bss      0
    0        5174                            xtrace   3,0,x1,xe,ae
  9C2        5202 3FC2                       entl     x0,r_te                .Enable traps
  9C4        5203 0F00                       cpyxs    x0,x0
  9C6        5204 B1500FA2                   keypoint oscmtr,x0,oskexc7
  9CA        5205 0200                       exchange                        .EXCHANGE TO NOS (NOS-BE)
  9CC        5206 D1860030                   lbyts,2  x_inmcr,a_innosx,x0,xpmcr .Get MCR
  9D0        5207 A962000D                   shfx     x2,x_inmcr,x0,13
  9D4        5208 B1521FA2                   keypoint oscmtr,x2,oskexc7x
  9D8        5209 3FC0                       entl     x0,r_td                .Disable traps
  9DA        5210 0F00                       cpyxs    x0,x0
  9DC        5211 D1860030                   lbyts,2  x_inmcr,a_innosx,x0,xpmcr .Get MCR
  9E0        5212 8D020400                   ente     x2,m_mcrexc            .Clear MCR except for EXCH
  9E4        5213 1A62                       andx     x2,x_inmcr
  9E6        5214 D9820030                   sbyts,2  x2,a_innosx,x0,xpmcr
    0        5215                            xtrace   4,x_inmcr,xe,xd,ae     .Save NOS MCR in trace buffer
```

```
OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                    CPU ASSEMBLY              LEVEL
   0             5243                     .
   0             5244                     .   Process 'give up CPU' if only bit set in the MCR is 'system call'.
   0             5245                     .
 A18             5246 8D020020                    ente     x2,m_mcrmc1
 A1C             5247 95620006                    brxne    x_inmcr,x2,runnos8     .Jump if not sys call ONLY.
 A20             5248 D0820008                    lbyts,1  x2,a_innosx,x0,xpvmid .Make sure request is from .
 A24             5249 952000CC                    brxne    x2,x0,runnos50         .  170 state.
   0             5250                     .
   0             5251                     .   Process hardware errors - DUE.
   0             5252                     .
 A28             5253 3FCA                runnos8  ent1     x0,r_kef0             .Save and clear KEF.
 A2A             5254 0E05                        cpysx    x_kef,x0
 A2C             5255 0F00                        cpyxs    x0,x0
 A2E             5256 A8620010                    shfc     x2,x_inmcr,x0,16       .Check for DUE.
 A32             5257 9320001D                    brrge    x2,x0,runnos10         .Jump if no DUE.
 A36             5258 0502                        purge    x0,2                   .Purge cache and map.
 A38             5259 050F                        purge    x0,15
 A3A             5260 3F01                        ent1     x0,1                   .Set up plist - DUE in 170 mode
 A3C             5261 83100000                    sx       x0,a_csf,0
 A40             5262 85180012                    sa       a_innosx,a_csf,18
   0             5263                             monreq   proc_due
 A54             5294 8D020490                    ente     x2,0490(16)            .Force async bit - may be lost.
 A58             5295 1826                        iorx     x_inmcr,x2
 A5A             5296 D0820010                    lbyts,1  x2,a_innosx,x0,xpflgte .Check PROCESS-NOT-DAMAGED.
 A5E             5297 AA22001B                    shfr     x2,x2,x0,27
 A62             5298 93200005                    brrge    x2,x0,runnos10         .Jump if damaged.
 A66             5299 8D027FFF                    ente     x2,7fff(16)            .Clear DUE in MCR.
 A6A             5300 1A26                        andx     x_inmcr,x2
   0             5301
   0             5302                     .
   0             5303                     .   Process short warning conditions.
   0             5304                     .
 A6C             5305 A8620012            runnos10 shfc     x2,x_inmcr,x0,18       .Check for SHORT WARNING.
 A70             5306 9320000A                    brrge    x2,x0,runnos11         .Jump if no SHORT WARNING.
   0             5307                             monreq   pswarn
   0             5338                     .
   0             5339                     .   Process SIT interrupt.
   0             5340                     .
 A84             5341 A862001B            runnos11 shfc     x2,x_inmcr,x0,27       .Check for SIT.
 A88             5342 8F0D0004                    addpxq   a_sitret,x0,runnos12   .Go process SIT interrupt.
 A8C             5343 9202FF60                    brrgt    x0,x2,prsit            .Jump if SIT present.
   0             5344                     .
   0             5345                     .   Process EXTERNAL INTERRUPT.
   0             5346                     .
 A90             5347 A8620018            runnos12 shfc     x2,x_inmcr,x0,24       .Check for EXT INT.
 A94             5348 8F0C0004                    addpxq   a_extret,x0,runnos16
 A98             5349 920200AA                    brrgt    x0,x2,extrq            .Jump if  EXT INT.
   0             5350                     .
   0             5351                     .   Process SYSTEM CALL requests.
   0             5352                     .
 A9C             5353 3FCA                runnos16 ent1     x0,r_kef0             .Restore KEF flag.
 A9E             5354 0F05                        cpyxs    x_kef,x0
 AA0             5355 A862001A                    shfc     x2,x_inmcr,x0,26       .Check for SYSTEM CALL.
 AA4             5356 9320003A                    brrge    x2,x0,runnos20         .Jump if no SYSTEM CALL.

                 0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                    CPU ASSEMBLY              LEVEL
 AA8             5357 D0820008                    lbyts,1  x2,a_innosx,x0,xpvmid .Check whether 170 or 180 request.
 AAC             5358 94200008                    brxeq    x2,x0,runnos18         .Jump if 180 request.
 AB0             5359 A8620015                    shfc     x2,x_inmcr,x0,21       .Exit if EXCH not set.
 AB4             5360 93200084                    brrge    x2,x0,runnos50
 AB8             5361 9400FF71                    brxeq    x0,x0,runnos6
   0             5362                     .
   0             5363                     .  The following code implements a debug monitor function to allow the
   0             5364                     .  170 trap handler to issue keypoints with:  EIMTRCAL ISSUEKPT,val,NOHLT
   0             5365                     .
   0             5366                     .
 ABC             5367 82810011            runnos18 lx       x1,a_innosx,xpxregs    .fetch request code
 AC0             5368 3D82                        entp     x2,issuekpt
 AC2             5369 9121000C                    brrne    x2,x1,runnos19         .if not a keypoint request
 AC6             5370 3FC2                        ent1     x0,r_te                .enable traps
 AC8             5371 0F00                        cpyxs    x0,x0
 ACA             5372 82820013                    lx       x2,a_innosx,xpxregs+(8*2)
 ACE             5373 A922000C                    shfx     x2,x2,x0,12
 AD2             5374 B1420FA5                    keypoint oscdbug,x2,osktrap    .issue keypoint
 AD6             5375 9000FF62                    brreq    x0,x0,runnos6          .branch back into 170 state
   0             5376                     .
   0             5377                     .  The following code implements the DONTHING function to call the dispatcher.
   0             5378                     .
 ADA             5379 3D32                runnos19 entp     x2,donthing
 ADC             5380 9121000B                    brrne    x2,x1,rnnos19          .if not a call-dispatcher request
 AE0             5381 3D11                        entp     x1,1
 AE2             5382 D8510030                    sbyts,1  x1,a_cst,x0,caldisp
 AE6             5383 A8620015                    shfc     x2,x_inmcr,x0,21       .Exit if EXCH not set.
 AEA             5384 93200069                    brrge    x2,x0,runnos50
 AEE             5385 9400FF56                    brxeq    x0,x0,runnos6
   0             5386
 AF2             5387 3D00                rnnos19  entp     x0,0
 AF4             5388 83100000                    sx       x0,a_csf,0*8
 AF8             5389 8D000020                    ente     x0,020(16)
 AFC             5390 83100001                    sx       x0,a_csf,1*8
 B00             5391 8E1E0018                    addaq    ae,a_csf,24
 B04             5392 851E0010                    sa       ae,a_csf,2*8
   0             5393                             monreq   mm_ei,0,runnos24
   0             5424                     .
   0             5425                     .   Check for FATAL NOS170 errors.  Stop running 170 if fatal
   0             5426                     .   errors occured.
   0             5427                     .
 B18             5428 D0820008            runnos20 lbyts,1  x2,a_innosx,x0,xpvmid .Check whether 170 or 180 mode.
 B1C             5429 94200006                    brxeq    x2,x0,runnos21         .Jump if 180 mode.   .
 B20             5430 82820006                    lx       x2,a_innosx,6*8        .Get exit mode halt bit from NOS XP
 B24             5431 A922001F                    shfx     x2,x2,x0,31
 B28             5432 8D01DB4C            runnos21 ente     x1,0DB4C(16)           .Check for fatal 170 MCR
 B2C             5433 1A61                        andx     x1,x_inmcr
 B2E             5434 96020004                    brxgt    x0,x2,runnos22         .Jump if exit mode halt is set and 170.
 B32             5435 90100027                    brreq    x1,x0,runnos30         .Jump if no fatal 170 errors
 B36             5436 3D20                runnos22 entp     x0,2
 B38             5437 83100000                    sx       x0,a_csf,0
 B3C             5438 83110001                    sx       x1,a_csf,1*8
 B40             5439 8E1E0018                    addaq    ae,a_csf,24
 B44             5440 851E0010                    sa       ae,a_csf,2*8

                 0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890
```

```
 OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                         CPU ASSEMBLY          LEVEL
     0           5441                            monreq  mm_ei
     0           5472                  .
   B58           5473 8515000A         runnos24 sa      a_cst,a_csf,10        .Restore CST_P in p-list.
   B5C           5474 D0120018                  lbyts,1 x2,a_csf,X0,24
   B60           5475 90200010                  brreq   x2,x0,runnos30        .if not a fatal nos error
   B64           5476 A9610FF1                  shfx    x1,x_inmcr,x0,-15     .Store termination status -
   B68           5477 1011                      incx    x1,1                  . 2=DUE, 1=other
   B6A           5478 D8410441                  sbyts,1 x1,a_root,x0,os_terms
   B6E           5479 3F00                      entl    x0,0                  .Clear dual state flag.
   B70           5480 DB500014                  sbyts,4 x0,a_cst,x0,dualstat  .Stop running NOS170.
   B74           5481 D0510052                  lbyts,1 x1,a_cst,x0,lpid8     .Clear 170 priority.
   B78           5482 D970100E                  sbyts,2 x0,a_dscb,x1,np170pr
   B7C           5483 94000020                  brxeq   x0,x0,runnos50
     0           5484                  .
     0           5485                  .   Check if it is time to run 180.  If not, exchange back to 170.
     0           5486                  .   If 180 needs the CPU and NOS170 is in job mode, it's OK to switch to 180.
     0           5487                  .
   B80           5488 82520006         runnos30 lx      x2,a_cst,discntl .Check if dispat should be called
   B84           5489 9420FF0B                  brxeq   x2,x0,runnos6
     0           5490                  .
     0           5491                  .   Its time to run 180 again. If NOS170 is in 170 job mode or in EI as
     0           5492                  .  a result of a call from job mode, its ok to exit. Otherwise, set 180
     0           5493                  .  priority to a high value and return to NOS170. It should give up control
     0           5494                  .  quickly.
     0           5495                  .
   B88           5496 A8620015                  shfc    x2,x_inmcr,x0,21      .Cant exit if EXCH is set in 170 XP.
   B8C           5497 92020010                  brrgt   x0,x2,runnos35        .Jump if EXCH is set.
   B90           5498 D0820008                  lbyts,1 x2,a_innosx,x0,xpvmid .Get 170 mode VMID.
   B94           5499 828D0005                  lx      xd,a_innosx,xpucr     .Get word that contains monitor flag.
   B98           5500 95200006                  brxne   x2,x0,runnos32         .Jump if in 170 mode.
   B9C           5501 848E001A                  la      ae,a_innosx,2*8+8+2   .get pointer to stack frame save area
   BA0           5502 82ED0005                  lx      xd,ae,xpucr           .Get monitor flag from save area.
   BA4           5503 A8DD001F         runnos32 shfc    xd,xd,x0,31           .Move monitor flag to bit 0.
   BA8           5504 97D0000A                  brxge   xd,x0,runnos50        .Jump if ok to exit from 170.
     0           5505                  .
   BAC           5506 D0510052         runnos35 lbyts,1 x1,a_cst,x0,lpid8     .Cant exit. Raise 180 priority.
   BB0           5507 8D020708                  ente    x2,708(16)
   BB4           5508 D9721096                  sbyts,2 x2,a_dscb,x1,np180pr
   BB8           5509 9400FEF1                  brxeq   x0,x0,runnos6
     0           5510                  .
     0           5511                  .
     0           5512                  .   End of EXCH loop.
     0           5513                  .
   BBC           5514 3D00             runnos50 entp    x0,0                  .Get current time
   BBE           5515 824200A4                  lx      x2,a_root,nostime     .Update total NOS cpu time
   BC2           5516 0801                      cpytx   x1,x0
   BC4           5517 8341008D                  sx      x1,a_root,nosexit
   BC8           5518 2571                      subx    x1,x_infrc
   BCA           5519 2412                      addx    x2,x1
   BCC           5520 D05E0002                  lbyts,1 xe,a_cst,x0,dsprior   .Get 180 priority to determine if idle
   BD0           5521 834200A4                  sx      x2,a_root,nostime
   BD4           5522 95E00007                  brxne   xe,x0,runnos55        .Jump if 180 was not idle
   BD8           5523 824200A5                  lx      x2,a_root,nostime+8   .Get total NOS cpu time.ve_idle
   BDC           5524 2412                      addx    x2,x1

                 0 1 2 3 4 5 6 7  123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
 OFFSET BIT      LINE BINARY           SOURCE STATEMENT                                         CPU ASSEMBLY          LEVEL
   BDE           5525 834200A5                  sx      x2,a_root,nostime+8   .Update total NOS cpu time.ve_idle
   BE2           5526 82510005         runnos55 lx      x1,a_cst,xcbrma
   BE6           5527 3F61                      entl    x0,r_jps              .Reset JPS
   BE8           5528 0F01                      cpyxs   x1,x0
   BEA           5529 2F90             runexit  brdir   a_inret,x0            .Return to where called from

                 0 1 2 3 4 5 6 7  123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
 OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                             CPU ASSEMBLY         LEVEL
      0           5531                     ..............................................................................
      0           5532                     .  EXTERNAL INTERRUPT PROCESSOR
      0           5533                     .          entry conditions:
      0           5534                     .             a_extret - return address
      0           5535                     ..............................................................................
      0           5536                     .
    BEC           5537 8241008B            extrq     lx       x1,a_root,multpro
    BF0           5538 94100032                      brxeq    x1,x0,extrq5          .Jump if not multiprocessor
    BF4           5539 3D02                          entp     x2,0
    BF6           5540 3F00                          entl     x0,tsk_sw
    BF8           5541 8251000A                      lx       x1,a_cst,ext_int
    BFC           5542 A8110000                      shfc     x1,x1,x0,tsk_sw
    C00           5543 97100007                      brxge    x1,x0,extrq1          .Jump if no task switch
    C04           5544 89520050                      sbit     x2,a_cst,ext_int,x0
    C08           5545 3F01                          entl     x0,1
    C0A           5546 D8500030                      sbyts,1  x0,a_cst,x0,caldisp   .Set task switch flag
    C0E           5547 3F01                extrq1    entl     x0,pur_ca
    C10           5548 A8110041                      shfc     x1,x1,x0,pur_ca-tsk_sw+64
    C14           5549 97100008                      brxge    x1,x0,extrq2          .Jump if cache purge not needed
    C18           5550 89520050                      sbit     x2,a_cst,ext_int,x0
    C1C           5551 0820                          cpytx    x0,x2                 .Free running clock
    C1E           5552 0502                          purge    x0,2                  .Purge cache
    C20           5553 8350000B                      sx       x0,a_cst,cachtim
    C24           5554 3F02                extrq2    entl     x0,pur_map
    C26           5555 A8110041                      shfc     x1,x1,x0,pur_map-pur_ca+64
    C2A           5556 97100008                      brxge    x1,x0,extrq3          .Jump if map purge not needed
    C2E           5557 89520050                      sbit     x2,a_cst,ext_int,x0
    C32           5558 0820                          cpytx    x0,x2                 .Free running clock
    C34           5559 050F                          purge    x0,15                 .Purge map
    C36           5560 8350000C                      sx       x0,a_cst,maptim
    C3A           5561 A8110041            extrq3    shfc     x1,x1,x0,step_pr-pur_map+64
    C3E           5562 97100005                      brxge    x1,x0,extrq4          .Jump if no error halt
    C42           5563 3F00                          entl     x0, 1
    C44           5564 D8500030                      sbyts,1  x0,a_cst,x0,caldisp   .Call dispatcher to process STEP
    C48           5565 D0510004            extrq4    lbyts,1  x1,a_cst,x0,memport   .Dont check IO completions if
    C4C           5566 D04204F6                      lbyts,1  x2,a_root,x0,intport  . IOU doesnt send them to this CPU.
    C50           5567 9512001F                      brxne    x1,x2,extrqx
    C54           5568 844E056D            extrq5    la       ae,a_root,pextiou
    C58           5569 82E10000                      lx       x1,ae,0               .Exit if no external interrupts
    C5C           5570 94100009                      brxeq    x1,x0,extrq6          . have been sent by IOU.
    C60           5571 3F01                          entl     x0,1
    C62           5572 83400095                      sx       x0,a_root,eiflag      .Set flag that ext interrupt.
    C66           5573 83400097                      sx       x0,a_root,asyntime
    C6A           5574 D8500034                      sbyts,1  x0,a_cst,x0,asyncp
    C6E           5575 844E0567            extrq6    la       ae,a_root,dpv$scd_block_p
    C72           5576 8EEE0004                      addaq    ae,ae,4
    C76           5577 3D00                          entp     x0,0
    C78           5578 14E1                          lbset    x1,ae,x0
    C7A           5579 9110000A                      brrne    x1,x0,extrqx          .If SCD block not updated
      0           5580                               monreq   ascii_kb
      0           5611                     .
    C8E           5612 2FC0                extrqx    brdir    a_extret,x0


                  0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
 OFFSET BIT       LINE BINARY              SOURCE STATEMENT                                             CPU ASSEMBLY         LEVEL
      0           5614                     ..............................................................................
      0           5615                     .
      0           5616                     .          MTP$IDLE_180  routine to idle 180.
      0           5617                     .
      0           5618                     .          This routine is called to put 180 into an idle state. Only
      0           5619                     .          the system console is kept alive and only the monitor window
      0           5620                     .          will respond to commands. If dual state is present, 180 will idle
      0           5621                     .          and give control to NOS/NOS-BE. Depending on why the system idled,
      0           5622                     .          the system may be able to be resumed via a RESUME_SYSTEM command.
      0           5623                     .
      0           5624                     .               mtp$idle_180 (resume_permitted: boolean)
      0           5625                     .
      0           5626                     ..............................................................................
    C90           5627                               align       0,8
      0           5628                     idle180   ALIAS       MTP$IDLE_180
      0           5629                     idle180   procedur
      0           5640                     idleres   param       val,subrange,1
      0           5700                     .
      0           5701                               ploadx      x_resume,idleres             .Load RESUME_ALLOWED - A4
      0           5722                     .                                                   gets clobbered later.
    C94           5723 84340572                      la          a_root,a_bindin,bs_root
      0           5724                     .
    C98           5725 84470470                      la          a_dscb,a_root,nostab
    C9C           5726 8E000020                      addaq       a0,a0,mstkfram
    CA0           5727 3FC9                          entl        x0,r_pit                     .Save PIT - dont charge current task
    CA2           5728 0E04                          cpysx       x_clock,x0                   . for idle time.
    CA4           5729 8D000047                      ente        x0,r_bc
    CA8           5730 0E01                          cpysx       x1,x0                        .Get  base constant.
    CAA           5731 0B42                          cpyax       x2,a_root
    CAC           5732 2421                          addx        x1,x2                        .Form pointer to cst
    CAE           5733 0A15                          cpyxa       a_cst,x1
    CB0           5734 8515000A                      sa          a_cst,a_csf,10               .Save CST_P in p-list.
      0           5735                     .
    CB4           5736 3FC2                          entl        x0,r_te                      .Enable traps in case we got here via
    CB6           5737 0F00                          cpyxs       x0,x0                        . trap handler.
      0           5738                     .
    CB8           5739 3D0E                i180a     entp        xe,0                         .Set 180 priority to 0.
    CBA           5740 8F090004                      addpxq      a_inret,x0,i180c             .Run NOS170 if it is present. (If not
    CBE           5741 9000FE55                      brreq       x0,x0,run_nos                .  present RUNNOS returns immediately.)
    CC2           5742 0000000000000000 i180c        bss         0
    CC2           5743 3D00                          entp        x0,0                         .Set lock for calling
    CC4           5744 8E4F054B                      addaq       af,a_root,asylocki           .  mtp$monitor_system_status.
    CC8           5745 14F1                          lbset       x1,af,x0
    CCA           5746 92100019                      brrgt       x1,x0,i180f                  .Jump if already locked.
    CCE           5747 844E054F                      la          ae,a_root,mtvdftb            .Fetch pointer to DFT block.
    CD2           5748 82E10000                      lx          x1,ae,dftcw                  .Get DFT control word.
    CD6           5749 A911003E                      shfx        x1,x1,x0,62                  .Check E8 field.
    CDA           5750 97100008                      brxge       x1,x0,i180e                  .Jump if not set.
    CDE           5751 8D0000FF                      ente        x0,00ff(16)
    CE2           5752 8E3E0420                      addaq       ae,a_bindin,16*proc_dft      .Set up call to dsp$process_dft_block.
    CE6           5753 B5E00000                      callseg     bs_rqtbl,ae,a0               .Call dsp$process_dft_block.
    CEA           5754 0000000000000000 i180e        bss         0
    CEA           5755 8D0000FF                      ente        x0,00ff(16)
    CEE           5756 8E3E0350                      addaq       ae,a_bindin,16*mon_smu

                  0 1 2 3 4 5 6 7   12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY                    LEVEL
   CF2          5757 B5E00000                  callseg  bs_rqtbl,ae,a0          .Call mtp$monitor_system_status.
   CF6          5758 3D00                      entp     x0,0                    .Clear call environment.
   CF8          5759 D8F00000                  sbyts,1  x0,af,x0,0              .Clear lock.
   CFC          5760 0000000000000000 i180f    bss      0
   CFC          5761 9F200002                  brcr     2,0,i180g               .Clear shortwarning from MCR.
   D00          5762 D0410032          i180g   lbyts,1   x1,a_root,x0,scb+scbstepr .Loop if STEP still requested.
   D04          5763 9110FFDA                  brrne    x1,x0,i180a
   D08          5764 9480FFD8                  brxeq    x_resume,x0,i180a       .Loop if resume not permitted.
     0          5765                           .
   D0C          5766 3FC9                      entl     x0,r_pit                .Restore PIT.
   D0E          5767 0F04                      cpyxs    x_clock,x0
   D10          5768 0400                      return
```

```
                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                        CPU ASSEMBLY                    LEVEL
     0          5770                           ..........................................................................
     0          5771                           .
     0          5772                           .   This routine is called from cybil to send interrupts to other processors.
     0          5773                           .
     0          5774                           .       PROCEDURE [XREF] mtp$interrupt_processor (port_mask: 0..255)
     0          5775                           .
     0          5776                           ..........................................................................
     0          5777                           .
     0          5778                   int     alias    MTP$INTERRUPT_PROCESSOR
     0          5779                   int     procedur
     0          5790                   intmask param    val,subrange,1
     0          5850                           ploadx   x2,intmask
   D1C          5871 0302                      intrupt  x2,0
   D1E          5872 0400                      return
```

```
                0 1 2 3 4 5 6 7  1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
OFFSET BIT    LINE BINARY            SOURCE STATEMENT                                              CPU ASSEMBLY          LEVEL
   0          5874                   ..........................................................................
   0          5875                   .
   0          5876                   .   MTP$SPIN_CPU
   0          5877                   .       Routine to make a CPU spin in a tight loop indefinitely.
   0          5878                   .
   0          5879                   .   This routine is called by a CPU which is about to be deconfigured
   0          5880                   .   out of the system due to hardware errors or an operator request.
   0          5881                   .   The CPU spins in a very tight loop, only checking whether it should
   0          5882                   .   continue to spin.  The intent is to have the CPU executing as little
   0          5883                   .   as possible before it is completely removed from the system
   0          5884                   .   configuration.  The CPU is expected to be executing this portion of
   0          5885                   .   code when Dedicated Fault Tolerance (DFT) stops a CPU which has been
   0          5886                   .   operational.  The boolean indicating whether or not the CPU should
   0          5887                   .   continue to spin will be changed asynchronously by another CPU.
   0          5888                   .
   0          5889                   .   PROCEDURE mtp$spin_cpu;
   0          5890                   .
   0          5891                   ..........................................................................
 D20          5892                            align     0,8
   0          5893                   spin_cpu ALIAS    mtp$spin_cpu
   0          5894                   spin_cpu procedur
   0          5905                   cpu_id   param    val,subrange,1
   0          5965                   .
 D20          5966 84340572                   la        a_root,a_bindin,bs_root
   0          5967                   .
 D24          5968 8D000047                   ente      x0,r_bc
 D28          5969 0E01                       cpysx     x1,x0                 .Get the base constant
 D2A          5970 0B42                       cpyax     x2,a_root
 D2C          5971 2421                       addx      x1,x2                 .Form a pointer to the CST
 D2E          5972 0A15                       cpyxa     a_cst,x1
   0          5973                   .
   0          5974                   . Keep checking to see if this CPU should continue to spin, during which time it
   0          5975                   . performs no useful system operations except purging its cache and maps.
   0          5976                   . Because traps are disabled at this point, this CPU should never come out of this
   0          5977                   . loop until the cpu_spin field is changed to TRUE.  Generally, the CPU will be
   0          5978                   . downed in the MRT after this point and therfore will never return to be used
   0          5979                   . in the system.
   0          5980                   .
 D30          5981 0502              spin_1   purge     x0,2                  .purge cache
 D32          5982 050F                       purge     x0,15                 .purge map
 D34          5983 D05100FA                   lbyts,1 x1,a_cst,x0,cpu_spin    .Get value of spin boolean in CST
 D38          5984 9110FFFC                   brrne     x1,x0,spin_1          .If spin is still required
   0          5985                   .
   0          5986                   . If the CPU reaches the following statement, it is no longer required to spin
   0          5987                   . indefinitely.  It may, in fact, be restarted by DFT.
   0          5988                   .
 D3C          5989 3D07                       entp      x_infrc,0             .Get current time
 D3E          5990 0877                       cpytx     x_infrc,x_infrc
 D40          5991 8351000B                   sx        x1,a_cst,cachtim      .Store time of last purge for cache
 D44          5992 8351000C                   sx        x1,a_cst,maptim       .Store time of last purge for maps
 D48          5993 3FC9                       entl      x0,r_pit              .Restore the PIT
 D4A          5994 0F04                       cpyxs     x_clock,x0
 D4C          5995 0400                       return

                   0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```



```
OFFSET BIT    LINE BINARY            SOURCE STATEMENT                                              CPU ASSEMBLY          LEVEL
   0          5997                   ...............................................................................
   0          5998                   .
   0          5999                   . The following is the definition of the oss$mainframe_wired_cb section.
   0          6000                   . It will ALWAYS be cache bypass
   0          6001                   ...............................................................................
   0          6002                   .
   0          6003                   oss$mainframe_wired_cb  SECTION working,read+write
   0          6004                            use      oss$mainframe_wired_cb
   0          6005 FFFF80000000000 osv$mainframe_wired_cb_heap  vfd,16,32,64 0ffff(16),080000000(16),0 .Pointer to heap
   0          6006 000000000000
   0          6007                            defg     osv$mainframe_wired_cb_heap
   0          6008                   .
   0          6009                   .        The following is the definition of the communication block to
   0          6010                   .        talk to the NOS/VE ascii console.
   0          6011                   .
  10          6012                            align    0,8
  10          6013 0000000000000000 asciiblk bss      0                    .ascii console communications block
  10          6014 00                         vfd,8    0                    .input buffer id
  11          6015 000000                     vfd,8,8,8 0,0,0               .character buffer
  14          6016 00000000                   vfd,32   0                    .rma of last output entry processed
  18          6017 00                         vfd,8    0                    .console driver command
  19          6018 00                         vfd,8    0                    .hold display flag
  1A          6019 00                         vfd,8    0                    .echo line size
  1B          6020 00                         vfd,8    0                    .undefined
  1C          6021 00000000                   vfd,32   0                    .rma of output list
   0          6022                   .
  20          6023                            align    0,8
  20          6024 0000000000000001 extiou    vfd,64   1                    .IOU sets this word non-zero when
   0          6025                                                          . sending external interrupt.
  28          6026 0000000000000000 dpv$scd_time  vfd,64 0
   0          6027                   .
  30          6028                            align    0,16
  30          6029 00000000000001A0 a170_xp   bssz     xpsize
 1D0          6030 0000000000000098 a170_st   bssz     a170_stl*8
   0          6031                   .
   0          6032                   .       Set up the NOS XP.
   0          6033                   .
   0          6034                   .  Initialize the NOS170 Exchange Package
   0          6035                   .
   0          6036                   .
   0          6037                            ref      mtp$170_trap_handler
 268          6038 0000000000000000 a170xpin bss      0
   0          6039                            xpa      a170_xp,2,mtp$170_trap_handler
   0          6050                            xpareg   a170_xp,a_tos,nil
   0          6061                            xpareg   a170_xp,a_csf,nil
   0          6072                            xpareg   a170_xp,a_psa,nil
   0          6083                            xpv      a170_xp,a_bindin*8+10,01000(16)+snsf170,16
   0          6089                            xpareg   a170_xp,a_plist,nil
   0          6100                            xpareg   a170_xp,5,nil
   0          6111                            xpareg   a170_xp,6,nil
   0          6122                            xpareg   a170_xp,7,nil
   0          6133                            xpareg   a170_xp,8,nil
   0          6144                            xpareg   a170_xp,9,nil
   0          6155                            xpareg   a170_xp,10,nil

                   0 1 2 3 4 5 6 7   1234567890123456789012345678901234567890123456789012345678901234567890
```

```
 OFFSET BIT      LINE BINARY            SOURCE STATEMENT                                          CPU ASSEMBLY                LEVEL
       0         6166                           xpareg  a170_xp,11,nil
       0         6177                           xpareg  a170_xp,12,nil
       0         6188                           xpareg  a170_xp,13,nil
       0         6199                           xpareg  a170_xp,14,nil
       0         6210                           xpareg  a170_xp,15,nil
       0         6221                           xpv     a170_xp,xpstl,a170_stl,16
       0         6227                           xpv     a170_xp,xpmm,0fbfc(16),16
       0         6233                           xpv     a170_xp,xpum,0ff7f(16),16
       0         6239                           xpv     a170_xp,xpkm,0ffff(16),16
       0         6245                           xpv     a170_xp,xppit,0000f(16),16
       0         6251                           xpv     a170_xp,xppit+8,04240(16),16
       0         6257                           xpv     a170_xp,xplrn,1,16
       0         6263                           xpv     a170_xp,xpflgte,00002(16),16
       0         6269
     1E8         6270 00000000000001E8          org     a170_st+snnos170*8
     1E8         6271 9A11FFFF00000000          vfd,64  09a11ffff00000000(16) .STE for NOS
     1F0         6272 00000000000001F0          org     a170_st+snsf170*8
     1F0         6273 0000000000000000          vfd,64  0000000000000000(16) .STE for NOS stack
     1F8         6274 00000000000001F8          org       a170_st+snnth170*8
     1F8         6275 BE11800100000000          vfd,64  0be118001000000000(16) .STE for NOS trap handler
     268         6276 0000000000000268          org     a170xpin
       0         6277                           end     begin

 **** NO DIAGNOSTICS
```

0 1 2 3 4 5 6 7 12345678901234567890123456789012345678901234567890123456789012345678901234567890


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                           ON LINE                               sec+off
A170XPIN                       5997        0 bit   LABEL       OSS$MAINFRAME_WIRED_CB+268

                                             REFS:   6235
A170_ST                        5989     1216 bit   LABEL       OSS$MAINFRAME_WIRED_CB+1D0

                                             REFS:   1243/P   6229      6231      6233
A170_STL                         63       19       CONSTANT
                                             REFS:   5989     6180/P
A170_XP                        5988     3328 bit   LABEL       OSS$MAINFRAME_WIRED_CB+30

                                             REFS:   1242/P   3282      3289      3299      3309      3318      3324      3334
                                                     3344     3354      3364      3374      3384      3394      3404      3414
                                                     3424     3434      3443      3448      3453      3458      3463      3468
                                                     3473     3478      5998/P    6009/P    6020/P    6031/P    6042/P    6048/P
                                                     6059/P   6070/P    6081/P    6092/P    6103/P    6114/P    6125/P    6136/P
                                                     6147/P   6158/P    6169/P    6180/P    6186/P    6192/P    6198/P    6204/P
                                                     6210/P   6216/P    6222/P
AJLLEN                           44      255       CONSTANT
                                             REFS:   3454     3461      3668/P
AJL0                            368       19       CONSTANT
                                             UNREFERENCED
AREG                            625                PROCEDURE
                                             REFS:    632/P    638/P     644/P     650/P     656/P     662/P     882/P     889/P
                                                      895/P    901/P     936/P     942/P     965/P     972/P     978/P     984/P
ASCIIBLK                       5972        0 bit   LABEL       OSS$MAINFRAME_WIRED_CB+10

                                             REFS:   1303/P
ASCII_KB                       3326       56       CONSTANT
                                             REFS:   4024/P   5561/P
ASTRING                           1                DEF
                                             REFS:   3108     3175      3180      3257      3272
ASYLOCK                        1294        8 bit   LABEL       OSS$MAINFRAME_WIRED+54A

                                             REFS:   3971/P   4203/P
ASYLOCKI                       1295        8 bit   LABEL       OSS$MAINFRAME_WIRED+54B

                                             REFS:   5718/P
ASYNC                          3965       16 bit   LABEL       CODE+1E8  ENTRY_POINT
                                             REFS:   3370     3439      4204/P    4317/P    4327/P    4373/P
ASYNC12                        4096       32 bit   LABEL       CODE+288
                                             REFS:   4061/P
ASYNC15                        4132       32 bit   LABEL       CODE+2A4
                                             REFS:   4097/P
ASYNC20                        4168       32 bit   LABEL       CODE+2C0
                                             REFS:   4133/P
ASYNC50                        4202       16 bit   LABEL       CODE+2D8
                                             REFS:   4169/P
ASYNC6                         4021       32 bit   LABEL       CODE+244
                                             REFS:   3980/P
ASYNC8                         4058       32 bit   LABEL       CODE+264
                                             REFS:   4023/P
ASYNC90                        4377       16 bit   LABEL       CODE+4A8
                                             REFS:   4302/P   4308/P    4314/P    4325/P

***  REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter
```

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
ASYNCP	400	52		CONSTANT		
				REFS:	3968/P	5555/P
ASYNINC	1265	64	bit	LABEL	OSS$MAINFRAME_WIRED+4C0	
				REFS:	3391	3975/P
ASYNTIME	1264	64	bit	LABEL	OSS$MAINFRAME_WIRED+4B8	
						ENTRY_POINT
				REFS:	3366	3366 3390 3967/P 3979/P 5554/P
A_BINDIN	93	3		CONSTANT		
				REFS:	94	656/P 2094 2300/P 2330/P 2360/P 2390/P 2420/P
					2450/P	2561/P 2591/P 2621/P 2651/P 2685/P 2694/P 2711/P
					2768/P	2798/P 2827/P 2842/P 2912/P 2942/P 2972/P 3002/P
					3032/P	3318 3519/P 4228/P 4339/P 4773/P 4896/P 4910/P
					5697/P	5726/P 5730/P 5926/P 6042/P
A_CSF	83	1		CONSTANT		
				REFS:	84	644/P 2074 3299 3497/P 3789/P 4231/P 4338/P
					4532/P	4533/P 4680/P 4781/P 4858/P 4859/P 4905/P 4906/P
					4907/P	4908/P 4911/P 4912/P 5075/P 5242/P 5243/P 5369/P
					5371/P	5372/P 5373/P 5418/P 5419/P 5420/P 5421/P 5454/P
					5455/P	5708/P 6020/P
A_CST	108	5		CONSTANT		
				REFS:	109	889/P 2472/P 2499/P 2526/P 2733/P 2850/P 2877/P
					3788/P	3789/P 3792/P 3794/P 3795/P 3822/P 3824/P 3828/P
					3842/P	3849/P 3905/P 3931/P 3939/P 3947/P 3968/P 3969/P
					4210/P	4211/P 4218/P 4223/P 4226/P 4246/P 4247/P 4253/P
					4254/P	4263/P 4265/P 4300/P 4303/P 4304/P 4316/P 4318/P
					4326/P	4335/P 4347/P 4352/P 4365/P 4394/P 4396/P 4479/P
					4591/P	4779/P 4781/P 4912/P 4933/P 4961/P 4966/P 5020/P
					5022/P	5072/P 5107/P 5108/P 5132/P 5139/P 5363/P 5454/P
					5461/P	5462/P 5469/P 5487/P 5501/P 5507/P 5522/P 5525/P
					5527/P	5531/P 5534/P 5538/P 5541/P 5545/P 5546/P 5555/P
					5707/P	5708/P 5932/P 5943/P 5951/P 5952/P
A_DSCB	118	7		CONSTANT		
				REFS:	119	901/P 3809/P 3810/P 3830/P 3852/P 3858/P 3871/P
					3873/P	3876/P 3880/P 3886/P 3917/P 3937/P 4305/P 4774/P
					5140/P	5148/P 5150/P 5463/P 5489/P 5699/P
A_DSP	78	0		CONSTANT		
				REFS:	79	638/P
A_EXTRET	173	12		CONSTANT		
				REFS:	174	978/P 4345/P 4580/P 4953/P 5329/P 5593/P
A_INNOSX	143	8		CONSTANT		
				REFS:	144	936/P 5137/P 5187/P 5192/P 5195/P 5229/P 5243/P
					5277/P	5338/P 5348/P 5353/P 5409/P 5411/P 5479/P 5480/P
					5482/P	
A_INRET	148	9		CONSTANT		
				REFS:	149	942/P 4320/P 4478/P 4590/P 4967/P 5510/P 5714/P
A_PLIST	98	4		CONSTANT		
				REFS:	99	662/P 3129/P 3211/P 3324 6048/P
A_PSA	88	2		CONSTANT		
				REFS:	89	650/P 2081 3309 3508/P 4783/P 4919/P 4926/P
					4973/P	6031/P
A_ROOT	103	4		CONSTANT		
				REFS:	104	882/P 2104 2299/P 2329/P 2359/P 2389/P 2419/P

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
					2449/P	2560/P 2590/P 2620/P 2650/P 2687/P 2693/P 2710/P
					2767/P	2797/P 2820/P 2835/P 2911/P 2941/P 2971/P 3001/P
					3031/P	3530/P 3786/P 3796/P 3801/P 3807/P 3809/P 3812/P
					3813/P	3818/P 3829/P 3830/P 3832/P 3834/P 3850/P 3851/P
					3861/P	3862/P 3864/P 3865/P 3880/P 3881/P 3883/P 3888/P
					3898/P	3899/P 3900/P 3901/P 3904/P 3906/P 3911/P 3912/P
					3919/P	3922/P 3925/P 3926/P 3937/P 3943/P 3967/P 3971/P
					3975/P	3976/P 3977/P 3979/P 3981/P 3982/P 3985/P 4021/P
					4058/P	4096/P 4132/P 4168/P 4203/P 4227/P 4357/P 4367/P
					4476/P	4588/P 4773/P 4774/P 4777/P 4958/P 4960/P 5002/P
					5008/P	5061/P 5062/P 5066/P 5068/P 5106/P 5110/P 5137/P
					5145/P	5459/P 5496/P 5498/P 5502/P 5504/P 5506/P 5518/P
					5547/P	5549/P 5553/P 5554/P 5556/P 5697/P 5699/P 5705/P
					5718/P	5721/P 5736/P 5926/P 5930/P
A_RQTBL	163	10		CONSTANT		
				REFS:	164	965/P 2299/P 2329/P 2359/P 2389/P 2419/P 2449/P
					2560/P	2590/P 2620/P 2650/P 2687/P 2688/P 2690/P 2691/P
					2693/P	2710/P 2767/P 2797/P 2911/P 2941/P 2971/P 3001/P
					3031/P	4227/P 4234/P 4235/P 4240/P 4248/P 5006/P 5079/P
					5080/P	5083/P 5089/P
A_RQ_RET	168	11		CONSTANT		
				REFS:	169	972/P 2293/P 2323/P 2353/P 2383/P 2413/P 2443/P
					2554/P	2584/P 2617/P 2647/P 2677/P 2707/P 2761/P 2791/P
					2905/P	2935/P 2968/P 2995/P 3025/P 5094/P
A_SITRET	178	13		CONSTANT		
				REFS:	179	984/P 4349/P 4577/P 4950/P 5114/P 5323/P
A_TOS	73	0		CONSTANT		
				REFS:	74	632/P 2064 3289 3486/P 6009/P
A_XCB	113	6		CONSTANT		
				REFS:	114	895/P 3805/P 3822/P 3823/P 3923/P 3924/P 4211/P
					4213/P	4216/P 4217/P 4247/P 4255/P 4258/P 4259/P 4264/P
					4324/P	4397/P 4435/P 4533/P 4584/P 4677/P 4678/P 4679/P
					4721/P	4725/P 4933/P 4934/P 4936/P 4940/P
BCRIT1	4392	0	bit	LABEL	CODE+4AC	
				REFS:	4923/P	
BEGIN	1			DEF		
				REFS:	3427	
BEGIN	3780	0	bit	LABEL	CODE+0	ENTRY_POINT
				REFS:	2040	2055 3464/P 3779
BEGIN22	3943	32	bit	LABEL	CODE+1D2	
				REFS:	3932/P	
BEGIN2_5	3848	0	bit	LABEL	CODE+B8	
				REFS:	3843/P	3846/P
BEGIN4	3922	32	bit	LABEL	CODE+188	
				REFS:	3889/P	3920/P
BEGIN5	3936	0	bit	LABEL	CODE+1C8	
				REFS:	3797/P	
BGNSTAK	3459	0	bit	LABEL	MTS$MONITOR_STACK+0	
				REFS:	3662/P	
BINDING	1	1456	byte	SECTION	+0	
				REFS:	331	331 351 351 371 371 391 391

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                            sec+off
                                                   411     411     431     431     451     451     471     471
                                                   491     491     511     511     531     531     551     551
                                                   571     571     591     591     611     611     631     631
                                                   651     651     671     671     691     691     711     711
                                                   731     731     751     751     771     771     791     791
                                                   811     811     831     831     851     851     871     871
                                                   891     891     911     911     931     931     951     951
                                                   971     971     991     991    1011    1011    1031    1031
                                                  1051    1051    1071    1071    1091    1091    1111    1111
                                                  1131    1131    1151    1151    1171    1171    1191    1192
                                                  1211    1212    1231    1232    1251    1253    1271    1274
                                                  1291    1295    1311    1316    1331    1337    1351    1357
                                                  1371    1386    1391    1406    1411    1426    1431    1446
                                                  1451    1453    1466    1468    1471    1486    1491    1506
                                                  1511    1526    1531    1546    1551    1566    1571    1586
                                                  1591    1606    1611    1626    1631    1646    1651    1666
                                                  1671    1686    1691    1706    1711    1726    1731    1746
                                                  1751    1766    1771    1786    1791    1806    1811    1826
                                                  1831    1846    1851    1866    1871    1886    1891    1906
                                                  1911    1926    1931    1946    1951    1966    1971    1986
                                                  1991    2006    2011    2026    2031    2046    3754    3769
BINDSEC                          1455       0 bit   LABEL        BINDING+0ENTRY_POINT
                                            REFS:   1454    2090    2105    3428    3519/P
BOOLEAN                             1               DEF
                                            REFS:   3091    3158    3173    3240    3255
BS_ERRST                         3766     128 bit   LABEL       BINDING+578
                                            REFS:   2827/P  2842/P
BS_MERRS                         3767     128 bit   LABEL       BINDING+588
                                            REFS:   4896/P
BS_PGFLT                         3768     128 bit   LABEL       BINDING+598
                                            REFS:   4910/P
BS_PTLOK                         3769      64 bit   LABEL       BINDING+5AA
                                            UNREFERENCED
BS_ROOT                          3759      64 bit   LABEL       BINDING+572
                                            REFS:   4773/P  5697/P  5926/P
BS_RQTBL                         1456       0 bit   LABEL        BINDING+0
                                            REFS:   4233/P  4340/P  5076/P  5727/P  5731/P
BS_TRAP                          3758     128 bit   LABEL       BINDING+560ENTRY_POINT
                                            REFS:   2260    2275    3710/P  3756    3757

C170_DUE                          608      33       CONSTANT
                                            UNREFERENCED
C180_DUE                          609      36       CONSTANT
                                            UNREFERENCED
CACHTIM                           381      88       CONSTANT
                                            REFS:    216     260    3794/P  5534/P  5951/P
CALDISP                           398      48       CONSTANT
                                            REFS:   3939/P  4352/P  4365/P  5108/P  5363/P  5527/P  5545/P
CKASYNC                          4570      32 bit   LABEL       CODE+5A0
                                            REFS:   4492/P  4528/P
CKASYNCX                         4595      16 bit   LABEL       CODE+5F8
                                            REFS:   4586/P
CKDUE                            4527      32 bit   LABEL       CODE+576
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                            sec+off
                                            REFS:   4494/P
CKEXCH                           4582      32 bit   LABEL       CODE+5CA
                                            REFS:   4580/P
CKEXCH5                          4593      16 bit   LABEL       CODE+5F4
                                            REFS:   4590/P
CKEXSP5                          4481      16 bit   LABEL       CODE+548
                                            REFS:   4478/P
CKEXTINT                         4579      32 bit   LABEL       CODE+5BE
                                            REFS:   4576/P  4577/P
CKHDW                            4490      32 bit   LABEL       CODE+554
                                            REFS:   4473/P
CKMCALL                          4675      32 bit   LABEL       CODE+634
                                            REFS:   4640/P
CKMCALL5                         4684      32 bit   LABEL       CODE+656
                                            REFS:   4682/P
CKPF                             4639      32 bit   LABEL       CODE+61C
                                            REFS:   4603/P
CKUCR                            4721      32 bit   LABEL       CODE+686
                                            REFS:   4676/P
CKUSER                           4601      32 bit   LABEL       CODE+600
                                            REFS:   4572/P
CMP$MONITOR_ROUTINES             1253               LABEL                  EXTERNAL
                                            REFS:   1252    2448/P
CODE                                1    3406 byte  SECTION        +0
                                            REFS:   3778    3793
COR_FRC                          1319      48 bit   LABEL       OSS$MAINFRAME_WIRED+618

                                            REFS:   3900/P
CPPREPRO                          391     249       CONSTANT
                                            UNREFERENCED
CPSTREAS                          390     248       CONSTANT
                                            UNREFERENCED
CPTIME                            375      56       CONSTANT
                                            REFS:    212     256    4246/P
CPUDOWN                          1363     248 bit   LABEL       OSS$MAINFRAME_WIRED+2206

                                            REFS:   5041/P
CPUSPOSS                         1234       8 bit   LABEL       OSS$MAINFRAME_WIRED+451
                                                                           ENTRY_POINT
                                            REFS:   3360    3393
CPUS_ON                          1285       8 bit   LABEL       OSS$MAINFRAME_WIRED+4FA
                                                                           ENTRY_POINT
                                            REFS:   3372    3385
CPU_ID                              1       7       VARIABLE
                                            UNREFERENCED
CPU_ID                           3273       7       VARIABLE
                                            REFS:   5869/P
CPU_SPIN                          392     250       CONSTANT
                                            REFS:   5943/P
CPU_STAT                          364       6       CONSTANT
                                            REFS:    195     239    3947/P
CPWELL                            366       8       CONSTANT
                                            REFS:   3969/P  5107/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                            sec+off
CP_CURST                          427        0          CONSTANT
                                                REFS:     206        250
CP_NXTST                          428        1          CONSTANT
                                                REFS:     208        252       5022/P
CP_STATE                          369       30          CONSTANT
                                                REFS:     206        208        250       252       5022/P
CSTO                              183        0 bit     LABEL        OSS$MAINFRAME_WIRED+220
                                                                      ENTRY_POINT
                                                REFS:    1090/P     3353       3423      3727/P    3733/P
CSTHALT                          1361      248 bit     LABEL        OSS$MAINFRAME_WIRED+21C8

                                               UNREFERENCED
CSTSIZE                           355      272          CONSTANT
                                                REFS:     187        226        231       270       3739/P

D7CM                               23       56          CONSTANT
                                                REFS:     500/P     3858/P
D7JP                                8        8          CONSTANT
                                                REFS:     482/P     578
D7RS                               18       32          CONSTANT
                                                REFS:     494/P
D7ST                               13       24          CONSTANT
                                                REFS:     488/P
D7SV                               28       72          CONSTANT
                                                REFS:     506/P
D7TY                                3        0          CONSTANT
                                                REFS:     476/P     577       3886/P    3917/P
D8DS                               53      168          CONSTANT
                                                REFS:     537/P     568        569       570       571       572       573
D8JP                               43      144          CONSTANT
                                                REFS:     525/P     579
D8ST                               48      160          CONSTANT
                                                REFS:     531/P
D8SV                               58      192          CONSTANT
                                                REFS:     543/P
D8TM                               38      128          CONSTANT
                                                REFS:     519/P     580
D8TY                               33      120          CONSTANT
                                                REFS:     513/P     3852/P
DEBUGO                           1308     1024 bit     LABEL        OSS$MAINFRAME_WIRED+588
                                                                      ENTRY_POINT
                                                REFS:    3364       3395
DEFCST                           1002                  PROCEDURE
                                                REFS:    1090/P
DFCM                               68      280          CONSTANT
                                                REFS:     556/P
DFP$MTR_FILE_SERVER_REQUEST      1793                  LABEL                  EXTERNAL
                                                REFS:    1792       3015/P
DFTCW                             676        0          CONSTANT
                                                REFS:    4059/P     5722/P
DFT_PSR                           588      739          CONSTANT
                                                REFS:     596
DISCNTL                           374       48          CONSTANT
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                            sec+off
                                                REFS:     398        400       4210/P    4253/P    4316/P    4326/P    4347/P    4394/P
                                                           5469/P
DMP$APPLY_MAT_CHANGES            1193                  LABEL                  EXTERNAL
                                                REFS:    1192       2385/P
DMP$MTR_ALLOCATE_FRONT_END       1153                  LABEL                  EXTERNAL
                                                REFS:    1152       2343/P
DMP$MTR_DEALLOCATE_FRONT_END     1173                  LABEL                  EXTERNAL
                                                REFS:    1172       2364/P
DMP$MTR_REALLOCATE_FRONT_END     1853                  LABEL                  EXTERNAL
                                                REFS:    1852       3078/P
DONTHING                          789        3          CONSTANT
                                                REFS:    5360/P
DOWN                              424        2          CONSTANT
                                               UNREFERENCED
DPINT                             395      256          CONSTANT
                                                REFS:     222        266
DPP$DISPLAY_REQUEST              1433                  LABEL                  EXTERNAL
                                                REFS:    1432       2637/P
DPP$PROCESS_SCD_BLOCK            1453                  LABEL                  EXTERNAL
                                                REFS:    1452       2658/P
DPV$SCD_BLOCK_P                  1303       48 bit     LABEL        OSS$MAINFRAME_WIRED+567
                                                                      ENTRY_POINT
                                                REFS:    3342       5556/P
DPV$SCD_TIME                     5985       64 bit     LABEL        OSS$MAINFRAME_WIRED_CB+28
                                                                      ENTRY_POINT
                                                REFS:    1305/P     3344
DSCBL                             563      368          CONSTANT
                                                REFS:     581
DSCBLN                            581      368          CONSTANT
                                               UNREFERENCED
DSCBW                             469                   PROCEDURE
                                                REFS:     476/P     482/P     488/P     494/P     500/P     506/P     513/P     519/P
                                                           525/P     531/P     537/P     543/P     550/P     556/P
DSCB_NXT                          473        0          VARIABLE
                                                REFS:       3         4         4         8         9         9        13        14
                                                           14        18        19        19        23        24        24        28
                                                           29        29        33        34        34        38        39        39
                                                           43        44        44        48        49        49        53        54
                                                           54        58        59        59        63        64        64        68
                                                           69        69       563
DSCM                               63      240          CONSTANT
                                                REFS:     550/P     3871/P    3873/P    3876/P
DSP$ACCESS_LOGGING_DATA          1633                  LABEL                  EXTERNAL
                                                REFS:    1632       2847/P
DSP$ISSUE_DFT_REQUEST            1313                  LABEL                  EXTERNAL
                                                REFS:    1312       2511/P
DSP$MTR_MANAGE_SYSTEM_DS_STATUS  1933                  LABEL                  EXTERNAL
                                                REFS:    1932       3162/P
DSP$PROCESS_DFT_ENTRY            1653                  LABEL                  EXTERNAL
                                                REFS:    1652       2868/P
DSPRIOR                           361        2          CONSTANT
                                                REFS:    4304/P     4479/P    4591/P    4961/P    5501/P
DSV$SSR_SDTE                     1341       64 bit     LABEL        OSS$MAINFRAME_WIRED+648
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=wrfte, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
                                                                          ENTRY_POINT
                                               REFS:    1335    3818/P
DS_FLAG                             569      172        CONSTANT
                                               UNREFERENCED
DS_STAT                             568      168        CONSTANT
                                               UNREFERENCED
DTRACE                             1352    16512 bit    LABEL    OSS$MAINFRAME_WIRED+1678
                                                                          ENTRY_POINT
                                               REFS:    3349    3416
DUALSTAT                            370       20        CONSTANT
                                               REFS:    3905/P  4300/P    4966/P    5132/P    5461/P
DUMMY4                              396      264        CONSTANT
                                               UNREFERENCED

ECRIT1                             4428       16 bit    LABEL    CODE+4EC
                                               REFS:    4920/P
EIFLAG                             1255       64 bit    LABEL    OSS$MAINFRAME_WIRED+4A8
                                                                          ENTRY_POINT
                                               REFS:    3361    3388      3976/P    3985/P    5553/P
EIINC                              1259       64 bit    LABEL    OSS$MAINFRAME_WIRED+4B0

                                               REFS:    3981/P
ELEM_ID                             384      112        CONSTANT
                                               REFS:    3792/P
ENDTBLS                            1306      112 bit    LABEL    OSS$MAINFRAME_WIRED+579
                                                                          ENTRY_POINT
                                               REFS:    3353    3419
ERRSTOP                             858                 PROCEDURE
                                               REFS:    5024/P  5041/P
EXCHLOOP                           4393        0 bit    LABEL    CODE+4AC  ENTRY_POINT
                                               REFS:    2613    2616      2628      2643      2646      2658      2673      2676
                                                        2688    2703      2706      2718      3370      3441      4485/P    4597/P
                                                        4605/P  4641/P    4685/P    4724/P    4728/P    4729/P
EXTIOU                             5983       64 bit    LABEL    OSS$MAINFRAME_WIRED_CB+20
                                                                          ENTRY_POINT
                                               REFS:    1304/P  3345      3409
EXTRQ                              5518       32 bit    LABEL    CODE+BEC  ENTRY_POINT
                                               REFS:    3370    3445      4346/P    4581/P    4954/P    5330/P
EXTRQ1                             5528       16 bit    LABEL    CODE+C0E
                                               REFS:    5524/P
EXTRQ2                             5535       16 bit    LABEL    CODE+C24
                                               REFS:    5530/P
EXTRQ3                             5542       32 bit    LABEL    CODE+C3A
                                               REFS:    5537/P
EXTRQ4                             5546       32 bit    LABEL    CODE+C48
                                               REFS:    5543/P
EXTRQ5                             5549       32 bit    LABEL    CODE+C54
                                               REFS:    5519/P
EXTRQ6                             5556       32 bit    LABEL    CODE+C6E
                                               REFS:    5551/P
EXTRQX                             5593       16 bit    LABEL    CODE+C8E
                                               REFS:    5548/P  5560/P
EXT_INT                             378       80        CONSTANT
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
                                               REFS:    5522/P  5525/P    5531/P    5538/P

FILL                                359        0        CONSTANT
                                               UNREFERENCED
FLTINJ                             1298        8 bit    LABEL    OSS$MAINFRAME_WIRED+54E
                                                                          ENTRY_POINT
                                               REFS:    3341    3381
FRC_P                              1246       64 bit    LABEL    OSS$MAINFRAME_WIRED+492

                                               REFS:    4357/P
FRETURNX                            179                 PROCEDURE
                                               UNREFERENCED
FUNCTION                            151                 PROCEDURE
                                               UNREFERENCED

HALTRING                           1292        8 bit    LABEL    OSS$MAINFRAME_WIRED+548
                                                                          ENTRY_POINT
                                               REFS:    3338    3407
HEAP_TR                            1296        8 bit    LABEL    OSS$MAINFRAME_WIRED+54C
                                                                          ENTRY_POINT
                                               REFS:    3376    3378
HEAP_VER                           1297        8 bit    LABEL    OSS$MAINFRAME_WIRED+54D
                                                                          ENTRY_POINT
                                               REFS:    3377    3379

I180A                              5713       16 bit    LABEL    CODE+CB8
                                               REFS:    5737/P  5738/P
I180C                              5716        0 bit    LABEL    CODE+CC2
                                               REFS:    5714/P
I180E                              5728        0 bit    LABEL    CODE+CEA
                                               REFS:    5724/P
I180F                              5734        0 bit    LABEL    CODE+CFC
                                               REFS:    5720/P
I180G                              5736       32 bit    LABEL    CODE+D00
                                               REFS:    5735/P
IDLE                               4331       16 bit    LABEL    CODE+428  ENTRY_POINT
                                               REFS:    3370    3440      4319/P
IDLE10                             4367       32 bit    LABEL    CODE+494
                                               REFS:    4348/P  4351/P    4353/P
IDLE180                               1                 DEF
                                               REFS:    5609
IDLE180                            3054        0 bit    LABEL    CODE+C90  ENTRY_POINT
                                               REFS:    3055    5610/P
IDLE3                              4334        0 bit    LABEL    CODE+430
                                               REFS:    4359/P  4362/P
IDLE4                              4341       32 bit    LABEL    CODE+446
                                               REFS:    4336/P
IDLE5                              4345       32 bit    LABEL    CODE+452
                                               REFS:    4345/P  4349/P
IDLE9                              4364       16 bit    LABEL    CODE+48E
                                               UNREFERENCED
IDLECODE                            379       81        CONSTANT
                                               UNREFERENCED
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                            sec+off
IDLERES                            1        7       VARIABLE
                                                  UNREFERENCED
IDLERES                         3109        7       VARIABLE
                                                  REFS:   3114      3117      3120      3125      5618/P    5675/P
IDLETYPE                         412       24       CONSTANT
                                                  UNREFERENCED
IDLE_CNT                         413       25       CONSTANT
                                                  REFS:    224       268
IDLSTART                         411       16       CONSTANT
                                                  UNREFERENCED
IDLSTATS                         387      128       CONSTANT
                                                  REFS:    224       268
IDL_NOID                         409        0       CONSTANT
                                                  UNREFERENCED
IDL_W_IO                         410        8       CONSTANT
                                                  UNREFERENCED
IF_LEVEL                         604        1       CONSTANT
                                                  UNREFERENCED
IF_VERSN                         603        2       CONSTANT
                                                  UNREFERENCED
IJLEP                            386      122       CONSTANT
                                                  REFS:    220       264
IJLO                             385      120       CONSTANT
                                                  UNREFERENCED
IL                              1421        1       CONSTANT
                                                  REFS:   5006/P
ILFLAG                          3304        0       CONSTANT
                                                  REFS:   5093/P
ILSIZE                          3303        8       CONSTANT
                                                  REFS:   3310
IL_TBL                          3310      384 bit   LABEL     OSS$MAINFRAME_WIRED+2A50
                                                                ENTRY_POINT
                                                  REFS:   3357      3398      5008/P
INITMXP                         1356     3328 bit   LABEL     OSS$MAINFRAME_WIRED+2028
                                                                ENTRY_POINT
                                                  REFS:   3368      3380      3926/P
INT                             3136        0 bit   LABEL     CODE+D18  ENTRY_POINT
                                                  REFS:   3137      3371      5752      5753/P
INTDISLP                        3958        0 bit   LABEL     CODE+1E4
                                                  REFS:   4395/P
INTEGER                            1              DEF
                                                  REFS:   3077      3144      3159      3226      3241
INTMASK                            1        7       VARIABLE
                                                  UNREFERENCED
INTMASK                         3191        7       VARIABLE
                                                  REFS:   3196      3199      3202      3207      5761/P    5817/P
INTPORT                         1280        8 bit   LABEL     OSS$MAINFRAME_WIRED+4F6
                                                                ENTRY_POINT
                                                  REFS:   3365      3389      3850/P    5547/P
IOP$IO_PROCESSOR                 413              LABEL       EXTERNAL
                                                  REFS:    412     1566/P
IOP$PROCESS_IO_COMPLETIONS      1413              LABEL       EXTERNAL
                                                  REFS:   1412     2616/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                            sec+off
IOP$REQUEST_PROCESSOR           1613              LABEL       EXTERNAL
                                                  REFS:   1612     2826/P
IOP$TAPE_QUEUE_REQUEST          1213              LABEL       EXTERNAL
                                                  REFS:   1212     2406/P
IOP$TRANSLATE_BYTE_ADDRESS      1233              LABEL       EXTERNAL
                                                  REFS:   1232     2427/P
ISSUEKPT                         790        8       CONSTANT
                                                  REFS:   5349/P

JCBP                             371       24       CONSTANT
                                                  REFS:    204       248
JMP$MTR_JOB_SCHEDULER_REQUESTS  1673              LABEL       EXTERNAL
                                                  REFS:   1672     2889/P
JMP$UPDATE_SERV_CLASS_STATS_REQ 1953              LABEL       EXTERNAL
                                                  REFS:   1952     3183/P
JROOTSIZ                          48      256       CONSTANT
                                                  REFS:    106
JR_MXCB                          106      256       CONSTANT
                                                  REFS:   3804/P
JSP$MTR_JOB_SWAPPING_REQUESTS    693              LABEL       EXTERNAL
                                                  REFS:    692     1860/P
JSP$SWAP_POLLING                1553              LABEL       EXTERNAL
                                                  REFS:   1552     2763/P
JSTKFRAM                          60       32       CONSTANT
                                                  UNREFERENCED
JSTKSIZ1                          57     1024       CONSTANT
                                                  UNREFERENCED
JSTKSIZ2                          58     2048       CONSTANT
                                                  UNREFERENCED
JSTKSIZ3                          59      512       CONSTANT
                                                  UNREFERENCED
JSTLEN                            61       94       CONSTANT
                                                  UNREFERENCED
JTIME                            376       64       CONSTANT
                                                  REFS:    214       258      4218/P    4223/P    4263/P
J_MCRHLT                          97    57344       CONSTANT
                                                  UNREFERENCED
J_MCRUSR                          99     6924       CONSTANT
                                                  REFS:   4601/P
J_MTRMSK                          91    65532       CONSTANT
                                                  UNREFERENCED
J_USRABT                          95    52224       CONSTANT
                                                  REFS:   4722/P
J_USRMSK                          93    65399       CONSTANT
                                                  UNREFERENCED

KCB_RMA                         1232       64 bit   LABEL     OSS$MAINFRAME_WIRED+448
                                                                ENTRY_POINT
                                                  REFS:   3351      3425

LOCKCP                          3305        2       CONSTANT
                                                  REFS:   5072/P
LOCKWAIT                        1269      128 bit   LABEL     OSS$MAINFRAME_WIRED+4E0
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER-------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                          sec+off
                                                                        ENTRY_POINT
                                              REFS:   3362     3386     5061/P   5062/P   5066/P   5068/P
LOG_STAT                         394      252         CONSTANT
                                              UNREFERENCED
LPID                             363        5         CONSTANT
                                              REFS:    191      235
LPID8                            380       82         CONSTANT
                                              REFS:    193      237     4303/P   5139/P   5462/P   5487/P
LPIDZ                            184        0         VARIABLE
                                              REFS:    185      199/P    227      227      229      243/P    271      271
                                                       273

MANDDLST                        1233        8 bit     LABEL       OSS$MAINFRAME_WIRED+450
                                                                        ENTRY_POINT
                                              REFS:   3359     3392
MAPTIM                           382       96         CONSTANT
                                              REFS:    218      262     3795/P   5541/P   5952/P
MAXCST                           354        2         CONSTANT
                                              REFS:    185      229      273     1090/P   1349
MAXILO                          3302        6         CONSTANT
                                              REFS:   3310
MEMLIMIT                        1074       32 bit     LABEL       OSS$MAINFRAME_WIRED+10
                                                                        ENTRY_POINT
                                              REFS:   3353     3420     3861/P   3864/P   3865/P
MEMPORT                          362        4         CONSTANT
                                              REFS:    189      233     3842/P   3849/P   5546/P
MLIST                           1249       16 bit     LABEL       OSS$MAINFRAME_WIRED+498
                                                                        ENTRY_POINT
                                              REFS:   3352     3431
MMP$ADVISE_REQUEST_PROCESSOR     433                  LABEL             EXTERNAL
                                              REFS:    432      452      472     1587/P   1608/P   1629/P
MMP$FREE_FLUSH                   573                  LABEL             EXTERNAL
                                              REFS:    572      592     1732     1734/P   1755/P   2952/P
MMP$MTR_CHANGE_SEGMENT_TABLE     613                  LABEL             EXTERNAL
                                              REFS:    612     1776/P
MMP$MTR_FETCH_OFFSET_MOD_PAGES  1693                  LABEL             EXTERNAL
                                              REFS:   1692     2910/P
MMP$MTR_FETCH_PVA_UNWRITTEN_PGS 1133                  LABEL             EXTERNAL
                                              REFS:   1132     2322/P
MMP$MTR_LOCK_RING_1_STACK        893                  LABEL             EXTERNAL
                                              REFS:    892     2070/P
MMP$MTR_LOCK_UNLOCK_PAGES       1093                  LABEL             EXTERNAL
                                              REFS:   1092     1112     2280/P   2301/P
MMP$MTR_LOCK_UNLOCK_SEGMENT     1293                  LABEL             EXTERNAL
                                              REFS:   1292     2490/P
MMP$MTR_R1_SERVER_SEG_REQUEST   1873                  LABEL             EXTERNAL
                                              REFS:   1872     3099/P
MMP$MTR_READ_WRITE_IO            953                  LABEL             EXTERNAL
                                              REFS:    952     2133/P
MMP$MTR_RING1_SEGMENT_REQUEST    993                  LABEL             EXTERNAL
                                              REFS:    992     2175/P
MMP$MTR_SET_GET_SEGMENT_LENGTH   933                  LABEL             EXTERNAL
                                              REFS:    932     2112/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER-------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                          sec+off
MMP$MTR_WAIT_IO_COMPLETION      1333                  LABEL             EXTERNAL
                                              REFS:   1332     2532/P
MMP$PERIODIC_CALL               1493                  LABEL             EXTERNAL
                                              REFS:   1492     2700/P
MMP$PROCESS_ASSIGN_CONTIG_MEM   1833                  LABEL             EXTERNAL
                                              REFS:   1832     3057/P
MMP$PROCESS_ASSIGN_PAGES_REQ    1713                  LABEL             EXTERNAL
                                              REFS:   1712     2931/P
MMP$PROCESS_MOVE_PAGES_REQUEST  1813                  LABEL             EXTERNAL
                                              REFS:   1812     3036/P
MMTIME                          1289       64 bit     LABEL       OSS$MAINFRAME_WIRED+530
                                                                        ENTRY_POINT
                                              REFS:   3339     3421     4168/P
MM_EI                           3330       62         CONSTANT
                                              REFS:   5374/P   5422/P
MONREQ                          1380                  PROCEDURE
                                              REFS:   3987/P   4024/P   4062/P   4098/P   4134/P   4170/P   4495/P   4534/P
                                                      4605/P   4641/P   4685/P   4729/P   4820/P   4860/P   5244/P   5288/P
                                                      5374/P   5422/P   5561/P
MON_SMU                         3324       53         CONSTANT
                                              REFS:   4098/P   5730/P
MPS                              383      104         CONSTANT
                                              REFS:   3828/P
MST                             3461    17664 bit     LABEL       MTS$MONITOR_STACK+1A0
                                                                        ENTRY_POINT
                                              REFS:   1301/P   3457     3662/P   3747
MSTACKL                         3454     9324         CONSTANT
                                              REFS:   3455
MSTACKLX                        1267       64 bit     LABEL       OSS$MAINFRAME_WIRED+4D0
                                                                        ENTRY_POINT
                                              REFS:   3374     3382
MSTKFRAM                          56       32         CONSTANT
                                              REFS:   3486/P   4780/P   5700/P
MSTKSIZE                          55     6700         CONSTANT
                                              REFS:   3454     3462
MSTLEN                            62       20         CONSTANT
                                              REFS:   3454     3461     3668/P   3802/P
MTIME                            377       72         CONSTANT
                                              REFS:   4226/P   4254/P
MTM$MONITOR_INTERRUPT_HANDLER      1                  DEF
                                              UNREFERENCED
MTP$170_TRAP_HANDLER            5996                  LABEL             EXTERNAL
                                              REFS:   3278     3293     5998/P
MTP$ERROR_STOP                  3761                  LABEL             EXTERNAL
                                              REFS:   3766/P
MTP$MONITOR_SYSTEM_STATUS       1393                  LABEL             EXTERNAL
                                              REFS:   1392     2595/P
MTP$MTR_ERROR_STOP              3762                  LABEL             EXTERNAL
                                              REFS:   3767/P
MTP$MTR_STEP_UNSTEP_SYSTEM       713                  LABEL             EXTERNAL
                                              REFS:    712     1881/P
MTP$PROCESS_170_MTR_REQUESTS    1573                  LABEL             EXTERNAL
                                              REFS:   1572     2784/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
MTP$PROCESS_CPU_STATE_CHANGE    1893                LABEL                   EXTERNAL
                                             REFS:   1892       3120/P
MTP$PROCESS_DUE                 1513                LABEL                   EXTERNAL
                                             REFS:   1512       2721/P
MTP$PROCESS_SHORT_WARNING       1373                LABEL                   EXTERNAL
                                             REFS:   1372       2574/P
MTRPRIOR                        1283         16 bit LABEL       OSS$MAINFRAME_WIRED+4F8
                                                                            ENTRY_POINT
                                             REFS:   3346       3396       4960/P
MTRQMAX                         1449          0     VARIABLE
                                             REFS:    327        330        347        348        350        367        368        370
                                                      387        388        390        407        408        410        427        428
                                                      430        447        448        450        467        468        470        487
                                                      488        490        507        508        510        527        528        530
                                                      547        548        550        567        568        570        587        588
                                                      590        607        608        610        627        628        630        647
                                                      648        650        667        668        670        687        688        690
                                                      707        708        710        727        728        730        747        748
                                                      750        767        768        770        787        788        790        807
                                                      808        810        827        828        830        847        848        850
                                                      867        868        870        887        888        890        907        908
                                                      910        927        928        930        947        948        950        967
                                                      968        970        987        988        990       1007       1008       1010
                                                     1027       1028       1030       1047       1048       1050       1067       1068
                                                     1070       1087       1088       1090       1107       1108       1110       1127
                                                     1128       1130       1147       1148       1150       1167       1168       1170
                                                     1187       1188       1190       1207       1208       1210       1227       1228
                                                     1230       1247       1248       1250       1267       1268       1270       1287
                                                     1288       1290       1307       1308       1310       1327       1328       1330
                                                     1347       1348       1350       1367       1368       1370       1387       1388
                                                     1390       1407       1408       1410       1427       1428       1430       1447
                                                     1448       1450       1467       1468       1470       1487       1488       1490
                                                     1507       1508       1510       1527       1528       1530       1547       1548
                                                     1550       1567       1568       1570       1587       1588       1590       1607
                                                     1608       1610       1627       1628       1630       1647       1648       1650
                                                     1667       1668       1670       1687       1688       1690       1707       1708
                                                     1710       1727       1728       1730       1747       1748       1750       1767
                                                     1768       1770       1787       1788       1790       1807       1808       1810
                                                     1827       1828       1830       1847       1848       1850       1867       1868
                                                     1870       1887       1888       1890       1907       1908       1910       1927
                                                     1928       1930       1947       1948       1950       1967       1968       1970
                                                     1987       1988       1990       2007       2008       2010       2027       2028
                                                     2030       4681/P
MTRSTAK                         3462      53600 bit LABEL       MTS$MONITOR_STACK+A40
                                                                            ENTRY_POINT
                                             REFS:   2050       2060       2065       2070       2075       2085       3457       3475/P
                                                     3486/P     3497/P
MTRSTAKE                        3463          0 bit LABEL       MTS$MONITOR_STACK+246C

                                             REFS:   3745
MTRSTP                          1301         48 bit LABEL       OSS$MAINFRAME_WIRED+55B

                                             REFS:   3812/P     3911/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
MTRXPP                          1302         48 bit LABEL       OSS$MAINFRAME_WIRED+561

                                             REFS:   3925/P
MTS$MONITOR_STACK               3455       9324 byte SECTION       +0
                                             REFS:   3456       3471
MTV$IDLE_MESSAGE_LINE           1082          0 bit LABEL       OSS$MAINFRAME_WIRED+1B0
                                                                            ENTRY_POINT
                                             REFS:   3343
MTVDFTB                         1299         48 bit LABEL       OSS$MAINFRAME_WIRED+54F
                                                                            ENTRY_POINT
                                             REFS:   3375       3410       3880/P     3881/P     3883/P     4058/P     5721/P
MULTPRO                         1237         64 bit LABEL       OSS$MAINFRAME_WIRED+458
                                                                            ENTRY_POINT
                                             REFS:   3358       3394       5002/P     5518/P
MXP                             3460       3328 bit LABEL       MTS$MONITOR_STACK+0
                                                                            ENTRY_POINT
                                             REFS:   1302/P     2044       2054       2064       2074       2081       2094       2104
                                                     2111       2121       2131       2141       2151       2161       2171       2181
                                                     2191       2201       2211       2220       2225       2230       2235       2240
                                                     2245       2250       2255       2264       2270       2275       2280       2285
                                                     3458       3464/P     3475/P     3486/P     3497/P     3508/P     3519/P     3530/P
                                                     3541/P     3552/P     3563/P     3574/P     3585/P     3596/P     3607/P     3618/P
                                                     3629/P     3640/P     3651/P     3662/P     3668/P     3674/P     3680/P     3686/P
                                                     3692/P     3698/P     3704/P     3710/P     3721/P     3727/P     3733/P     3739/P
                                                     3746
M_MCRASY                          98       1168         CONSTANT
                                             REFS:   4332/P     4565/P     4570/P     4854/P     4938/P     4945/P
M_MCRDUE                         749      32768         CONSTANT
                                             UNREFERENCED
M_MCREI                          750        128         CONSTANT
                                             UNREFERENCED
M_MCRELT                         756          3         CONSTANT
                                             UNREFERENCED
M_MCREXC                         751       1024         CONSTANT
                                             REFS:   4472/P     4587/P     5193/P
M_MCREXS                         746        144         CONSTANT
                                             REFS:   4582/P
M_MCRHDW                         747      40960         CONSTANT
                                             REFS:   4490/P     4815/P
M_MCRHLT                          96      23340         CONSTANT
                                             REFS:   4815/P     4892/P
M_MCRMCL                         753         32         CONSTANT
                                             REFS:   5227/P
M_MCRPF                          752         64         CONSTANT
                                             REFS:   4902/P
M_MCRSEL                         754          2         CONSTANT
                                             UNREFERENCED
M_MCRSIT                         745         16         CONSTANT
                                             UNREFERENCED
M_MCRSRW                         748       8192         CONSTANT
                                             UNREFERENCED
M_MCRSW                          101       8192         CONSTANT
                                             REFS:   4332/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
M_MCRTRX                          755       1       CONSTANT
                                               UNREFERENCED
M_MTRMSK                           90   65532       CONSTANT
                                               REFS:  3674/P    4332/P    4371/P
M_UCRCFF                          759    1024       CONSTANT
                                               UNREFERENCED
M_UCRDB                           761     128       CONSTANT
                                               UNREFERENCED
M_UCRFF                           758    8192       CONSTANT
                                               UNREFERENCED
M_UCRKP                           760     512       CONSTANT
                                               UNREFERENCED
M_USRABT                           94   60927       CONSTANT
                                               REFS:  4972/P
M_USRMSK                           92   65407       CONSTANT
                                               REFS:  3680/P

NAP$MTR_REQUEST_PROCESSOR        1773               LABEL                 EXTERNAL
                                               REFS:  1772      2994/P
NEXTSTAT                          365       7       CONSTANT
                                               REFS:  4318/P    4335/P    5020/P
NIL                                 1               DEF
                                               REFS:  2050      2055      2060      2065      2070      2075      2080      2080
                                                      2085      2090      2100      2105      2110      2110      2115      2120
                                                      2120      2130      2130      2140      2140      2150      2150      2160
                                                      2160      2170      2170      2180      2180      2190      2190      2200
                                                      2200      2210      2210      2260      2275      3278      3288      3288
                                                      3293      3298      3298      3308      3308      3323      3323      3333
                                                      3333      3343      3343      3353      3353      3363      3363      3373
                                                      3373      3383      3383      3393      3393      3403      3403      3413
                                                      3413      3423      3423      3433      3433      3508/P    3541/P    3552/P
                                                      3563/P    3574/P    3585/P    3596/P    3607/P    3618/P    3629/P    3640/P
                                                      3651/P    6009/P    6020/P    6031/P    6048/P    6059/P    6070/P    6081/P
                                                      6092/P    6103/P    6114/P    6125/P    6136/P    6147/P    6158/P    6169/P
NOPF                             4914       0 bit   LABEL       CODE+79C
                                               REFS:  4904/P
NOSDATE                          1325     657       CONSTANT
                                               REFS:  3894/P
NOSEXIT                          1239      64 bit   LABEL       OSS$MAINFRAME_WIRED+468

                                               REFS:  3796/P    3834/P    5145/P    5498/P
NOSJPS                           1238      64 bit   LABEL       OSS$MAINFRAME_WIRED+460
                                                                          ENTRY_POINT
                                               REFS:  3348      3401      3904/P
NOSSEGP                          1300      48 bit   LABEL       OSS$MAINFRAME_WIRED+555
                                                                          ENTRY_POINT
                                               REFS:  3350      3405
NOSSEGT                          1243      48 bit   LABEL       OSS$MAINFRAME_WIRED+47C
                                                                          ENTRY_POINT
                                               REFS:  3371      3415      3813/P    3906/P
NOSSF                            1333      48 bit   LABEL       OSS$MAINFRAME_WIRED+62B

                                               REFS:  3862/P

               ***  REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter
```



```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
NOSTAB                           1240      48 bit   LABEL       OSS$MAINFRAME_WIRED+470
                                                                          ENTRY_POINT
                                               REFS:  3348      3414      3809/P    3829/P    3830/P    3937/P    4774/P    5699/P
NOSTIME                          1288     128 bit   LABEL       OSS$MAINFRAME_WIRED+520
                                                                          ENTRY_POINT
                                               REFS:  3339      3406      5496/P    5502/P    5504/P    5506/P
NOSTOD                           1324    1809       CONSTANT
                                               REFS:  3893/P
NOSVE_BT                         1320     104 bit   LABEL       OSS$MAINFRAME_WIRED+61E
                                                                          ENTRY_POINT
                                               REFS:  3352      3429
NOSXP                            1242      48 bit   LABEL       OSS$MAINFRAME_WIRED+476
                                                                          ENTRY_POINT
                                               REFS:  3347      3413      3901/P    4476/P    4588/P    4958/P    5137/P
NOS_DATE                         1318      64 bit   LABEL       OSS$MAINFRAME_WIRED+610

                                               REFS:  3899/P
NOS_TOD                          1317      64 bit   LABEL       OSS$MAINFRAME_WIRED+608
                                                                          ENTRY_POINT
                                               REFS:  3352      3430      3898/P
NP170PR                           578      14       CONSTANT
                                               REFS:  4305/P    5463/P
NP170TY                           577       0       CONSTANT
                                               UNREFERENCED
NP180PR                           579     150       CONSTANT
                                               REFS:  5140/P    5489/P
NPXTIME                           580     128       CONSTANT
                                               REFS:  5148/P    5150/P
NUM_CST                          1268      64 bit   LABEL       OSS$MAINFRAME_WIRED+4D8
                                                                          ENTRY_POINT
                                               REFS:  3373      3383
NUM_PROC                         1282       8 bit   LABEL       OSS$MAINFRAME_WIRED+4F7
                                                                          ENTRY_POINT
                                               REFS:  3367      3384

OFF                               423       1       CONSTANT
                                               UNREFERENCED
ON                                422       0       CONSTANT
                                               UNREFERENCED
OSCDATA                           681       0       CONSTANT
                                               UNREFERENCED
OSCDBUG                           685       4       CONSTANT
                                               REFS:  5355/P
OSCENT                            683       2       CONSTANT
                                               UNREFERENCED
OSCEXIT                           684       3       CONSTANT
                                               UNREFERENCED
OSCMTR                            686       5       CONSTANT
                                               REFS:  4427/P    4465/P    5185/P    5189/P
OSCUNUS                           682       1       CONSTANT
                                               UNREFERENCED
OSKEXC7                           699    4002       CONSTANT
                                               REFS:   703      5185/P

               ***  REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter
```

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
OSKEXC7X	703	8098		CONSTANT		
				REFS: 5189/P		
OSKEXC8	698	4001		CONSTANT		
				REFS: 702	4465/P	
OSKEXC8X	702	8097		CONSTANT		
				REFS: 4427/P		
OSKPURG	697	4000		CONSTANT		
				UNREFERENCED		
OSKTRAP	791	4005		CONSTANT		
				REFS: 5355/P		
OSKTRPJ	701	4004		CONSTANT		
				REFS: 705		
OSKTRPJX	705	8100		CONSTANT		
				UNREFERENCED		
OSKTRPM	700	4003		CONSTANT		
				REFS: 704		
OSKTRPMX	704	8099		CONSTANT		
				UNREFERENCED		
OSKXBIAS	690	4096		CONSTANT		
				REFS: 702	703	704 705
OSP$PROCESS_JOB_KEYPOINT_REQ	1473			LABEL		EXTERNAL
				REFS: 1472	2679/P	
OSP$PROCESS_MTR_PAGE_FAULT	3763			LABEL		EXTERNAL
				REFS: 3768/P		
OSS$MAINFRAME_WIRED	1058	0	byte	SECTION	+0	
				REFS: 1059	1059	
OSS$MAINFRAME_WIRED_CB	5963	0	byte	SECTION	+0	
				REFS: 5964	6004	
OST$EI	592	1		CONSTANT		
				UNREFERENCED		
OST$NBE	594	2		CONSTANT		
				UNREFERENCED		
OST$NOS	593	1		CONSTANT		
				UNREFERENCED		
OST$NVE	595	2		CONSTANT		
				UNREFERENCED		
OST$PSR	596	739		CONSTANT		
				UNREFERENCED		
OSV$BOOT	1339	8	bit	LABEL	OSS$MAINFRAME_WIRED+63F	
					ENTRY_POINT	
				REFS: 1336	3807/P	
OSV$BOOT_IS_EXECUTING	1342	8	bit	LABEL	OSS$MAINFRAME_WIRED+650	
					ENTRY_POINT	
				REFS: 1337		
OSV$BOOT_SDTE	1340	64	bit	LABEL	OSS$MAINFRAME_WIRED+640	
					ENTRY_POINT	
				REFS: 1334	3801/P	
OSV$MAINFRAME_WIRED_CB_HEAP	5965	112	bit	LABEL	OSS$MAINFRAME_WIRED_CB+0	
					ENTRY_POINT	
				REFS: 5966		
OSV_BL	1287	256	bit	LABEL	OSS$MAINFRAME_WIRED+500	
					ENTRY_POINT	
				REFS: 3369	3418	

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
OS_JPS	571	180		CONSTANT		
				UNREFERENCED		
OS_SFSA	570	176		CONSTANT		
				UNREFERENCED		
OS_TERMS	1230	8	bit	LABEL	OSS$MAINFRAME_WIRED+441	
					ENTRY_POINT	
				REFS: 3355	3404	5459/P
OS_TYPE	1229	8	bit	LABEL	OSS$MAINFRAME_WIRED+440	
					ENTRY_POINT	
				REFS: 3354	3403	3888/P 3919/P
PARAM	209			PROCEDURE		
				REFS: 5618/P	5761/P	5869/P
PDPV$SCD_TIME	1305	48	bit	LABEL	OSS$MAINFRAME_WIRED+573	
				REFS: 4021/P		
PER_CALL	3327	58		CONSTANT		
				REFS: 4170/P		
PEXTIOU	1304	48	bit	LABEL	OSS$MAINFRAME_WIRED+56D	
				REFS: 3982/P	5549/P	
PLOADA	266			PROCEDURE		
				UNREFERENCED		
PLOADX	289			PROCEDURE		
				REFS: 5675/P	5817/P	
POINTER	1			DEF		
				REFS: 3081	3148	3163 3230 3245
PREVSTAT	393	251		CONSTANT		
				UNREFERENCED		
PRIOR180	360	1		CONSTANT		
				REFS: 202	246	
PROCEDUR	143			PROCEDURE		
				REFS: 5610/P	5753/P	5861/P
PROC_CPU	3332	78		CONSTANT		
				REFS: 4339/P		
PROC_DFT	3331	66		CONSTANT		
				REFS: 4062/P	5726/P	
PROC_DUE	3328	59		CONSTANT		
				REFS: 4534/P	4860/P	5244/P
PROC_IO	3325	54		CONSTANT		
				REFS: 3987/P		
PRSIT	5102	0	bit	LABEL	CODE+94C	
				REFS: 4350/P	4578/P	4951/P 5324/P
PR_PF	513			LABEL		EXTERNAL
				REFS: 512	1671/P	
PSTORAP	333			PROCEDURE		
				UNREFERENCED		
PSTORXP	323			PROCEDURE		
				UNREFERENCED		
PSTRING	312			PROCEDURE		
				UNREFERENCED		
PSWARN	3323	52		CONSTANT		
				REFS: 4495/P	4820/P	5288/P

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
PUR_CA	404	1		CONSTANT		
				REFS: 5528/P	5529/P	5536/P
PUR_MAP	405	2		CONSTANT		
				REFS: 5535/P	5536/P	5542/P
RC	1422	2		CONSTANT		
				UNREFERENCED		
REF	1			DEF		
				REFS: 3113	3180	3195 3262 3277
REQTBL	1481	0	bit	LABEL	OSS$MAINFRAME_WIRED+2240	
					ENTRY_POINT	
				REFS: 320	330	340 350 360 370 380 390
				400	410	420 430 440 450 460 470
				480	490	500 510 520 530 540 550
				560	570	580 590 600 610 620 630
				640	650	660 670 680 690 700 710
				720	730	740 750 760 770 780 790
				800	810	820 830 840 850 860 870
				880	890	900 910 920 930 940 950
				960	970	980 990 1000 1010 1020 1030
				1040	1050	1060 1070 1080 1090 1100 1110
				1120	1130	1140 1150 1160 1170 1180 1190
				1200	1210	1220 1230 1240 1250 1260 1270
				1280	1290	1300 1310 1320 1330 1340 1350
				1360	1370	1380 1390 1400 1410 1420 1430
				1440	1450	1460 1470 1480 1490 1500 1510
				1520	1530	1540 1550 1560 1570 1580 1590
				1600	1610	1620 1630 1640 1650 1660 1670
				1680	1690	1700 1710 1720 1730 1740 1750
				1760	1770	1780 1790 1800 1810 1820 1830
				1840	1850	1860 1870 1880 1890 1900 1910
				1920	1930	1940 1950 1960 1970 1980 1990
				2000	2010	2020 2030 2289/P 2328/P 2358/P 2389/P
				2419/P	2449/P	2560/P 2590/P 2620/P 2650/P 2687/P 2693/P
				2710/P	2767/P	2797/P 2911/P 2941/P 2971/P 3001/P 3031/P
				3356	3397	4227/P
RETRY_DUE	611	16		CONSTANT		
				UNREFERENCED		
RETRY_FAILED	610	1		CONSTANT		
				UNREFERENCED		
RFP$QUEUE_DATA_FRAGMENTS	1753			LABEL	EXTERNAL	
				REFS: 1752	2973/P	
RN	1420	0		CONSTANT		
				REFS: 2691/P		
RNNOS19	5368	16	bit	LABEL	CODE+AF2	
				REFS: 5361/P		
ROOT	1061	64	bit	LABEL	OSS$MAINFRAME_WIRED+0	
					ENTRY_POINT	
				REFS: 1060	2100	2115 3426 3530/P 3759/P
RQCNTMAX	1424	16		CONSTANT		
				REFS: 4235/P	4248/P	5080/P 5089/P
RQFAULT	3321	20		CONSTANT		
				REFS: 4605/P	4729/P	

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
RQPF	3320	9		CONSTANT		
				REFS: 4641/P		
RQPR12	5071	16	bit	LABEL	CODE+906	
				REFS: 5013/P		
RQPR14	5073	0	bit	LABEL	CODE+90C	
				REFS: 5005/P	5007/P	
RQPR20	5089	32	bit	LABEL	CODE+93A	
				REFS: 5086/P		
RQPR30	5094	16	bit	LABEL	CODE+94A	
				REFS: 5090/P		
RQPR4	5015	16	bit	LABEL	CODE+86A	
				REFS: 5023/P	5040/P	
RQPR55	2833	32	bit	LABEL	CODE+8B8	
				REFS: 5021/P	5041/P	
RQPR6	5059	16	bit	LABEL	CODE+8E8	
				REFS: 5017/P		
RQPR0C	5001	0	bit	LABEL	CODE+844	ENTRY_POINT
				REFS: 2315/P	2345/P	2375/P 2405/P 2435/P 2465/P 2576/P 2606/P
				2636/P	2666/P	2692/P 2696/P 2726/P 2783/P 2813/P 2927/P
				2957/P	2987/P	3017/P 3047/P 3370 3443
RQTABLE	1427			PROCEDURE		
				REFS: 1482/P	1503/P	1524/P 1545/P 1566/P 1587/P 1608/P 1629/P
				1650/P	1671/P	1692/P 1713/P 1734/P 1755/P 1776/P 1797/P
				1818/P	1839/P	1860/P 1881/P 1902/P 1923/P 1944/P 1965/P
				1986/P	2007/P	2028/P 2049/P 2070/P 2091/P 2112/P 2133/P
				2154/P	2175/P	2196/P 2217/P 2238/P 2259/P 2280/P 2301/P
				2322/P	2343/P	2364/P 2385/P 2406/P 2427/P 2448/P 2469/P
				2490/P	2511/P	2532/P 2553/P 2574/P 2595/P 2616/P 2637/P
				2658/P	2679/P	2700/P 2721/P 2742/P 2763/P 2784/P 2805/P
				2826/P	2847/P	2868/P 2889/P 2910/P 2931/P 2952/P 2973/P
				2994/P	3015/P	3036/P 3057/P 3078/P 3099/P 3120/P 3141/P
				3162/P	3183/P	3204/P 3225/P 3246/P 3267/P
RQTBLES	1419	24		CONSTANT		
				REFS: 320	330	330 340 350 350 360 370
				370	380	390 390 400 410 410 420
				430	430	440 450 450 460 470 470
				480	490	490 500 510 510 520 530
				530	540	550 550 560 570 570 580
				590	590	600 610 610 620 630 630
				640	650	650 660 670 670 680 690
				690	700	710 710 720 730 730 740
				750	750	760 770 770 780 790 790
				800	810	810 820 830 830 840 850
				850	860	870 870 880 890 890 900
				910	910	920 930 930 940 950 950
				960	970	970 980 990 990 1000 1010
				1010	1020	1030 1030 1040 1050 1050 1060
				1070	1070	1080 1090 1090 1100 1110 1110
				1120	1130	1130 1140 1150 1150 1160 1170
				1170	1180	1190 1190 1200 1210 1210 1220
				1230	1230	1240 1250 1250 1260 1270 1270
				1280	1290	1290 1300 1310 1310 1320 1330
				1330	1340	1350 1350 1360 1370 1370 1380

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
                                                   1390      1390      1400      1410      1410      1420      1430      1430
                                                   1440      1450      1450      1460      1470      1470      1480      1490
                                                   1490      1500      1510      1510      1520      1530      1530      1540
                                                   1550      1550      1560      1570      1570      1580      1590      1590
                                                   1600      1610      1610      1620      1630      1630      1640      1650
                                                   1650      1660      1670      1670      1680      1690      1690      1700
                                                   1710      1710      1720      1730      1730      1740      1750      1750
                                                   1760      1770      1770      1780      1790      1790      1800      1810
                                                   1810      1820      1830      1830      1840      1850      1850      1860
                                                   1870      1870      1880      1890      1890      1900      1910      1910
                                                   1920      1930      1930      1940      1950      1950      1960      1970
                                                   1970      1980      1990      1990      2000      2010      2010      2020
                                                   2030      2030      2299/P    2329/P    2359/P    2389/P    2419/P    2449/P
                                                   2560/P    2590/P    2620/P    2650/P    2710/P    2767/P    2797/P    2911/P
                                                   2941/P    2971/P    3001/P    3031/P    4227/P
RQUNIM                           3319          0      CONSTANT
                                             REFS:     2694/P
RUNEXIT                          5510         16 bit   LABEL     CODE+BEA
                                             REFS:     5133/P
RUNNING                           431          0      CONSTANT
                                             UNREFERENCED
RUNNOS10                         5286         32 bit   LABEL     CODE+A6C
                                             REFS:     5238/P    5279/P
RUNNOS11                         5322         32 bit   LABEL     CODE+A84
                                             REFS:     5287/P
RUNNOS12                         5328         32 bit   LABEL     CODE+A90
                                             REFS:     5323/P
RUNNOS16                         5334         16 bit   LABEL     CODE+A9C
                                             REFS:     5329/P
RUNNOS18                         5348         32 bit   LABEL     CODE+ABC
                                             REFS:     5339/P
RUNNOS19                         5360         16 bit   LABEL     CODE+ADA
                                             REFS:     5350/P
RUNNOS20                         5409         32 bit   LABEL     CODE+B18
                                             REFS:     5337/P
RUNNOS21                         5413         32 bit   LABEL     CODE+B28
                                             REFS:     5410/P
RUNNOS22                         5417         16 bit   LABEL     CODE+B36
                                             REFS:     5415/P
RUNNOS24                         5454         32 bit   LABEL     CODE+B58
                                             REFS:     2964      2967      2979      5374/P
RUNNOS30                         5469         32 bit   LABEL     CODE+B80
                                             REFS:     5416/P    5456/P
RUNNOS32                         5484         32 bit   LABEL     CODE+BA4
                                             REFS:     5481/P
RUNNOS35                         5487         32 bit   LABEL     CODE+BAC
                                             REFS:     5478/P
RUNNOS50                         5495         16 bit   LABEL     CODE+BBC
                                             REFS:     5230/P    5341/P    5365/P    5464/P    5485/P
RUNNOS55                         5507         32 bit   LABEL     CODE+BE2
                                             REFS:     5503/P
RUNNOS6                          5154          0 bit   LABEL     CODE+99A
                                             REFS:     5342/P    5356/P    5366/P    5470/P    5490/P

                          ***  REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter
```



```
IDENTIFIER--------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                              ON LINE                           sec+off
RUNNOS8                          5234         16 bit   LABEL     CODE+A28
                                             REFS:     5228/P
RUN_NOS                          5130          0 bit   LABEL     CODE+968  ENTRY_POINT
                                             REFS:     3370      3444      4321/P    4480/P    4592/P    4968/P    5715/P
R_BC                              779         71      CONSTANT
                                             REFS:     3784/P    4775/P    5703/P    5928/P
R_CFF_C                           773        224      CONSTANT
                                             UNREFERENCED
R_DI                              782        228      CONSTANT
                                             UNREFERENCED
R_DLP                             787        197      CONSTANT
                                             UNREFERENCED
R_DMR                             783        229      CONSTANT
                                             UNREFERENCED
R_EID                             765         16      CONSTANT
                                             REFS:     3790/P    3833/P
R_JPS                             774         97      CONSTANT
                                             REFS:     3825/P    3929/P    4266/P    4964/P    5135/P    5508/P
R_KEF0                            780        202      CONSTANT
                                             REFS:     4770/P    4982/P    5234/P    5334/P
R_KEF1                            781        203      CONSTANT
                                             UNREFERENCED
R_MCR                             778         66      CONSTANT
                                             UNREFERENCED
R_MM                              785         96      CONSTANT
                                             REFS:     4331/P    4370/P
R_PID                             766         17      CONSTANT
                                             UNREFERENCED
R_PIT                             776        201      CONSTANT
                                             REFS:     3959/P    4229/P    4236/P    4377/P    4474/P    4481/P    4573/P    4595/P
                                                       4768/P    4984/P    5003/P    5059/P    5077/P    5701/P    5740/P    5953/P
R_PSM                             769         74      CONSTANT
                                             UNREFERENCED
R_PTA                             767         72      CONSTANT
                                             UNREFERENCED
R_PTL                             768         73      CONSTANT
                                             UNREFERENCED
R_SIT                             775         98      CONSTANT
                                             REFS:     3944/P    4250/P    4368/P    5111/P
R_STL                             777         69      CONSTANT
                                             UNREFERENCED
R_TD                              771        192      CONSTANT
                                             REFS:     5190/P
R_TE                              770        194      CONSTANT
                                             REFS:     3948/P    4322/P    4483/P    4593/P    5183/P    5351/P    5710/P
R_TED                             772        195      CONSTANT
                                             REFS:     4980/P
R_TP                              786        196      CONSTANT
                                             UNREFERENCED
R_UM                              784        230      CONSTANT
                                             UNREFERENCED

SCB                              1079       3200 bit   LABEL     OSS$MAINFRAME_WIRED+20

                          ***  REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter
```

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
						ENTRY_POINT
			REFS:	1080	3354	3411 3832/P 3977/P 5106/P 5736/P
SCBIDLER	442	16		CONSTANT		
				UNREFERENCED		
SCBNSRV	444	24		CONSTANT		
			REFS:	3832/P	3977/P	5106/P
SCBSIZE	439	400		CONSTANT		
			REFS:	1079		
SCBSTEPR	443	18		CONSTANT		
			REFS:	5736/P		
SCBTIME	1291	64	bit	LABEL	OSS$MAINFRAME_WIRED+540	
						ENTRY_POINT
			REFS:	3340	3422	4096/P
SCBVEC	1080	0	bit	LABEL	OSS$MAINFRAME_WIRED+28	
						ENTRY_POINT
			REFS:	3354	3412	
SCBVECSIM	441	11		CONSTANT		
			REFS:	1080		
SCB_CPUS	440	9		CONSTANT		
				UNREFERENCED		
SDTXSIZE	42	48		CONSTANT		
				UNREFERENCED		
SFP$MTR_STATS_FACILITY_REQUESTS	1913			LABEL		EXTERNAL
			REFS:	1912	3141/P	
SFSA_MCR	715	48		CONSTANT		
			REFS:	4783/P		
SFSA_UCR	716	40		CONSTANT		
			REFS:	4973/P		
SITVALUE	1266	64	bit	LABEL	OSS$MAINFRAME_WIRED+4C8	
						ENTRY_POINT
			REFS:	3341	3402	3943/P 4367/P 5110/P
SJMTRXCB	1271	48	bit	LABEL	OSS$MAINFRAME_WIRED+4F0	
						ENTRY_POINT
			REFS:	3363	3387	
SN170MCB	72	2		CONSTANT		
			REFS:	3879/P		
SNJFJOB	77	3		CONSTANT		
				UNREFERENCED		
SNNOS170	81	3		CONSTANT		
			REFS:	6229		
SNNOSMTR	73	3		CONSTANT		
				UNREFERENCED		
SNNTH170	83	5		CONSTANT		
			REFS:	3815/P	6233	
SNNTHMTR	75	5		CONSTANT		
			REFS:	3814/P		
SNPTMTR	71	0		CONSTANT		
				UNREFERENCED		
SNSF170	82	4		CONSTANT		
			REFS:	3817/P	6042/P	6231
SNSFMTR	74	4		CONSTANT		
			REFS:	3816/P		
SPIN_1	5941	16	bit	LABEL	CODE+D30	

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
			REFS:	5944/P		
SPIN_CPU	1			DEF		
			REFS:	5860		
SPIN_CPU	3218	0	bit	LABEL	CODE+D20	ENTRY_POINT
			REFS:	3219	5861/P	
STATSIZE	43	280		CONSTANT		
				UNREFERENCED		
STEPMES	1362	248	bit	LABEL	OSS$MAINFRAME_WIRED+21E7	
			REFS:	5024/P		
STEPPED	432	1		CONSTANT		
				UNREFERENCED		
STEP_PR	406	3		CONSTANT		
			REFS:	5542/P		
STRING	1			DEF		
			REFS:	3096	3163	3178 3245 3260
SUBRANGE	1			DEF		
			REFS:	3062	3066	3071 3071 3076 3077 3081 3081
				3091	3093	3096 3108 3144 3148 3153 3153
				3158	3159	3163 3163 3173 3175 3178 3190
				3226	3230	3235 3235 3240 3241 3245 3245
				3255	3257	3260 3272 5618/P 5761/P 5869/P
SWAPTIME	1290	64	bit	LABEL	OSS$MAINFRAME_WIRED+538	
						ENTRY_POINT
			REFS:	3340	3432	4132/P
SWAP_JOB	3329	61		CONSTANT		
			REFS:	4134/P		
SYP$MTR_INJECT_HARDWARE_FAULT	2033			LABEL		EXTERNAL
			REFS:	2032	3267/P	
SYSTEMHR	1293	8	bit	LABEL	OSS$MAINFRAME_WIRED+549	
						ENTRY_POINT
			REFS:	3338	3408	
TASKID	367	16		CONSTANT		
			REFS:	200	244	4396/P
TERMMESS	389	168		CONSTANT		
				UNREFERENCED		
TMP$CREATE_JOB	533			LABEL		EXTERNAL
			REFS:	532	1692/P	
TMP$CREATE_TASK	493			LABEL		EXTERNAL
			REFS:	492	1650/P	
TMP$CYCLE	353			LABEL		EXTERNAL
			REFS:	352	1503/P	
TMP$DELAY	373			LABEL		EXTERNAL
			REFS:	372	1524/P	
TMP$EXIT_JOB	553			LABEL		EXTERNAL
			REFS:	552	1713/P	
TMP$FETCH_TASK_STATISTICS	773			LABEL		EXTERNAL
			REFS:	772	1944/P	
TMP$JOB_RECOVERY_REQUESTS	973			LABEL		EXTERNAL
			REFS:	972	2154/P	
TMP$MTR_PROCESS_SYSTEM_ERROR	753			LABEL		EXTERNAL
			REFS:	752	1923/P	

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

```
IDENTIFIER-------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                           sec+off
TMP$MTR_READY_SYSTEM_TASK       1273                 LABEL               EXTERNAL
                                              REFS:   1272     2469/P
TMP$MTR_READY_TASK               833                 LABEL               EXTERNAL
                                              REFS:    832     2007/P
TMP$MTR_SEND_SIGNAL              913                 LABEL               EXTERNAL
                                              REFS:    912     2091/P
TMP$MTR_SET_SYSTEM_FLAG          853                 LABEL               EXTERNAL
                                              REFS:    852     2028/P
TMP$MTR_UPDATE_JOB_TASK_ENVIRO  1053                 LABEL               EXTERNAL
                                              REFS:   1052     2238/P
TMP$MTR_WAIT                     873                 LABEL               EXTERNAL
                                              REFS:    872     2049/P
TMP$PROCESS_TASK_MCR_FAULT       733                 LABEL               EXTERNAL
                                              REFS:    732     1902/P
TMP$PROCESS_UNKNOWN_REQ_FAULT    333                 LABEL               EXTERNAL
                                              REFS:    332     392      632      652      672      792      812      1032
                                                      1072     1482/P   1532     1545/P   1592     1797/P   1818/P   1839/P
                                                      1965/P   1972     1986/P   1992     2012     2217/P   2259/P   2742/P
                                                      2805/P   3204/P   3225/P   3246/P
TMP$SWITCH_TASK                 1353                 LABEL               EXTERNAL
                                              REFS:   1352     2553/P
TMP$TASK_EXIT                   1013                 LABEL               EXTERNAL
                                              REFS:   1012     2196/P
TMV$PTL_LOCK                    3764                 LABEL               EXTERNAL
                                              REFS:   3769/P
TM_SIZE                          416         0       CONSTANT
                                              UNREFERENCED
TM_TEXT                          418         2       CONSTANT
                                              UNREFERENCED
TM_UN_ID                         417         1       CONSTANT
                                              UNREFERENCED
TOTALT                          1423         8       CONSTANT
                                              REFS:   4234/P   4240/P   5079/P   5083/P
TRACE                           1349     33024 bit   LABEL     OSS$MAINFRAME_WIRED+658
                                                                         ENTRY_POINT
                                              REFS:    199/P    243/P   3349     3417
TRACECTL                         388       160       CONSTANT
                                              REFS:    197      241     2472/P   2499/P   2526/P   2733/P   2850/P   2877/P
TRACESIZ                        1345       256       CONSTANT
                                              REFS:    199/P    243/P   1349     1352
TRAPRTN                         4767         0 bit   LABEL     CODE+6B8  ENTRY_POINT
                                              REFS:   3370     3442     3758/P
TRASY15                         4972        32 bit   LABEL     CODE+828
                                              REFS:   4947/P   4956/P   4967/P
TRASY5                          4952        32 bit   LABEL     CODE+7EC
                                              REFS:   4949/P   4950/P
TRASY8                          4955        32 bit   LABEL     CODE+7F8
                                              REFS:   4953/P
TRASY9                          4964        16 bit   LABEL     CODE+818
                                              REFS:   4962/P
TRCKDUE                         4852        32 bit   LABEL     CODE+732
                                              REFS:   4819/P
TREXIT                          4980        16 bit   LABEL     CODE+836
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter


```
IDENTIFIER-------------------DEFINED-----SIZE unit--TYPE-------LOCATION--ATTRIBUTES
                             ON LINE                           sec+off
                                              UNREFERENCED
TRHDW5                          4892        32 bit   LABEL     CODE+75C
                                              REFS:   4853/P
TRHDWX                          4898         0 bit   LABEL     CODE+770
                                              REFS:   4817/P   4894/P
TRNOM                           4941         0 bit   LABEL     CODE+7D2
                                              REFS:   4935/P
TRRESEX                         4927         0 bit   LABEL     CODE+7B8
                                              REFS:   4922/P   4925/P
TRSTOP                          4895        32 bit   LABEL     CODE+766
                                              REFS:   4913/P   4975/P
TSCKPR                          4300        32 bit   LABEL     CODE+3C6
                                              REFS:   4214/P
TSCKPR3                         4316        32 bit   LABEL     CODE+3FE
                                              REFS:   4301/P   4309/P
TSCKPR5                         4322        16 bit   LABEL     CODE+416
                                              REFS:   4320/P
TSKSW                           3322        51       CONSTANT
                                              REFS:   4227/P   4228/P
TSK_SW                           403         0       CONSTANT
                                              REFS:   5521/P   5523/P   5529/P
TSWIT                           4210        32 bit   LABEL     CODE+2E2
                                              REFS:   3970/P   3973/P
TSWIT4                          4227        32 bit   LABEL     CODE+31C
                                              REFS:   4215/P
TSWIT5                          4246        32 bit   LABEL     CODE+356
                                              REFS:   4243/P
TSWIT8                          2470         0 bit   LABEL     CODE+398
                                              REFS:   4256/P   4268/P

VAL                                1                 DEF
                                              REFS:   3098     3113     3180     3195     3262     3277     5618/P   5761/P
                                                      5869/P
VE_JPS                           573       188       CONSTANT
                                              UNREFERENCED
VE_SFSA                          572       184       CONSTANT
                                              UNREFERENCED
VE_VRSN                         1251        64 bit   LABEL     OSS$MAINFRAME_WIRED+4A0

                                              REFS:   3851/P

XCBP                             372        32       CONSTANT
                                              REFS:    210      254     3822/P   4211/P   4247/P   4933/P
XCBRMA                           373        40       CONSTANT
                                              REFS:   3824/P   3931/P   4265/P   5507/P
XCBSIZE                           41      1024       CONSTANT
                                              UNREFERENCED
XFRC_P                          3400                 LABEL               EXTERNAL
                                              REFS:   1246/P   3399
XP17OMF                          728        43       CONSTANT
                                              UNREFERENCED
XPA                              122                 PROCEDURE
                                              REFS:   3464/P   3475/P   3710/P   5998/P
```

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
XPAREG	112			PROCEDURE		
				REFS:	3486/P 3497/P 3508/P 3519/P 3530/P 3541/P 3552/P 3563/P	
					3574/P 3585/P 3596/P 3607/P 3618/P 3629/P 3640/P 3651/P	
					6009/P 6020/P 6031/P 6048/P 6059/P 6070/P 6081/P 6092/P	
					6103/P 6114/P 6125/P 6136/P 6147/P 6158/P 6169/P	
XPBC1	737	104		CONSTANT		
				UNREFERENCED		
XPBC2	738	112		CONSTANT		
				REFS:	2275 3727/P	
XPCFF	731	16		CONSTANT		
				UNREFERENCED		
XPDEBUGI	722	288		CONSTANT		
				UNREFERENCED		
XPDEBUGM	723	289		CONSTANT		
				UNREFERENCED		
XPDLP	719	290		CONSTANT		
				UNREFERENCED		
XPFDESC	730	16		CONSTANT		
				UNREFERENCED		
XPFLGTE	725	16		CONSTANT		
				REFS:	2270 3478 3721/P 4725/P 5277/P 6222/P	
XPINITV	1354	3328	bit	LABEL	OSS$MAINFRAME_WIRED+1E88 ENTRY_POINT	
				REFS:	3354 3424 3922/P	
XPKM	735	64		CONSTANT		
				REFS:	2240 3458 3686/P 6198/P	
XPLRN	739	296		CONSTANT		
				REFS:	2255 3473 3704/P 6216/P	
XPMCR	726	48		CONSTANT		
				REFS:	4397/P 4435/P 4477/P 4584/P 4589/P 4936/P 4940/P 4959/P	
					5187/P 5192/P 5195/P	
XPMM	734	32		CONSTANT		
				REFS:	2230 3448 3674/P 6186/P	
XPPIT	736	88		CONSTANT		
				REFS:	2245 2250 3463 3468 3692/P 3698/P 4216/P 4217/P	
					4258/P 4259/P 6204/P 6210/P	
XPSIZE	717	416		CONSTANT		
				REFS:	1354 1356 3454 3460 3924/P 3924/P 3928/P 3928/P	
					5988	
XPSTAL	721	280		CONSTANT		
				REFS:	2220 3662/P 3908/P	
XPSTAU	720	272		CONSTANT		
				REFS:	3910/P	
XPSTL	732	128		CONSTANT		
				REFS:	2225 3443 3668/P 6180/P	
XPTOS	741	298		CONSTANT		
				REFS:	2054 3475/P	
XPTP	718	282		CONSTANT		
				REFS:	2264 3710/P	
XPUCR	727	40		CONSTANT		
				REFS:	4721/P 5480/P 5483/P	
XPUM	733	24		CONSTANT		
				REFS:	2235 3453 3680/P 6192/P	

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter

IDENTIFIER	DEFINED ON LINE	SIZE	unit	TYPE	LOCATION sec+off	ATTRIBUTES
XPUTP	724	274		CONSTANT		
				UNREFERENCED		
XPV	132			PROCEDURE		
				REFS:	3662/P 3668/P 3674/P 3680/P 3686/P 3692/P 3698/P 3704/P	
					3721/P 3727/P 3733/P 3739/P 6042/P 6180/P 6186/P 6192/P	
					6198/P 6204/P 6210/P 6216/P 6222/P	
XPVMID	729	8		CONSTANT		
				REFS:	5228/P 5338/P 5409/P 5479/P	
XPXREGS	740	136		CONSTANT		
				REFS:	4677/P 4679/P 5348/P 5353/P	
XREG	620			PROCEDURE		
				REFS:	908/P 914/P 920/P 926/P 949/P 955/P	
XTRACE	821			PROCEDURE		
				REFS:	4268/P 4398/P 4436/P 4784/P 5155/P 5196/P	
XXXPLOC	3052	0		VARIABLE		
				REFS:	3109 3110 3110 3134 3191 3192 3192 3216	
					3273 3274 3274	
X_CLOCK	128	4		CONSTANT		
				REFS:	129 914/P 3960/P 4225/P 4249/P 4378/P 4475/P 4482/P	
					4574/P 4596/P 4769/P 4985/P 5702/P 5741/P 5954/P	
X_ENVIR1	994	199		CONSTANT		
				REFS:	4232/P 4337/P 4909/P 5074/P	
X_INFRC	158	7		CONSTANT		
				REFS:	159 955/P 5143/P 5144/P 5144/P 5146/P 5499/P 5949/P	
					5950/P 5950/P	
X_INMCR	153	6		CONSTANT		
				REFS:	154 949/P 2883 2887 2898 5187/P 5188/P 5192/P	
					5194/P 5196/P 5228/P 5237/P 5276/P 5281/P 5286/P 5322/P	
					5328/P 5336/P 5340/P 5364/P 5414/P 5457/P 5477/P	
X_KEF	133	5		CONSTANT		
				REFS:	134 920/P 4771/P 4983/P 5235/P 5335/P	
X_MCR	123	3		CONSTANT		
				REFS:	124 908/P 2532 2536 2547 2739 2743 2754	
					4435/P 4436/P 4464/P 4473/P 4481/P 4493/P 4527/P 4565/P	
					4571/P 4575/P 4579/P 4583/P 4584/P 4585/P 4597/P 4602/P	
					4639/P 4675/P 4783/P 4784/P 4816/P 4818/P 4852/P 4854/P	
					4893/P 4903/P 4937/P 4946/P 4948/P 4952/P 4955/P	
X_RESUME	138	8		CONSTANT		
				REFS:	139 926/P 5675/P 5738/P	

13.614738 SECONDS CPU TIME for ASSEMBLY

*** REFERENCE ABBREVIATIONS : M=modify, A=attribute, S=subscript, I=I/O ref, R=read, W=write, P=parameter