```
SERIES-III PL/M-86 X19B COMPILATION OF MODULE MMXONOFF
OBJECT MODULE PLACED IN :F1:MMXTST.OBJ
COMPILER INVOKED BY:  PLM86.86 :F1:MMXTST.P86 XREF


   1          MMXON$OFF:
              DO;
              $include(:f1:mmxp88.lit)
        =     /*****************************************************************************
        =     *                                                                           *
        =     *                      INCLUDE FILE MMXPRT.LIT                              *
        =     *                      =======================                              *
        =     *                                                                           *
        =     *   THIS INCLUDE FILE DEFINES THE MMX, MIP AND GMF SYSTEM PORT NAMES        *
        =     *                                                                           *
        =     *   USED IN THE MSP OPERATING SYSTEM. INCLUDE THIS FILE IN YOUR CODE        *
        =     *                                                                           *
        =     *   TO MAKE IT INDEPENDENT OF CHANGES IN PORT NAMES OR DEVICES.             *
        =     *                                                                           *
        =     *****************************************************************************/
        =              /* SPU-0 IS DEVICE 0 IT CURRENTLY HAS 13 SYSTEM      */
        =              /* PORTS USED AS SHOWN BELOW                         */

   2    1  =     DECLARE

        =        XTH$RETURN$PORT       LITERALLY  '000H', /* XTH BIOS RECEIVE PORT        */
        =        CXU$RETURN$PORT1      LITERALLY  '001H', /* CXU RECEIVE PORT1   */
        =        CXU$RETURN$PORT2      LITERALLY  '002H', /* CXU RECEIVE PORT2    */
        =        TMC                   LITERALLY  '003H', /* TAPS MASTER CONTROLLER       */
        =        CM                    LITERALLY  '004H', /* TAPS CM                      */
        =        TAC                   LITERALLY  '005H', /* TAPS APPLICATION CONTROLLER  */
        =        AM1                   LITERALLY  '006H', /* APPLICATION MANAGER #1       */
        =        AM2                   LITERALLY  '007H', /* APPLICATION MANAGER #2       */
        =        AM3                   LITERALLY  '008H', /* APPLICATION MANAGER #3       */
        =        IO1                   LITERALLY  '009H', /* I/O MANAGER #1               */
        =        IO2                   LITERALLY  '00AH', /* I/O MANAGER #2               */
        =        TS1                   LITERALLY  '003H', /* TAPS SPARE PORT #1           */
        =        TS2                   LITERALLY  '00CH', /* TAPS SPARE PORT #2           */

        =              /* CXU IS DEVICE 1 IT CURRENTLY HAS 2 SYSTEM      */
        =              /* PORTS USED AS SHOWN                             */

        =        CXU$PORT1             LITERALLY  '100H', /* CXU  PORT1        */
        =        CXU$PORT2             LITERALLY  '101H', /* CXU  PORT2        */

        =              /* 544 IS DEVICE 2 IT HAS 1 SYSTEM PORT            */

        =        XTH$i544$PORT1        LITERALLY  '200H'; /* XTH BIOS XMIT PORT */
        =      /* XTH$i544$PORT1 SHOULD BE SET TO 200H IF 544 BOARD*/
               /*
                   RMX88 TASK TO TURN ON CXU YELLOW LED
               */

   3    1          DECLARE LOCATION LITERALLY 'POINTER';

   4    1          rq$wait: PROCEDURE( exchange$address, time$limit ) LOCATION EXTERNAL;
```

```
 5   2          DECLARE exchange$address LOCATION;
 6   2          DECLARE time$limit WORD;
 7   2      END rq$wait;

 8   1      rq$send: PROCEDURE( exchange$address, message$address ) EXTERNAL;
 9   2          DECLARE exchange$address LOCATION;
10   2          DECLARE message$address LOCATION;
11   2      END rq$send;

12   1      rq$ctsk: PROCEDURE( std$pointer ) EXTERNAL;
13   2          DECLARE std$pointer POINTER;
14   2      END rq$ctsk;

15   1      rq$cxch: PROCEDURE( exchange$address ) EXTERNAL;
16   2          DECLARE exchange$address LOCATION;
17   2      END rq$cxch;

18   1  cqactv:
           PROCEDURE (function$name, status$ptr) POINTER EXTERNAL;
19   2      DECLARE function$name  WORD,
                    status$ptr    POINTER;
20   2  END cqactv;


21   1      DECLARE task$links LITERALLY
                'link$forward       LOCATION,
                 link$back          LOCATION,
                 thread             LOCATION';

22   1      DECLARE task$links$a LITERALLY
                'link$forward       LOCATION,
                 link$back          LOCATION,
                 thread             LOCATION,
                 delay              WORD';

23   1      DECLARE task$descriptor LITERALLY 'STRUCTURE (
                task$links,
                delay               WORD,
                unused              WORD,
                exchange$address    LOCATION,
                sp                  LOCATION,
                marker              ADDRESS,
                priority            BYTE,
                status              BYTE,
                name$ptr            POINTER,
                ios$links           LOCATION,
                task$link           LOCATION,
                mask                BYTE)';

24   1      DECLARE task$desc$overlay LITERALLY 'STRUCTURE (
                task$links,
                message$address     LOCATION,
                exchange$address    LOCATION,
                sp                  LOCATION,
                marker              ADDRESS,
                priority            BYTE,
                status              BYTE,
```

```
                    name$ptr            POINTER,
                    ios$links           LOCATION,
                    task$link           LOCATION,
                    mask                BYTE)';

25  1       DECLARE task$descriptor$length LITERALLY '41';

26  1       DECLARE running     LITERALLY   '00010000B';
27  1       DECLARE ndp$task    LITERALLY   '00001000B';
28  1       DECLARE mask$int    LITERALLY   '00000100B';
29  1       DECLARE suspended   LITERALLY   '00000010B';
30  1       DECLARE delayed     LITERALLY   '00000001B';

31  1       DECLARE bottom$flag LITERALLY   '0c7c7h';
32  1       DECLARE unused$flag LITERALLY   '0c7h';

33  1       DECLARE static$task$descriptor LITERALLY
                'STRUCTURE (
                    name(6)     BYTE,
                    pc          POINTER,
                    sp          LOCATION,
                    stklen      WORD,
                    ds          POINTER,
                    priority    BYTE,
                    exchange$address    LOCATION,
                    task$ptr    LOCATION,
                    task$ndp    BYTE)';

34  1       DECLARE ndp$used LITERALLY '1';

35  1       DECLARE msg$hdr LITERALLY
                'link               LOCATION,
                length              WORD,
                type                BYTE,
                home$exchange       LOCATION,
                response$exchange   LOCATION';

36  1       DECLARE msg$descriptor LITERALLY 'STRUCTURE(
                msg$hdr,
                remainder(1)    BYTE)';

37  1       DECLARE message MSG$DESCRIPTOR;

38  1       DECLARE exchange$descriptor LITERALLY 'STRUCTURE (
                message$head    LOCATION,
                message$tail    LOCATION,
                task$head       LOCATION,
                task$tail       LOCATION,
                exchange$link   LOCATION)';

39  1       DECLARE sleep  EXCHANGE$DESCRIPTOR EXTERNAL;
40  1       DECLARE msgex  EXCHANGE$DESCRIPTOR EXTERNAL;

41  1       DECLARE switches BYTE EXTERNAL;

42  1   MMXTST: PROCEDURE PUBLIC;
```

```
43   2          declare (i, status$byte) byte;
44   2          declare mmx$ex$ptr  pointer;
45   2          declare mmx$msg$ptr pointer;
46   2          declare msg$ptr pointer;

47   2          DECLARE mmx$msg BASED mmx$msg$ptr MSG$DESCRIPTOR;

48   2          declare time$out literally '3';
49   2          declare TRUE literally '1';
50   2          declare FALSE literally '0';


51   2          mmx$ex$ptr = cqactv( CXU$PORT1, @status$byte);

52   2          DO WHILE 1;
                /*
                wait forever for MMX message
                */
53   3              mmx$msg$ptr= rqwait(mmx$ex$ptr, 0);

54   3              switches= mmx$msg.remainder(0);

                /*
                    SYNC UP WITH LEDON TASK (wait til it's ready)
                */
55   3              msg$ptr= rqwait(@MSGEX, 0);
                /*
                    SEND MESSAGE TO LED TASK
                */
56   3                  CALL rqsend(@SLEEP,msg$ptr);
57   3              end;
58   2      end MMXTST;

59   1      end;
```

```
 DEFN  ADDR    SIZE  NAME, ATTRIBUTES, AND REFERENCES
-----  ------  ----  ---------------------------------

    2                 AM1. . . . . .    LITERALLY '006H'
    2                 AM2. . . . . .    LITERALLY '007H'
    2                 AM3. . . . . .    LITERALLY '008H'
   31                 BOTTOMFLAG . . .  LITERALLY '0c7c7h'
    2                 CM . . . . . .    LITERALLY '004H'
   18  0000H          CQACTV . . . . .  PROCEDURE POINTER EXTERNAL(4) STACK=0000H       51
    2                 CXUPORT1 . . . .  LITERALLY '100H'        51
    2                 CXUPORT2 . . . .  LITERALLY '101H'
    2                 CXURETURNPORT1 .  LITERALLY '001H'
    2                 CXURETURNPORT2 .  LITERALLY '002H'
   30                 DELAYED. . . . .  LITERALLY '00000001B'
   16  0000H      2   EXCHANGEADDRESS. .  POINTER IN PROC (RQCXCH) PARAMETER         16
    9  0000H      2   EXCHANGEADDRESS. .  POINTER IN PROC (RQSEND) PARAMETER          9
    5  0000H      2   EXCHANGEADDRESS. .  POINTER IN PROC (RQWAIT) PARAMETER          5
   38                 EXCHANGEDESCRIPTOR .  LITERALLY 'STRUCTURE (message$headLOCATION,message$tailLOCATION,t
                       -ask$headLOCATION,task$tailLOCATION,exchange$linkLOCATION)'
                         39   40
   50                 FALSE. . . . . .  LITERALLY '0'  IN PROC (MMXTST)
   19  0000H      2   FUNCTIONNAME . . .  WORD IN PROC (CQACTV) PARAMETER            19
   43  0010H      1   I. . . . . .      BYTE IN PROC (MMXTST)
    2                 I01. . . . . .    LITERALLY '009H'
    2                 I02. . . . . .    LITERALLY '00AH'
    3                 LOCATION . . . .  LITERALLY 'POINTER'          4    5    9   10   16   37   39
                         40   47
   28                 MASKINT. . . . .  LITERALLY '000001000B'
   37  0000H     10   MESSAGE. . . . .  STRUCTURE
       0000H      2       LINK . . . .  POINTER
       0002H      2       LENGTH . . .  WORD
       0004H      1       TYPE . . . .  BYTE
       0005H      2       HOMEEXCHANGE .  POINTER
       0007H      2       RESPONSEEXCHANGE  POINTER
       0009H      1       REMAINDER. . .  BYTE ARRAY(1)
    9  0000H      2   MESSAGEADDRESS . .  POINTER IN PROC (RQSEND) PARAMETER         10
   44  000AH      2   MMXEXPTR . . . .  POINTER IN PROC (MMXTST)       51*   53
   47  0000H     10   MMXMSG . . . . .  STRUCTURE BASED(MMXMSGPTR) IN PROC (MMXTST)
       0000H      2       LINK . . . .  POINTER
       0002H      2       LENGTH . . .  WORD
       0004H      1       TYPE . . . .  BYTE
       0005H      2       HOMEEXCHANGE .  POINTER
       0007H      2       RESPONSEEXCHANGE  POINTER
       0009H      1       REMAINDER. . .  BYTE ARRAY(1)          54
   45  000CH      2   MMXMSGPTR. . . . .  POINTER IN PROC (MMXTST)       53*   54
       0000H          MMXONOFF . . . .  PROCEDURE STACK=0000H
   42  0000H     68   MMXTST . . . . .  PROCEDURE PUBLIC STACK=0008H
   36                 MSGDESCRIPTOR. . .  LITERALLY 'STRUCTURE(msg$hdr,remainder(1)BYTE)'        37   47
   40  0000H     10   MSGEX. . . . . .  STRUCTURE EXTERNAL(6)          55
       0000H      2       MESSAGEHEAD. .  POINTER
       0002H      2       MESSAGETAIL. .  POINTER
       0004H      2       TASKHEAD . . .  POINTER
       0006H      2       TASKTAIL . . .  POINTER
       0008H      2       EXCHANGELINK .  POINTER
   35                 MSGHDR . . . . .  LITERALLY 'linkLOCATION,lengthWORD,typeBYTE,home$exchan
```

```
                                        -geLOCATION,response$exchangeLOCATION'          37   47
     46  000EH     2  MSGPTR . . . . . .  POINTER IN PROC (MMXTST)      55*   56
     27             NDPTASK. . . . . .  LITERALLY '000010000B'
     34             NDPUSED. . . . . .  LITERALLY '1'
     12  0000H      RQCTSK . . . . . .  PROCEDURE EXTERNAL(2) STACK=0000H
     15  0000H      RQCXCH . . . . . .  PROCEDURE EXTERNAL(3) STACK=0000H
      8  0000H      RQSEND . . . . . .  PROCEDURE EXTERNAL(1) STACK=0000H         56
      4  0000H      RQWAIT . . . . . .  PROCEDURE POINTER EXTERNAL(0) STACK=0000H       53   55
     26             RUNNING. . . . . .  LITERALLY '000100008'
     39  0000H   10 SLEEP. . . . . .  STRUCTURE EXTERNAL(5)         56
         0000H    2     MESSAGEHEAD. . .  POINTER
         0002H    2     MESSAGETAIL. . .  POINTER
         0004H    2     TASKHEAD . . . .  POINTER
         0006H    2     TASKTAIL . . . .  POINTER
         0008H    2     EXCHANGELINK . .  POINTER
     33             STATICTASKDESCRIPTOR  LITERALLY 'STRUCTURE (name(6)BYTE,pcPOINTER,spLOCATIO
                                        -N,stklenWORD,dsPOINTER,priorityBYTE,exchange$addr
                                        -essLOCATION,task$ptrLOCATION,task$ndpBYTE)'
     43  0011H     1  STATUSBYTE . . . . .  BYTE IN PROC (MMXTST)         51
     19  0000H     2  STATUSPTR. . . . . .  POINTER IN PROC (CQACTV) PARAMETER       19
     13  0000H     2  STDPOINTER . . . . .  POINTER IN PROC (RQCTSK) PARAMETER       13
     29             SUSPENDED. . . . . .  LITERALLY '00C00010B'
     41  0000H     1  SWITCHES . . . . . .  BYTE EXTERNAL(7)       54*
      2             TAC. . . . . . . . .  LITERALLY '005H'
OCATION,markerADDRESS,PRIORITY,statusBYTE,nameSTRUCTURE(6)BYTE,pcPOINTER,spLOCATION,maskBYTE,exchange$addressLOCATION,spL
TION,markerADDRESS,priORITY,statusBYTE,nameSTRUCTURE(6)BYTE,pcPOINTER,spLOCATION,maskBYTE,exchange$addressLOCATION,spLOCA
     25             TASKDESCRIPTORLENGTH  LITERALLY '41'
     21             TASKLINKS. . . . . .  LITERALLY 'link$forwardLOCATION, link$backLOCATION, threadLO
                                        -CATION'
     22             TASKLINKSA . . . . .  LITERALLY 'link$forwardLOCATION, link$backLOCATION, threadLO
                                        -CATION, delayWORD'
      5  0000H     2  TIMELIMIT. . . . . .  WORD IN PROC (RQWAIT) PARAMETER          6
     48             TIMEOUT. . . . . . .  LITERALLY '3'  IN PROC (MMXTST)
      2             TMC. . . . . . . . .  LITERALLY '003H'
     49             TRUE . . . . . . . .  LITERALLY '1'  IN PROC (MMXTST)
      2             TS1. . . . . . . . .  LITERALLY '00BH'
      2             TS2. . . . . . . . .  LITERALLY '00CH'
     32             UNUSEDFLAG . . . . .  LITERALLY '0c7h'
      2             XTHI544PORT1 . . . .  LITERALLY '200H'
      2             XTHRETURNPORT. . . .  LITERALLY '000H'
```

MODULE INFORMATION:

```
     CODE AREA SIZE      = 0044H     68D
     CONSTANT AREA SIZE  = 0000H      0D
     VARIABLE AREA SIZE  = 0012H     18D
     MAXIMUM STACK SIZE  = 0008H      8D
     200 LINES READ
     0 PROGRAM WARNINGS
     0 PROGRAM ERRORS

END OF PL/M-86 COMPILATION
```