```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE KAOS_RQPROC_A
OBJECT MODULE PLACED IN :F1:RQPRCC.OBJ
INVOCATION LINE CONTROLS:  DEBUG


LOC  OBJ                  LINE     SOURCE

                             1 +1  $TITLE('KAOS MIP Request Queue Routines - REVISED 03/03/82')
                             2
                             3             NAME    KAOS_RQPROC_A
                             4 +1  $include(:f1:propa.lit)
                       =1    5     ;
                       =1    6     ;   Intel Corporation Proprietary Information. This listing is
                       =1    7     ;   supplied under the terms of a license agreement with Intel
                       =1    8     ;   Corporation and may not be copied nor disclosed except in
                       =1    9     ;   accordance with the terms of the agreement.
                       =1   10     ;
                            11
                            12 +1  $include (:f1:mip.equ)
                       =1   13     ;
                       =1   14     ; DEFINE RQD RESULTS
                       =1   15     ;
  0001                 =1   16     GERROR  EQU     1H
  0004                 =1   17     GBUSY   EQU     4H
  0008                 =1   18     FIRSTG  EQU     8H
  0010                 =1   19     GDISAB  EQU     10H
  0020                 =1   20     GFULL   EQU     20H
  0040                 =1   21     DISABT  EQU     40H
  0080                 =1   22     FULLF   EQU     80H
                       =1   23
  0001                 =1   24     TERROR  EQU     1H
  0004                 =1   25     TBUSY   EQU     4H
  0008                 =1   26     FIRSTT  EQU     8H
  0010                 =1   27     TDISAB  EQU     10H
  0020                 =1   28     TEMPTY  EQU     20H
  0040                 =1   29     DISABG  EQU     40H
  0080                 =1   30     EMPTYF  EQU     80H
                       =1   31     ;
                       =1   32     ; DEFINE MIP CMDS AND RESPONSES
                       =1   33     ;
  0070                 =1   34     CSEND   EQU     70H
  0080                 =1   35     SENTOK  EQU     80H
  0081                 =1   36     UNKNP   EQU     81H
  0083                 =1   37     ACTIVP  EQU     83H
  0085                 =1   38     INSUFM  EQU     85H
  0087                 =1   39     INACTP  EQU     87H
  0089                 =1   40     DEADP   EQU     89H
                       =1   41     ;
                       =1   42     ; DEFINE MIP-ISIS PARAMETERS
                       =1   43     ;
  0000                 =1   44     MYIDS   EQU     0
  0003                 =1   45     THIDEV  EQU     3
                            46
                            47     CGROUP  GROUP   CODE
----                        48     CODE    SEGMENT BYTE PUBLIC 'CODE'
                            49             ASSUME  CS:CGROUP
                            50     ;
```

```
LOC  OBJ                  LINE     SOURCE

                            51     ;
                            52     ; REQUEST POINTER ROUTINES
                            53     ;
                            54             PUBLIC  REQUESTGIVEPTR,REQUESTTAKEPTR
0000                        55     REQUESTGIVEPTR:
0000 E8D700                 56             CALL    GIVE_LOAD
0003 EB03                   57             JMP     SHORT L1
                            58
0005                        59     REQUESTTAKEPTR:
0005 E8E300                 60             CALL    TAKE_LOAD
                            61     ;
                            62     ; CHECK IF DISABLED
                            63     ;
0008 50                     64     L1:     PUSH    AX
0009 268A4401               65             MOV     AL,ES:[SI+1]    ; GET STATE/
000D 260A4501               66             OR      AL,ES:[DI+1]    ; TAKESTATE
0011 2450                   67             AND     AL,DISABG OR GDISAB
0013 58                     68             POP     AX
0014 7403                   69             JZ      L4
0016 B011                   70             MOV     AL,GDISAB OR GERROR ; IS DISABLED, RETURN
0018 C3                     71             RET             ; ZERO FLAG IS RESET
                            72     ;
                            73     ; WAS NOT DISABLED, SEE IF FULL/EMPTY
                            74     ;
0019 3AD0                   75     L4:     CMP     DL,AL           ;\OTHER INDEX ONLY PART IN BL
001B 750F                   76             JNE     L6
001D 263235                 77             XOR     DH,ES:[DI]      ; INDEX PARTS ARE EQUAL, LOOK AT STATE PARTS
0020 80E680                 78             AND     DH,80H
0023 3AF4                   79             CMP     DH,AH           ; DH = 0 IF EQUAL, 80H IF NOT EQUAL
0025 7505                   80             JNE     L6              ; AH = 80 IF GIVE, AH = 0H IF TAKE
0027 B021                   81             MOV     AL,GFULL OR GERROR ; IS FULL/EMPTY
0029 0AC0                   82             OR      AL,AL           ; WILL RESET ZERO FLAG
002B C3                     83             RET             ; ZERO FLAG IS RESET
                            84     ;
                            85     ; HAVE ENTRY, CALCULATE ADDRESS OF IT
                            86     ;
002C 8ACD                   87     L6:     MOV     CL,CH           ; GET RQESIZE
002E 81E27F0C               88             AND     DX,7FH          ; GET INDEX PART ONLY
0032 D3E2                   89             SHL     DX,CL
0034 8CC1                   90             MOV     CX,ES           ; MOVE BASE OF RQDPTR
0036 83C208                 91             ADD     DX,8            ; ADD RQD AREA
0039 03DA                   92             ADD     BX,DX           ; ADD ENTRY OFFSET
003B 7304                   93             JNC     L8
003D 81C1001C               94             ADD     CX,1OCCH        ; HAD OVERFLOW IN OFFSET, ADD TO BASE
                            95     ;
                            96     ; ALL DONE, RETURN ENTRY TO USER
                            97     ;
0041 8EC1                   98     L8:     MOV     ES,CX
0043 33C0                   99             XOR     AX,AX           ; ZERO FLAG IS SET
0045 C3                    100             RET
                           101
                           102 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           103     ;
                           104     ; RELEASE POINTER ROUTINES
                           105     ;
                           106             PUBLIC  RELEASEGIVEPTR,RELEASETAKEPTR
                           107     ;
0046                       108     RELEASEGIVEPTR:
0046 57                    109             PUSH    DI
0047 E89000                110             CALL    GIVE_LOAD
004A EB04                  111             JMP     SHORT M1
                           112
004C                       113     RELEASETAKEPTR:
004C 57                    114             PUSH    DI                  ; SAVE PTR TO DEVICE INFO
004D E89B00                115             CALL    TAKE_LOAD
                           116     ;
                           117     ; BUMP POINTERS AND UPDATE STATUS/INDEX
                           118     ;
0050 FEC9                  119     M1:     DEC     CL                  ; GET MASK FOR MODULO ARITHMETIC
0052 FEC2                  120             INC     DL
0054 22D1                  121             AND     DL,CL               ; DO MODULO
0056 3AD0                  122             CMP     DL,AL               ; SEE IF GI=TI
0058 750D                  123             JNE     M6                  ; JMP IF INDEXES ARE NOT EQUAL
005A 268A05                124             MOV     AL,ES:[DI]          ; AL = TS IF GIVE , GS IF TAKE
005D 0AE4                  125             OR      AH,AH
005F 7402                  126             JZ      M5                  ; JMP IF TAKE PTR
0061 F6D0                  127             NOT     AL                  ; IS GIVE, INVERT TS TO GET GS
0063 2480                  128     M5:     AND     AL,80H              ; ISOLATE NEW ITEM
0065 8AF0                  129             MOV     DH,AL               ; MOVE NEW STATE TO DH
                           130     ;
                           131     ; STORE INDEX,STATE
                           132     ;
0067 0AF2                  133     M6:     OR      DH,DL               ; FORM NEW COMPOSITE VARIABLE
0069 268834                134             MOV     BYTE PTR ES:[SI],DH     ; AND STORE IT
                           135     ;
                           136     ; SEE IF FIRST TAKE/GIVE
                           137     ;
006C 268A05                138             MOV     AL,ES:[DI]          ; LOAD OTHER INDEX/STATE
006F FEC0                  139             INC     AL
0071 22C1                  140             AND     AL,CL               ; DO MOD ARITHMET
0073 3AC2                  141             CMP     AL,DL
0075 5F                    142             POP     DI                  ; PTR TO DEVICE INFO
0076 7553                  143             JNZ     M20                 ; NOT FIRST GIVE/TAKE
0078 0AE4                  144             OR      AH,AH
007A 7406                  145             JZ      SHORT M10
007C 26C60701              146             MOV     BYTE PTR ES:[BX],1H             ; SET GIVE FLAG
0080 EB08                  147             JMP     SHORT M12
0082 C47505                148     M10:    LES     SI,DWORD PTR [DI+5]
0085 26C6440180            149             MOV     BYTE PTR ES:[SI+1],80H  ; SET FIRST TAKE FLAG
                           150     ;
                           151     ;       NOW GENERATE INTERRUPT
                           152     ;
                           153     ;  do case DI.Int$type;
008A 8A4D09                154     M12:    MOV     CL,BYTE PTR[DI+9H]      ; INT TYPE
008D B500                  155             MOV     CH,0
008F 03C9                  156             ADD     CX,CX
0091 8D36CE0C      R       157             LEA     SI,WORD PTR CGROUP:@19
```

```
LOC  OBJ                  LINE     SOURCE

0095 03F1                  158              ADD     SI,CX
0097 2E8B0C                159              MOV     CX,WORD PTR CS:[SI]
009A 8E450B                160              MOV     ES,WORD PTR [DI+0BH]    ; INTADR
009D B002                  161              MOV     AL,2
009F 33F6                  162              XOR     SI,SI
00A1 FFE1                  163              JMP     CX
                           164
                           165      ;     ; /* no interrupt */
                           166
                           167      ;     do; /* IO mapped interrupt */
00A3                       168      @3:
                           169      ;       Output(Di.Intadr) = 2H;
00A3 8CC2                  170              MOV     DX,ES
00A5 BB00F4                171              MOV     BX,0F400H               ; SET IO ADDRESSES TO MULTIBUS
00A8 8EC3                  172              MOV     ES,BX
00AA FA                    173              CLI                             ; INTERRUPT HERE IS A DISASTER
00AB 268804                174              MOV     BYTE PTR ES:[SI],AL     ; DO IT VIA WRITE TO SPECIAL AREA
00AE EE                    175              OUT     DX,AL
00AF BE0020                176              MOV     SI,2000H
00B2 268804                177              MOV     BYTE PTR ES:[SI],AL     ; FLIP BACK TO COMM BOARD PORTS
00B5 FB                    178              STI
                           179      ;     end;
00B6 EB13                  180      @1:     JMP     SHORT M20
                           181
                           182      ;     do; /* memory mapped interrupt */
00B8                       183      @4:
                           184      ;       Mem$map$ptr$o = 0;
                           185      ;       Mem$map$ptr$b = Di.Intadr;
                           186
                           187      ;       Mem$map = 2H;
00B8 268804                188              MOV     BYTE PTR ES:[SI],AL
                           189      ;     end;
00BB EB0E                  190              JMP     SHORT M20
                           191
                           192      ;     do;  /* edge/level */
00BD                       193      @5:
                           194      ;      Output(9CH) = 0 ; /* clear it */
00BD E690                  195              OUT     9CH,AL
00BF E680                  196      @7:     OUT     8CH,AL          ; ENTRY FOR PURE LEVEL
00C1 EB08                  197              JMP     SHORT M20
                           198      ;      Output(8CH) = 0 ; /* set it */
                           199      ;     end;
                           200
                           201      ;     do; /* pure level */
                           202      ;      Output(8CH) = 0;  /* raise high and leave it there for awhile
00C3 E680                  203      @6:     OUT     8CH,AL
00C5 FEC0                  204              INC     AL
00C7 75FA                  205              JNZ     @6
00C9 E65A                  206              OUT     9C,AL
                           207      ;     end;
                           208      ;   end;
00CB 33C0                  209      M20:    XOR     AX,AX
00CD C3                    210              RET
                           211
00CE                       212      @19:
```

```
LOC  OBJ                  LINE     SOURCE

00CE B600           R      213              DW      CGROUP:@1
00D0 A300           R      214              DW      CGROUP:@3
00D2 B800           R      215              DW      CGROUP:@4
00D4 BD00           R      216              DW      CGROUP:@5
00D6 C300           R      217              DW      CGROUP:@6
00D8 BF00           R      218              DW      CGROUP:@7
                           219
                           220 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           221     ;
                           222     ; THIS ROUTINE LOADS THE ADDRESSES OF THE RQD ENTRIES FOR GIVE POINTER
                           223     ; FUNCTIONS. IT ALSO LOADS RQE/RQ SIZE AND SETS AH TO INDICATE GIVE
                           224     ;
00DA                       225     GIVE_LOAD:
00DA C45D05                226             LES     BX,DWORD PTR [DI+5]
00DD 8D7704                227             LEA     SI,[BX+4]
00E0 8D7F06                228             LEA     DI,[BX+6]
00E3 268B4F02              229             MOV     CX,ES:[BX+2]
00E7 B480                  230             MOV     AH,80H
00E9 EB0F                  231             JMP     SHORT LOAD_REG
                           232
                           233     ;
                           234     ; THIS ROUTINE DOES THE SAME THING AS THE ONE ABOVE EXCEPT FOR TAKE
                           235     ;
00EB                       236     TAKE_LOAD:
00EB C45D01                237             LES     BX,DWORD PTR [DI+1]
00EE 8D7706                238             LEA     SI,[BX+6]
00F1 8D7F04                239             LEA     DI,[BX+4]
00F4 268B4F02              240             MOV     CX,ES:[BX+2]
00F8 32E4                  241             XOR     AH,AH
                           242
                           243     ;
                           244     ; THIS ROUTINE LOADS REGISTERS WITH THE VALUES OF RQD CONTROL VARIABLES
                           245     ;
00FA                       246     LOAD_REG:
00FA 268A14                247             MOV     DL,ES:[SI]      ; MAIN INDEX/STATE
00FD 8AF2                  248             MOV     DH,DL
00FF 80E680                249             AND     DH,80H          ; STATE
0102 80E27F                250             AND     DL,7FH          ; INDEX
0105 268A05                251             MOV     AL,ES:[DI]      ; OTHER STATE/INDEX
0108 247F                  252             AND     AL,7FH          ; GET INDEX PART ONLY
010A C3                    253             RET
                           254
----                       255     CODE    ENDS
                           256
                           257             END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```