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


LOC  OBJ                  LINE     SOURCE

                             1 +1  $TITLE ('INITIALIZATION TIME CREATED OBJECTS')
                             2 +1  $NOLIST INCLUDE(:F1:SYSGEN.MAC)
                           111             NAME    ROMOBJLST
                           112
                           113 +1  $INCLUDE(:F1:PROPA.LIT)
                     =1    114     ;
                     =1    115     ;   Intel Corporation Proprietary Information. This listing is
                     =1    116     ;   supplied under the terms of a license agreement with Intel
                     =1    117     ;   Corporation and may not be copied nor disclosed except in
                     =1    118     ;   accordance with the terms of the agreement.
                     =1    119     ;
                           120     ;
                           121     ;  THIS MODULE CONTAINS DEFINITIONS OF CMX OBJECTS CREATED
                           122     ;  AT SYSTEM STARTUP TIME.
                           123     ;
                           124 +1
                           125 +1  CGROUP  GROUP   CODE
                           126 +1  DGROUP  GROUP   DATA,STACKS
                           127 +1
                           128
                           129 +1
----                       130 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
                           131 +2          PUBLIC  STARTINGLIST
  0000                     132 +2  STARTINGLIST    EQU     $
----                       133 +1  CODE    ENDS
                           134 +1
                           135
                           136 +1
----                       137 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
0000 0A                    138 +1          DB      10
----                       139 +1  CODE    ENDS
----                       140 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
                           141 +1          PUBLIC  READYLIST,READYLISTSIZE
0000 (16                   142 +2  READYLIST       DW      16 DUP (?,?,?)
     ????
     ????
     ????
     )
  0010                     143 +2  READYLISTSIZE   EQU     16
----                       144 +1  DATA    ENDS
                           145
                           146 +1
                           147 +2          EXTRN   CQMIPTASK:FAR
----                       148 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
0001 02                    149 +1          DB      2
0002 6000          R       150 +2          DW      OFFSET DGROUP:PCB_CQMIPTASK
0004 0200                  151 +2          DW      2
0006 0000          E       152 +2          DW      OFFSET CGROUP:CQMIPTASK
0008 6000          R       153 +2          DW      OFFSET DGROUP:STK_CQMIPTASK
----                       154 +1  CODE    ENDS
```

objects

process

PCB
Priority
CS
DS/stack offset

```
LOC  OBJ                  LINE     SOURCE

----                       155 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
0060 (7                    156 +2  PCB_CQMIPTASK   DW      7 DUP (?)
     ????
     )
----                       157 +1  DATA    ENDS
----                       158 +1  STACKS  SEGMENT BYTE PUBLIC 'STACKS'          ;CREATE STACK
0000 (48                   159 +2          DW      3CH DUP (?)
     ????
     )
  0060                     160 +2  STK_CQMIPTASK   EQU $
----                       161 +1  STACKS  ENDS
                           162 +1
                           163 +2          EXTRN   CQBOOTTASK:FAR
----                       164 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
000A 02                    165 +1          DB      2
000B 6E00            R     166 +2          DW      OFFSET DGROUP:PCB_CQBOOTTASK
000D 0E00                  167 +2          DW      14
000F 0000            E     168 +2          DW      OFFSET CGROUP:CQBOOTTASK
0011 C000            R     169 +2          DW      OFFSET DGROUP:STK_CQBOOTTASK
----                       170 +1  CODE    ENDS
----                       171 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
006E (7                    172 +2  PCB_CQBOOTTASK  DW      7 DUP (?)
     ????
     )
----                       173 +1  DATA    ENDS
----                       174 +1  STACKS  SEGMENT BYTE PUBLIC 'STACKS'          ;CREATE STACK
0060 (48                   175 +2          DW      30H DUP (?)
     ????
     )
  00C0                     176 +2  STK_CQBOOTTASK  EQU $
----                       177 +1  STACKS  ENDS
                           178 +1
                           179 +2          EXTRN   IDLE:FAR
----                       180 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
C013 02                    181 +1          DB      2
0014 7C00            R     182 +2          DW      OFFSET DGROUP:PCB_IDLE
0016 0F00                  183 +2          DW      15
0018 0000            E     184 +2          DW      OFFSET CGROUP:IDLE
001A FC00            R     185 +2          DW      OFFSET DGROUP:STK_IDLE
----                       186 +1  CODE    ENDS
----                       187 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
007C (7                    188 +2  PCB_IDLE        DW      7 DUP (?)
     ????
     )
----                       189 +1  DATA    ENDS
----                       190 +1  STACKS  SEGMENT BYTE PUBLIC 'STACKS'          ;CREATE STACK
00C0 (30                   191 +2          DW      1EH DUP (?)
     ????
     )
  00FC                     192 +2  STK_IDLE        EQU $
----                       193 +1  STACKS  ENDS
                           194
                           195 +1
                           196 +2          PUBLIC  CQBOOTCMDMB
----                       197 +1  CODE    SEGMENT BYTE    PUBLIC 'CODE'
```

```
LOC  OBJ                  LINE     SOURCE

001C 06                    198 +1          DB      6
001D 8A00              R   199 +2          DW      OFFSET DGROUP:CQBOOTCMDMB
----                       200 +1  CODE    ENDS
----                       201 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
008A (8                    202 +2  CQBOOTCMDMB     DW      8 DUP (?)
     ????
     )
----                       203 +1  DATA    ENDS
                           204 +1
                           205 +2          PUBLIC  CQREMOTEWAITINGMB
----                       206 +1  CODE    SEGMENT BYTE    PUBLIC 'CODE'
001F 06                    207 +1          DB      6
0020 9A00              R   208 +2          DW      OFFSET DGROUP:CQREMOTEWAITINGMB
----                       209 +1  CODE    ENDS
----                       210 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
C09A (8                    211 +2  CQREMOTEWAITINGMB       DW      8 DUP (?)
     ????
     )
----                       212 +1  DATA    ENDS
                           213 +1
                           214 +2          PUBLIC  CQMIPSENDWTMBX
----                       215 +1  CODE    SEGMENT BYTE    PUBLIC 'CODE'
0022 06                    216 +1          DB      6
0023 AA00              R   217 +2          DW      OFFSET DGROUP:CQMIPSENDWTMBX
----                       218 +1  CODE    ENDS
----                       219 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
00AA (8                    220 +2  CQMIPSENDWTMBX  DW      8 DUP (?)
     ????
     )
----                       221 +1  DATA    ENDS
                           222 +1
                           223 +2          PUBLIC  CQMIPREMOTEMBX
----                       224 +1  CODE    SEGMENT BYTE    PUBLIC 'CODE'
0025 06                    225 +1          DB      6
0026 BA00              R   226 +2          DW      OFFSET DGROUP:CQMIPREMOTEMBX
----                       227 +1  CODE    ENDS
----                       228 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
C0BA (8                    229 +2  CQMIPREMOTEMBX  DW      8 DUP (?)
     ????
     )
----                       230 +1  DATA    ENDS
                           231
                           232 +1
----                       233 +1  CODE    SEGMENT BYTE    PUBLIC 'CODE'
                           234 +2          PUBLIC  CQMIPUSEPERMIT
0028 04                    235 +1          DB      4
0029 CA00              R   236 +2          DW      OFFSET DGROUP:CQMIPUSEPERMIT
002B 0100                  237 +2          DW      1
----                       238 +1  CODE    ENDS
----                       239 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
C0CA (4                    240 +2  CQMIPUSEPERMIT  DW      4 DUP (?)
     ????
     )
----                       241 +1  DATA    ENDS
                           242
```

mailbox

1st MBX used

mailbox

mailbox

MIP semaphore

```
LOC  OBJ                  LINE     SOURCE

                           243 +1
                           244 +2          PUBLIC  CQWAITACB
----                       245 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
002D 08                    246 +1          DB      8
002E ----            R     247 +2          DW      SEG DGROUP:CQWAITACB
0030 D200            R     248 +2          DW      OFFSET DGROUP:CQWAITACB
----                       249 +1  CODE    ENDS
----                       250 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
00D2 (8                    251 +2  CQWAITACB       DW      8 DUP (?)
     ????
     )
----                       252 +1  DATA    ENDS
                           253 +1
                           254 +2          PUBLIC  CQMIPSENDACB
----                       255 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
0032 08                    256 +1          DB      8
0033 ----            R     257 +2          DW      SEG DGROUP:CQMIPSENDACB
0035 E200            R     258 +2          DW      OFFSET DGROUP:CQMIPSENDACB
----                       259 +1  CODE    ENDS
----                       260 +1  DATA    SEGMENT BYTE PUBLIC 'DATA'
00E2 (8                    261 +2  CQMIPSENDACB    DW      8 DUP (?)
     ????
     )
----                       262 +1  DATA    ENDS
                           263
                           264 +1
----                       265 +1  CODE    SEGMENT BYTE PUBLIC 'CODE'
0037 00                    266 +1          DB      0
----                       267 +1  CODE    ENDS
                           268 +1
                           269
                           270 +1  $TITLE('SPECIAL JUMP ROUTINES')
                           271
                               ;
                               ; THIS MODULE CONTAINS VARIOUS ROUTINES TO CALL OR JUMP TO
                               ; ADDRESSES/ROUTINES THAT ARE NOT ACCESSIBLE IN THE PLM-86
                               ; COMPACT MODEL.
                               ;
                           277
----                       278     CODE    SEGMENT BYTE PUBLIC 'CODE'
                           279             ASSUME  CS:CGROUP
                           280
                               ;
                               ; THIS ROUTINE DOES A LONG JUMP TO THE SUPPLIED ADDRESS.
                               ;
                           284             PUBLIC  LONGGOTO
0038                       285     LONGGOTO:
0038                       286     LP      PROC    FAR
0038 58                    287             POP     AX      ; GET RID OF SHORT CALL RETURN ADDRESS
0039 CB                    288             RET             ; RETURNS TO PARAMETER
                           289     LP      ENDP
                           290
                           291
                               ;
                               ; THIS ROUTINE DOES A SHORT CALL TO THE ADDRESS SUPPLIED ON THE
```

```
LOC  OBJ                  LINE     SOURCE

                           294     ; STACK. IT MAINTAINS THE COMPACT MODEL AND MAY BE USED WHEN
                           295     ; PARAMETERS ARE TO BE PASSED.
                           296     ;
                           297             PUBLIC  SHORTCALL
003A                       298     SHORTCALL:
003A 5B                    299             POP     BX      ; RET ADR
003B 58                    300             POP     AX      ; VALUE TO JUMP TO
003C 53                    301             PUSH    BX
003D 50                    302             PUSH    AX
003E C3                    303             RET
                           304
----                       305     CODE    ENDS
                           306
                           307             END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```