

/\*.....\*

SPOOL  
TCLGNL.LST  
05/04/82  
15:52:55

\*\*\*\*\*/

SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.1 ASSEMBLY OF MODULE TCLGEN  
OBJECT MODULE PLACED IN :F1:TCLGNL.OBJ  
ASSEMBLER INVOKED BY: ASM86.86 :F1:TCLGNL.A86 PRINT(:F1:TCLGNL.LST) DEBUG

```
LOC  OBJ          LINE    SOURCE
                                1 +1  $TITLE(iLNA Transport Control Layer Object Generation Macros    12:35)
                                2      NAME    TCLGEN
                                3 +1  $include (:f1:cpyrt.dca)
=1      4      ;
=1      5      ;          /* Intel Corporation Proprietary Information.
=1      6      ;          This listing is supplied under the terms of a
=1      7      ;          license agreement with Intel Corporaton and
=1      8      ;          may not be copied nor disclosed except in
=1      9      ;          accordance with the terms of that agreement. */
=1     10      ;
                                11 +1  $INCLUDE (:f1:SYSGEN.GEO)
=1     12      ;;;      (C) INTEL CORPORATION 1981. ALL RIGHTS RESERVED. NO PART
=1     13      ;          OF THIS PROGRAM OR PUBLICATION MAY BE REPRODUCED, TRANS-
=1     14      ;          MITTED, TRANSCRIBED, STORED IN A RETRIEVAL SYSTEM, OR
=1     15      ;          TRANSLATED INTO ANY LANGUAGE OR COMPUTER LANGUAGE, IN ANY
=1     16      ;          FORM OR BY ANY MEANS, ELECTRONIC, MECHANICAL, MAGNETIC,
=1     17      ;          OPTICAL, CHEMICAL, MANUAL OR OTHERWISE, WITHOUT THE PRIOR
=1     18      ;          WRITTEN PERMISSION OF INTEL CORPORATION, 3065 BOWERS AVENUE,
=1     19      ;          SANTA CLARA, CALIFORNIA, 95051, ATTN: SOFTWARE LICENSE
=1     20      ;          ADMINISTRATION.
=1     21      ;
=1     22      ;
=1     23      ;
=1     24      ;
=1     25      ;;;      GROUPDEF - DEFINE LINKER GROUPS
=1     26      ;
=1     27      ;          THIS MACRO DEFINES CGRUOP AND DGROUP.
=1     28      ;
=1     29      ;
=1     30      ;
=1     31      ;
=1     32      ;
=1     33      ;
=1     34      ;
=1     35      ;;;      OBJECTLIST (NAME) - START OBJECT LIST FOR "CREATELIST".
=1     36      ;
=1     37      ;          THIS MACRO GIVES A NAME TO AND MARKS THE BEGINNING OF
=1     38      ;          A LIST TO BE PASSED TO "CREATELIST".
=1     39      ;
=1     40      ;          PARAMETER:
=1     41      ;          NAME - THE NAME TO BE DECLARED PUBLIC AS THE NAME OF THIS LIST.
=1     42      ;
=1     43      ;
=1     44      ;
=1     45      ;
=1     46      ;
=1     47      ;
=1     48      ;
=1     49      ;;;      PROCESS (PRI, ENTRY, STACKSIZE) - CREATE PCB AND STACK FOR PROCESS.
=1     50      ;
```

```
LOC  OBJ          LINE    SOURCE
=1     51          ;      PARAMETERS:
=1     52          ;      PRI - THE PRIORITY OF THE PROCESS, ZERO BEING THE HIGHEST
=1     53          ;      AND SIZE-1 (FROM THE "READYLIST" MACRO) BEING THE LOWEST.
=1     54          ;      ENTRY - THE ENTRY NAME OF THE PROCESS. THIS SYMBOL
=1     55          ;      WILL BE DECLARED EXTERNAL.
=1     56          ;      STACKSIZE - THE NUMBER OF BYTES OF USER SPACE TO
=1     57          ;      CREATE IN THE STACK.
=1     58
=1     59
=1     60
=1     61
=1     62
=1     63
=1     64
=1     65          ;;;      SEMAPHORE (NAME, INITIAL) - CREATE SEMAPHORE.
=1     66          ;
=1     67          ;      PARAMETERS:
=1     68          ;      NAME - THE NAME OF THE SEMAPHORE. THIS NAME WILL BE
=1     69          ;      DECLARED PUBLIC.
=1     70          ;      INITIAL - THE INITIAL NUMBER OF SIGNALS ON THE SEMAPHORE.
=1     71
=1     72
=1     73
=1     74
=1     75
=1     76
=1     77
=1     78          ;;;      MAILBOX (NAME) - CREATE MAILBOX.
=1     79          ;
=1     80          ;      PARAMETER:
=1     81          ;      NAME - THE NAME OF THE MAILBOX. THIS NAME WILL BE
=1     82          ;      DECLARED PUBLIC.
=1     83
=1     84
=1     85
=1     86
=1     87
=1     88
=1     89
=1     90          ;;;      ALARM(NAME) - CREATE ALARM CONTROL BLOCK.
=1     91          ;
=1     92          ;      PARAMETERS:
=1     93          ;      SIZE - NUMBER OF PRIORITIES IN READY LIST.
=1     94
=1     95
=1     96
=1     97
=1     98
=1     99
=1    100
=1    101          ;;;      READYLIST - DEFINE READY LIST.
=1    102          ;
=1    103          ;      PARAMETERS:
=1    104          ;      SIZE - NUMBER OF PRIORITIES IN READY LIST.
=1    105
```

```

LOC  OBJ          LINE  SOURCE
-----
=1    106
=1    107
=1    108
=1    109
=1    110
=1    111
=1    112      ;;;      ENDLIST - MARK END OF LIST.
=1    113      ;
=1    114      ;      PARAMETERS:
=1    115      ;      NONE.
=1    116
=1    117
=1    118
119      ; Date 10/26/81 Take out logrbmbx
120      ; Date 09/30/81 Last modified
121      ; Date 09/01/81 Modified to reduce stack size
122      ; Date 09/24/81 added bufmipmbx, tbmsproc
123      ;      9/30/81 Dropped stack requirements for non-logging version
124      ;Transport Control Layer KAOS generation macros for TCL objects,
125      ; not test objects.
126      ; Note: duplicates of every stmt (without percent) inserted so they
127      ; will show on listing.
128      ;
129      ;
130      ;
131 +1  CGROUP  GROUP  CODE
132 +1  DGROUP  GROUP  DATA,STACKS
133 +1
134      ;
135 +1
136 +1  CODE      SEGMENT PUBLIC 'CODE'
137 +2          PUBLIC TCLOBJECTS
138 +2  TCLOBJECTS EQU    $
139 +1  CODE      ENDS
140 +1
141      ;
142      ;
143 +1
144 +2          EXTRN  RCVPROC:FAR
145 +1  CODE      SEGMENT PUBLIC 'CODE'
146 +1          DB      2
147 +2          DW      OFFSET DGROUP:PCB_RCVPROC
148 +2          DW      4
149 +2          DW      OFFSET CGROUP:RCVPROC
150 +2          DW      OFFSET DGROUP:STK_RCVPROC
151 +1  CODE      ENDS
152 +1  DATA      SEGMENT PUBLIC 'DATA'
153 +2  PCB_RCVPROC DW      7 DUP (?)
154 +1  DATA      ENDS
155 +1  STACKS     SEGMENT PUBLIC 'STACKS' ;CREATE STACK
156 +2  LIM_RCVPROC DW      2FH DUP (AAAAAH)
-----
0000 02
0001 0000      R
0003 0400
0005 0000      E
0007 5E00      R
-----
0000 (7
      ????)
      )
-----
0000 (47
      AAAA

```

GROUPDEF

OBJECTLIST(TCLOBJECTS)

28 bytes less than logging ve

PROCESS(4,RCVPROC,84)

```

LOC  OBJ                LINE  SOURCE
-----
)
005E                157 +2  STK_RCVPROC    EQU $
-----
158 +1  STACKS    ENDS
159      ;
                                36 bytes less than logging ve
                                PROCESS(5,TRANSPROC,82)
160      ;
161 +1
162 +2      EXTRN  TRANSPROC:FAR
163 +1  CODE     SEGMENT PUBLIC 'CODE'
164 +1      DB      2
0009 02          R      165 +2      DW      OFFSET DGROUP:PCB_TRANSPROC
000A 0E00          R      166 +2      DW      5
000C 0500          E      167 +2      DW      OFFSET CGROUP:TRANSPROC
000E 0000          R      168 +2      DW      OFFSET DGROUP:STK_TRANSPROC
0010 BA00          R      169 +1  CODE     ENDS
-----
170 +1  DATA   SEGMENT PUBLIC 'DATA'
000E (7          171 +2  PCB_TRANSPROC DW      7 DUP (?)
      ????)
)
-----
172 +1  DATA   ENDS
173 +1  STACKS  SEGMENT PUBLIC 'STACKS' ;CREATE STACK
005E (46          174 +2  LIM_TRANSPROC DW      2EH DUP (OAAAAH)
      AAAA)
)
00BA                175 +2  STK_TRANSPROC EQU $
-----
176 +1  STACKS  ENDS
177      ;
                                52 bytes less than logging ve
                                PROCESS(6,IPPROC,72)
178      ;
179 +1
180 +2      EXTRN  IPPROC:FAR
181 +1  CODE     SEGMENT PUBLIC 'CODE'
182 +1      DB      2
0012 02          R      183 +2      DW      OFFSET DGROUP:PCB_IPPROC
0013 1C00          R      184 +2      DW      6
0015 0600          E      185 +2      DW      OFFSET CGROUP:IPPROC
0017 0000          R      186 +2      DW      OFFSET DGROUP:STK_IPPROC
0019 0C01          R      187 +1  CODE     ENDS
-----
188 +1  DATA   SEGMENT PUBLIC 'DATA'
001C (7          189 +2  PCB_IPPROC   DW      7 DUP (?)
      ????)
)
-----
190 +1  DATA   ENDS
191 +1  STACKS  SEGMENT PUBLIC 'STACKS' ;CREATE STACK
00BA (41          192 +2  LIM_IPPROC   DW      29H DUP (OAAAAH)
      AAAA)
)
010C                193 +2  STK_IPPROC    EQU $
-----
194 +1  STACKS  ENDS
195      ;
                                34 bytes less than logging ve
                                PROCESS(7,TBMSPROC,66)
196      ;
197 +1
198 +2      EXTRN  TBMSPROC:FAR
199 +1  CODE     SEGMENT PUBLIC 'CODE'

```

LOC	OBJ	LINE	SOURCE
001B	02	200 +1	DB 2
001C	2A00	R 201 +2	DW OFFSET DGROUP:PCB_TBMSPROC
001E	0700	202 +2	DW 7
0020	0000	E 203 +2	DW OFFSET CGROUP:TBMSPROC
0022	5801	R 204 +2	DW OFFSET DGROUP:STK_TBMSPROC
----		205 +1	CODE ENDS
----		206 +1	DATA SEGMENT PUBLIC 'DATA'
002A	(7	207 +2	PCB_TBMSPROC DW 7 DUP (?)
	????		
	)		
----		208 +1	DATA ENDS
----		209 +1	STACKS SEGMENT PUBLIC 'STACKS' ;CREATE STACK
010C	(38	210 +2	LIM_TBMSPROC DW 26H DUP (OAAAAH)
	AAAA		
	)		
0158		211 +2	STK_TBMSPROC EQU 5
----		212 +1	STACKS ENDS
		213	;
		214 +1	
		215 +2	PUBLIC TPMBX
0024	06	216 +1	CODE SEGMENT PUBLIC 'CODE'
0025	3800	R 217 +1	DB 6
----		218 +2	DW OFFSET DGROUP:TPMBX
----		219 +1	CODE ENDS
0038	(8	220 +1	DATA SEGMENT PUBLIC 'DATA'
	????	221 +2	TPMBX DW 8 DUP (?)
	)		
----		222 +1	DATA ENDS
		223	;
		224 +1	
		225 +2	PUBLIC RPMBX
0027	06	226 +1	CODE SEGMENT PUBLIC 'CODE'
0028	4800	R 227 +1	DB 6
----		228 +2	DW OFFSET DGROUP:RPMBX
----		229 +1	CODE ENDS
0048	(8	230 +1	DATA SEGMENT PUBLIC 'DATA'
	????	231 +2	RPMBX DW 8 DUP (?)
	)		
----		232 +1	DATA ENDS
		233	;
		234 +1	
		235 +2	PUBLIC IPINMBX
002A	06	236 +1	CODE SEGMENT PUBLIC 'CODE'
002B	5800	R 237 +1	DB 6
----		238 +2	DW OFFSET DGROUP:IPINMBX
----		239 +1	CODE ENDS
0058	(8	240 +1	DATA SEGMENT PUBLIC 'DATA'
	????	241 +2	IPINMBX DW 8 DUP (?)
	)		
----		242 +1	DATA ENDS
		243	;
		244 +1	

```

LOC  OBJ          LINE  SOURCE
-----
                245 +2      PUBLIC FREEIRBMBX
002D  06          246 +1      CODE   SEGMENT PUBLIC 'CODE'
                247 +1      DB     6
002E  6800       R        248 +2      DW     OFFSET DGROUP:FREEIRBMBX
                249 +1      CODE   ENDS
                250 +1      DATA  SEGMENT PUBLIC 'DATA'
0068  (8        251 +2      FREEIRBMBX DW     8 DUP (?)
      )
                252 +1      DATA  ENDS
                253          ;
                254 +1
                255 +2      PUBLIC FREELIRBMBX
0030  06          256 +1      CODE   SEGMENT PUBLIC 'CODE'
                257 +1      DB     6
0031  7800       R        258 +2      DW     OFFSET DGROUP:FREELIRBMBX
                259 +1      CODE   ENDS
                260 +1      DATA  SEGMENT PUBLIC 'DATA'
0078  (3        261 +2      FREELIRBMBX DW     8 DUP (?)
      )
                262 +1      DATA  ENDS
                263          ;
                264 +1
                265 +2      PUBLIC BUFMIPMBX
0033  06          266 +1      CODE   SEGMENT PUBLIC 'CODE'
                267 +1      DB     6
0034  8800       R        268 +2      DW     OFFSET DGROUP:BUFMIPMBX
                269 +1      CODE   ENDS
                270 +1      DATA  SEGMENT PUBLIC 'DATA'
0088  (8        271 +2      BUFMIPMBX  DW     8 DUP (?)
      )
                272 +1      DATA  ENDS
                273          ;
                274 +1
                275 +1      CODE   SEGMENT PUBLIC 'CODE'
                276 +2      PUBLIC SCHEDLOCK
0036  04          277 +1      DB     4
0037  9800       R        278 +2      DW     OFFSET DGROUP:SCHEDLOCK
0039  0100       279 +2      DW     1
                280 +1      CODE   ENDS
                281 +1      DATA  SEGMENT PUBLIC 'DATA'
0098  (4        282 +2      SCHEDLOCK DW     4 DUP (?)
      )
                283 +1      DATA  ENDS
                284          ;
                285 +1
                286 +1      CODE   SEGMENT PUBLIC 'CODE'
003B  00          287 +1      DB     0
                288 +1      CODE   ENDS
                289 +1
                290          END

```

MAILBOX(FREELIRBMBX)

MAILBOX(BUFMIPMBX)

SEMAPHORE(SCHEDLOCK,1)

ENDLIST