```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.1 ASSEMBLY OF MODULE CF8825
OBJECT MODULE PLACED IN :F1:CF8825.OBJ
ASSEMBLER INVOKED BY:  ASM86.86 :F1:CF8825.A86 PRINT(:F1:CF8825.LST) DATE(1-FEB-84) ERRORPRINT


LOC  OBJ                  LINE     SOURCE

                             1 +1  $TITLE(iAPX 86, 88 MONITOR CONFIGURATION TABLES FOR THE iSBC 88/25 SBC)
                             2     NAME    CF8825
                             3     ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                             4     ;
                             5     ;   TITLE: iSBC 88/25 SBC CONFIGURATION SOURCE MODULE
                             6     ;
                             7     ;       ABSTRACT:
                             8     ;       This module contains information necessary to configure
                             9     ;       the iAPX 86, 88 Monitor to run on the iSBC 88/25 SBC.  It is
                            10     ;       divided into two tables: 1) Device Configuration and
                            11     ;       2) Bootstrap Configuration.
                            12     ;
                            13     ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            14     ;
                            15     ;               iAPX 86, 88 DEVICE CONFIGURATION TABLE
                            16     ;
                            17     ;   There are a total of 9 macros which must all be invoked to
                            18     ;   configure the iAPX 86, 88 Monitor.  They must be invoked in the
                            19     ;   following order: CPU, MAX_BUAD_COUNT, BAUD_RATE, BAUD_RATE_TIMER,
                            20     ;   EXTRA_TIMER, SERIAL_PORT, PARALLEL_PORT, INTERRUPT_CONTROLLER AND
                            21     ;   NPX.  Of these, the BAUD_RATE_TIMER, EXTRA_TIMER, SERIAL_PORT,
                            22     ;   PARALLEL_PORT AND INTERRUPT_CONTROLLER macros may have a type
                            23     ;   specification of 'NONE' if there is none on the iAPX 86, 88
                            24     ;   based board being used.  In this case, no other parameters
                            25     ;   need to be specified.  If the type is a valid type, all parameters
                            26     ;   in the macro invocation must be specified, there are no defaults.
                            27     ;
                            28     ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            29 +1  $INCLUDE(:F1:CF957B.MAC)
                       =1   30 +1  $genonly
                       =1   31 +1  $save list
                       =1   32 +2          ;
                       =1   33     PUBLIC  baud_rate, baud_rate_count, b9600_count, b1200_count
                       =1   34     PUBLIC  pit_control_port, baud_counter_port, baud_counter_mode
                       =1   35     PUBLIC  unused_ctr_x_mode, unused_ctr_y_mode
                       =1   36     PUBLIC  extra_pit_exists, eeprom_exists, xit_control_port, xit_ctr0_mode
                       =1   37     PUBLIC  xit_ctr1_mode, xit_ctr2_mode, eeprom_timer_port
                       =1   38     PUBLIC  serial_exists, sio_data_port, sio_status_port, sio_mode
                       =1   39     PUBLIC  sio_command
                       =1   40     PUBLIC  parallel_exists, ppi_mode, ppi_input_port
                       =1   41     PUBLIC  ppi_output_port, ppi_status_port, ppi_control_port
                       =1   42     PUBLIC  pic_porta, pic_portb, icw1, icw2, icw4, int_controller_exists
                       =1   43
                       =1   44             DGROUP  GROUP   DATA
                       =1   45             ASSUME  CS:CODE, DS:DGROUP
                       =1   46
----                   =1   47     DATA    SEGMENT PUBLIC  'DATA'
----                   =1   48     DATA    ENDS
                       =1   49
                       =1   50 +1  $restore
```

```
LOC  OBJ                  LINE     SOURCE

                            51
                            52 +2
                            53 +2
                            54 +2  $eject
```

```
LOC  OBJ                  LINE     SOURCE

                            55 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            56 +1  ;
                            57 +1  ;       Baud Rate Public variables
                            58 +1  ;
                            59 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
----                        60 +4  CODE    SEGMENT PUBLIC  'CODE'
                            61 +2
0000 0000                   62 +2  baud_rate       DW      00H
0002 0000                   63 +2  baud_rate_count DW      00H
0004 0800                   64 +2  b9600_count     DW      08H
0006 4000                   65 +2  b1200_count     DW      40H
                            66 +1
----                        67 +1  CODE    ENDS
                            68 +1
                            69 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            70 +1  ;
                            71 +1  ;       Baud Rate Timer Public variables
                            72 +1  ;
                            73 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
----                        74 +6   CODE   SEGMENT PUBLIC  'CODE'
                            75 +1
0008 D600                   76 +2  pit_control_port        DW      0D6H
000A D400                   77 +2  baud_counter_port       DW      0D4H
000C B6                     78 +2  baud_counter_mode       DB      0B6H
000D 30                     79 +2  unused_ctr_x_mode       DB      30H
000E 70                     80 +2  unused_ctr_y_mode       DB      70H
                            81 +1
----                        82 +1  CODE    ENDS
                            83 +1
                            84 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            85 +1  ;
                            86 +1  ;       Extra Timer Public variables
                            87 +1  ;
                            88 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                            89 +2  ;
                            90 +2  ;       There is no Extra Timer in this configuration.
                            91 +2  ;
----                        92 +2  CODE    SEGMENT PUBLIC  'CODE'
                            93 +2
000F 00                     94 +2  extra_pit_exists        DB      0H
0010 00                     95 +2  eeprom_exists           DB      0H
0011 0000                   96 +2  xit_control_port        DW      0H
0013 0000                   97 +2  eeprom_timer_port       DW      0H
0015 00                     98 +2  xit_ctr0_mode           DB      0H
0016 00                     99 +2  xit_ctr1_mode           DB      0H
0017 00                    100 +2  xit_ctr2_mode           DB      0H
                           101 +2
----                       102 +2  CODE    ENDS
                           103 +2
                           104 +2  $eject
```

```
LOC  OBJ                  LINE     SOURCE

                           105 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           106 +1  ;
                           107 +1  ;       Serial Port Public variables
                           108 +1  ;
                           109 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
----                       110 +3  CODE    SEGMENT PUBLIC  'CODE'
                           111 +3
0018 FF                    112 +3  serial_exists    DB      0FFH
0019 D800                  113 +4  sio_data_port    DW      0D8H
001B DA00                  114 +4  sio_status_port DW      0DAH
001D 4E                    115 +3  sio_mode         DB      04EH
001E 37                    116 +3  sio_command      DB      037H
                           117 +3
----                       118 +3  CODE    ENDS
                           119 +3
                           120 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           121 +1  ;
                           122 +1  ;       Parallel Port Public variables
                           123 +1  ;
                           124 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
----                       125 +3  CODE    SEGMENT PUBLIC  'CODE'
                           126 +3
001F FF                    127 +3  parallel_exists          DB      0FFH
0020 C6                    128 +3  ppi_mode                 DB      0C6H
0021 C800                  129 +4  ppi_input_port           DW      0C8H
0023 CA00                  130 +4  ppi_output_port          DW      0CAH
0025 CC00                  131 +4  ppi_status_port          DW      0CCH
0027 CE00                  132 +4  ppi_control_port         DW      0CEH
                           133 +3
----                       134 +3  CODE    ENDS
                           135 +3
                           136 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           137 +1  ;
                           138 +1  ;       Interrupt Controller Public variables
                           139 +1  ;
                           140 +1  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           141 +5
----                       142 +6   CODE   SEGMENT PUBLIC  'CODE'
                           143 +1
0029 C000                  144 +2  pic_porta        DW      0C0H
002B C200                  145 +2  pic_portb                DW      0C2H
002D 1700                  146 +2  icw1                     DW      17H
002F 2000                  147 +2  icw2                     DW      20H
0031 1D00                  148 +2  icw4                     DW      1DH
0033 FF                    149 +2  int_controller_exists    DB      0FFH
                           150 +1
----                       151 +1  CODE    ENDS
                           152 +1
                           153 +2
                           154 +3  $eject
```

```
LOC  OBJ                 LINE     SOURCE

                          155 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                          156 +2  ;
                          157 +2  ;       TITLE:  NPX_SUPPORT
                          158 +2  ;
                          159 +2  ;       ABSTRACT:
                          160 +2  ;       THE NPX IN NOT AVAILABLE IN THIS CONFIGURATION OF THE
                          161 +2  ;       iAPX 86, 88 MONITOR.  THE PROCEDURES CONTAINED IN THIS MODULE
                          162 +2  ;       SATISFY CALLS TO NPX SUPPORT MODULES BY OUTPUTING AN ERROR
                          163 +2  ;       MESSAGE THAT THE NPX IS NOT AVAILABLE.
                          164 +2  ;
                          165 +2  ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                          166 +2
                          167 +2          public  npx_register, npx_display, npx_flags, maximum_precision
                          168 +2          public  print_npx_value, set_npx_value, move_npx_value
                          169 +2          public  npx_exists
                          170 +2
                          171 +2          extrn   npx_error:near
                          172 +2
----                      173 +2  data    segment public  'DATA'
0000 ??                   174 +2  npx_flags       db      ?
0001 ????                 175 +2  maximum_precision       dw      ?
----                      176 +2  data    ends
                          177 +2
----                      178 +2  code    segment public  'CODE'
                          179 +2
0034 00                   180 +2  npx_exists      db      0H
                          181 +2
0035                      182 +2  npx_register    proc    near
0035 E90000          E    183 +2          jmp     npx_error
                          184 +2  npx_register    endp
                          185 +2
0038                      186 +2  npx_display     proc    near
0038 E90000          E    187 +2          jmp     npx_error
                          188 +2  npx_display     endp
                          189 +2
003B                      190 +2  print_npx_value proc    near
003B E90000          E    191 +2          jmp     npx_error
                          192 +2  print_npx_value endp
                          193 +2
003E                      194 +2  set_npx_value   proc    near
003E E90000          E    195 +2          jmp     npx_error
                          196 +2  set_npx_value   endp
                          197 +2
0041                      198 +2  move_npx_value  proc    near
0041 E90000          E    199 +2          jmp     npx_error
                          200 +2  move_npx_value  endp
                          201 +2
----                      202 +2  code    ends
                          203 +2
                          204
                          205 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                           206     ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           207     ;
                           208     ;                iAPX 86, 88 MONITOR - BOOTSTRAP LOADER
                           209     ;                        CONFIGURATION TABLE
                           210     ;
                           211     ;   The iAPX 86, 88 Monitor - Bootstrap Loader Configuration Table is
                           212     ;   is used to configure the iRMX 86 or iRMX 88 Bootstrap Loader
                           213     ;   into the iAPX 86, 88 monitor.  To use this table, determine which
                           214     ;   devices you want to be able to bootstrap off of, replace the
                           215     ;   semi-colon with a percent sign by the BOOTSTRAP macro, the
                           216     ;   DEVICE macros which describe the bootstrap devices (one or more)
                           217     ;   and the END_BOOTSTRAP macro in that order only.  The Bootstrap
                           218     ;   Loader will be configured in MANUAL mode for use in the monitor.
                           219     ;
                           220     ; *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                           221     ;BOOTSTRAP
                           222             ;DEVICE(f0,0,deviceinit204,deviceread204)
                           223             ;DEVICE(f1,1,deviceinit204,deviceread204)
                           224             ;DEVICE(f2,2,deviceinit204,deviceread204)
                           225             ;DEVICE(f3,3,deviceinit204,deviceread204)
                           226             ;DEVICE(d0,0,deviceinit206,deviceread206)
                           227             ;DEVICE(w0,0,deviceinit215,deviceread215)
                           228             ;DEVICE(wf0,8,deviceinit215,deviceread215)
                           229             ;DEVICE(wf1,9,deviceinit215,deviceread215)
                           230             ;DEVICE(wf2,10,deviceinit215,deviceread215)
                           231             ;DEVICE(wf3,11,deviceinit215,deviceread215)
                           232             ;DEVICE(b0,0,deviceinit254,deviceread254)
                           233     ;END_BOOTSTRAP
                           234
                           235     END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```