

```

*-----*
* Program Number: Display ROM at 8000H
* Written by      : John Monahan
* Date Created   : 11/9/2011
* Description    : Check Board is reading ROM correctly (Note No Stack/RAM is used)
*

```

```

*-----*
BELL      EQU      $07
BLANK     EQU      $20
CR        EQU      $0D
LF        EQU      $0A

                ORG      $0000

                DC.L     $0000FFFH ;SSP = 0FFFH
                DC.L     $00001000 ;PC = 1000H

                ORG      $1000

                MOVE.L   #$00FF0000,A0 ;Point to status Port 0H (CRT, Propeller Console IO Board)
                MOVE.L   #$00FF0001,A1 ;Point to hardware Port 01H (CRT, Propeller Console IO Board)
                MOVE.B   #6,D0          ;For Trap #15 D0 = #6 (always), to display a character
                                        ;(Character will always be in D1).

START:         MOVE.L   #$00001000,A2 ;Pointer to ROM area, Copy contents to RAM area
                MOVE.L   #$00008000,A3 ;RAM Area
                MOVE.L   #$000011FF,D4 ;Limit for ROM display

MOREMOVE:     MOVE.B   (A2)+,(A3)+ ;ROM copy routine
                SUBI.L   #1,D4
                TST.B   D4          ;Have we done all teh area
                BNE     MOREMOVE

                MOVE.L   #$00008000,A3 ;Pointer to RAM area now containing ROM data
                MOVE.L   #$000011FF,D4 ;Limit for RAM display

NEWLINE:     MOVE.B   #32,D3          ;32 bytes across per line

                MOVE.B   #CR,D1          ;SEND cr/lf
STAT1:;      MOVE.B   (A3),D5          ;Check CRT status is ready to recieve character
;           ANDI.B   #$04,D5
;           TST.B   D5
;           BEQ     STAT1
;           MOVE.B   D1,(A0)          ;Output ASCII (in D1) to hardware port 01H
;           TRAP    #15

                MOVE.B   #LF,D1
STAT2:;      MOVE.B   (A3),D5          ;Check CRT status is ready to recieve character
;           ANDI.B   #$04,D5
;           TST.B   D5
;           BEQ     STAT2
;           MOVE.B   D1,(A0)          ;Output ASCII (in D1) to hardware port 01H
;           TRAP    #15

                MOVE.W   A3,D1          ;Address upper high byte nibble
                LSR.W   #8,D1          ;Move address @A3 to D1
                LSR.W   #4,D1          ;Shift upper byte to lower 8 bits
                ANDI.B   #$0F,D1          ;SAVE LOWER NIBBLE
                ORI.B   #$30,D1          ;CONVERT TO ASCII
                CMPI.B   #$39,D1          ;SEE IF IT IS>9
                BLE.S   HEXOK1
                ADD     #7,D1          ;ADD TO MAKE 10=>A
HEXOK1:      NOP
STAT3:;      MOVE.B   (A3),D5          ;Check CRT status is ready to recieve character
;           ANDI.B   #$04,D5
;           TST.B   D5
;           BEQ     STAT3
;           MOVE.B   D1,(A0)          ;Output ASCII (in A1, high byte) to hardware port 01H
;           trap    #15

                MOVE.W   A3,D1          ;Address lower high byte nibble
                LSR.W   #8,D1          ;Move address @A3 to D1
                LSR.W   #8,D1          ;Shift upper byte to lower 8 bits

```

```

ANDI.B  #$0F,D1          ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK2
ADD     #7,D1           ;ADD TO MAKE 10=>A

HEXOK2:
NOP
STAT4:;  MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;        ANDI.B   #$04,D5
;        TST.B    D5
;        BEQ     STAT4
;        MOVE.B  D1,(A0)      ;Output ASCII (in A1, high byte) to hardware port 01H
;        trap    #15

;Address upper lower byte nibble
MOVE.W  A3,D1          ;Move address @A3 to D1
LSR.W   #4,D1         ;Shift upper byte to lower 8 bits
ANDI.B  #$0F,D1          ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK3
ADD     #7,D1           ;ADD TO MAKE 10=>A

HEXOK3:
NOP
STAT5:;  MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;        ANDI.B   #$04,D5
;        TST.B    D5
;        BEQ     STAT5
;        MOVE.B  D1,(A0)      ;Output ASCII (in A1, high byte) to hardware port 01H
;        trap    #15

;Address lower lower byte nibble
MOVE.W  A3,D1          ;Move address @A3 to D1
ANDI.B  #$0F,D1          ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK4
ADD     #7,D1           ;ADD TO MAKE 10=>A

HEXOK4:
NOP
STAT6:;  MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;        ANDI.B   #$04,D5
;        TST.B    D5
;        BEQ     STAT6
;        MOVE.B  D1,(A0)      ;Output ASCII (in A1, low byte) to hardware port 01H
;        trap    #15

MOVE.B  #BLANK,D1
STAT7:;  MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;        ANDI.B   #$04,D5
;        TST.B    D5
;        BEQ     STAT7
;        MOVE.B  D1,(A0)      ;Output ASCII (in A1, low byte) to hardware port 01H
;        TRAP    #15

START1:
MOVE.B  (A3),D1
LSR.W   #4,D1          ;Shift upper nibble to lower 4 bits
ANDI.B  #$0F,D1          ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK5
ADD     #7,D1           ;ADD TO MAKE 10=>A

HEXOK5:
NOP
STAT8:;  MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;        ANDI.B   #$04,D5
;        TST.B    D5
;        BEQ     STAT8
;        MOVE.B  D1,(A0)      ;Output ASCII (in D1) to hardware port 01H
;        trap    #15

MOVE.B  (A3),D1
ANDI.B  #$0F,D1          ;SAVE LOWER NIBBLE
ORI.B   #$30,D1         ;CONVERT TO ASCII
CMPI.B  #$39,D1         ;SEE IF IT IS>9
BLE.S   HEXOK6
ADD     #7,D1           ;ADD TO MAKE 10=>A

HEXOK6:
NOP

```

```
STAT9:;      MOVE.B   (A3),D5      ;Check CRT status is ready to recieve character
;           ANDI.B   #$04,D5
;           TST.B    D5
;           BEQ     STAT9
;           MOVE.B  D1,(A0)    ;Output ASCII (in D1) to hardware port 01H
;           trap    #15

           ADD     #1,A3      ;No matter what point to next byte

           SUBI.B  #1,D3
           TST.B   D3        ;Have we done 32X2 characters across
           BEQ     NEWLINE

           SUBI.B  #1,D4
           TST.B   D4        ;Have we done 32X2 characters across
           BNE    START1
DONE:      BRA     DONE      ;Loop continously

           END     $1000
```