MOTOROLA
Semiconductor Products Inc.

AN-809
Application Note

INTERFACING THE MC68000 TO THE MC6846 RIOT

Prepared by
David Ruhberg
Microprocessor Applications Engineer
Austin, Texas

The MC6846 ROM I/O Timer (RIOT) provides several versatile functions which the MC68000 may use with minimal effort. The RIOT features a 2K by 8 mask-programmable ROM, an 8-bit I/O port, and a 16-bit programmable timer/counter, in one forty-pin package. The MC68000 has the option of addressing the RIOT singly or in pairs, depending on the desired bus width. The 8-bit bus can be used if the upper and lower data strobes are used. Note that if a single RIOT is used, the MC68000 will not be able to obtain executable code from the ROM within the RIOT. This is due to the limitation introduced by the width of the data bus on the RIOT. Therefore, to effectively interface the ROM in the RIOT to the MC68000, a 16-bit data bus is used in this application. This configuration makes three 16-bit capabilities available to the MC68000. They are:

- 2K by 16 bits of mask-programmable ROM
- two parallel, 8-bit I/O ports, or one parallel, 16-bit I/O port
- two 16-bit timers that can be used together or independently

HARDWARE

The basic connections needed for the RIOT to function with the MC68000 are: the lower ten address lines (A1-A10), the sixteen data lines (D0-D15), and the R/$\overline{W}$, $\overline{RESET}$, E, and chip select signals. All of these may be obtained directly from the MC68000 with the exception of the chip select signals. As shown in Figure 1, the eight high-order data lines go to one RIOT and the eight low-order data lines go to the other RIOT. All other connections between the RIOTs are made in parallel. To obtain the chip select signals, some decoding circuitry must be provided. The RIOT may be run synchronously or asynchronously with the MC68000.

Synchronous Operation — To run the RIOT synchronously, some decoding circuitry must be used to provide a low input to the $\overline{VPA}$ pin of the MC68000 when the RIOT is selected. This synchronizes the MC68000 with E and generates the $\overline{VMA}$ signal.

To use the synchronous output of the MC68000, the decoding circuitry must also generate a $\overline{VPA}$ signal, in addition to the chip selects. This signal informs the MC68000 that it is addressing a M6800 peripheral and synchronizes the processor with the E clock. The $\overline{VPA}$ signal also causes $\overline{VMA}$ to be generated which can be used for other M6800 peripherals.

Asynchronous Operation — Operating the RIOTs asynchronously with the MC68000 allows the processor to begin executing the next instruction without waiting to synchronize with the E clock. To operate asynchronously, the decoding circuitry must generate a $\overline{DTACK}$ signal in addition to the chip selects.

SAMPLE CIRCUIT

The sample circuit interfaces parallel RIOTs through the MC68000 Design Module (MEX68KDM) data bus, as shown in Figure 2. Since only sixteen address lines (A1-A16) are brought out on this bus, addressing from the MC68000 is incomplete. Special attention should be given in addressing these devices from the MC68000 since the A0 address line on the RIOT corresponds to the A1 address line on the MC68000.

The RIOT used in this sample circuit (MC6846P3 TVBug) has the following characteristics. Having CS0 high and CS1 low selects the ROM, while having CS0 low and CS1 high selects the I/O Timer. Also, address lines A6 and A10 must be high and lines A3, A4, and A5 must be low for the I/O Timer to be selected. The three least-significant address bits are used to address the various I/O Timer control registers. A memory map is given in Figure 3. Besides power and ground, the E, CS0, CS1, R/$\overline{W}$, $\overline{RESET}$, and the ten address lines are connected in parallel to the RIOTs.

As mentioned earlier, the address lines are obtained from the limited address bus of the Design Module. Buffers are required to reduce noise on the bus lines. Bidirectional, three-stateable buffers are used on the data lines so that data may be transmitted to and received from the Module. The receive enable ($\overline{RE}$) signal will go low when R/$\overline{W}$ is low and the I/O

Timer chip select (CS1) is high. The driver enable (DE) signal will go high when R/$\overline{W}$ is high and either chip select is high.

The decoding circuitry shown in Figure 4 generates the chip select signals for synchronous operation. A high chip select signal (CS0) is generated for the ROM at addresses $10000 through $10FFF. The high chip select signal (CS1) is generated for the I/O Timer registers at addresses $11880 through $1188F. The two chip select signals are NORed together to generate the $\overline{VPA}$ signal. Valid peripheral address should be generated from an open collector gate or passed through a three-stateable buffer to permit a wire-ORed signal.

For asynchronous operation, the U10 NOR gate used to generate $\overline{VPA}$ is replaced with the TTL circuit shown in Figure 5.

The seven-segment displays are used to show the contents on the parallel I/O data registers in the RIOTs.

Address lines A4, A5, and A7 are decoded to allow software manipulation of the timer contained in each RIOT.

SOFTWARE

The software needed for the MC68000 to use the RIOT is straightforward. One point to keep in mind is that the MC68000 addresses every eight bits, even though it executes sixteen-bit instructions. This means that the least-significant byte of an instruction is always located at an odd address, and likewise the most-significant byte is always located at an even address. Since the hardware writes to all sixteen bits at once, the addresses for the control registers are located two addresses apart (i.e., PCR: 11882, DDR: 11884, etc.). In using the ROM, no preliminary software is necessary. However, to use the I/O lines, the peripheral control register must be initialized and the data direction register must be configured before data may be transmitted to or received from the peripheral data register.

The software for the sample circuit is given in Figure 6. The I/O lines are connected to four, seven-segment displays through MC14511 BCD-to-decimal decoders. The software initializes the I/O lines (as outputs) and then outputs the first ten bytes of each ROM. Since the decoders cannot decode hexadecimal numbers greater than nine, the software subtracts eight if the number is greater than nine. Then the code is output to the display for operator inspection. This software is included to give you an idea of the simplicity involved in interfacing to these M6800 peripherals.

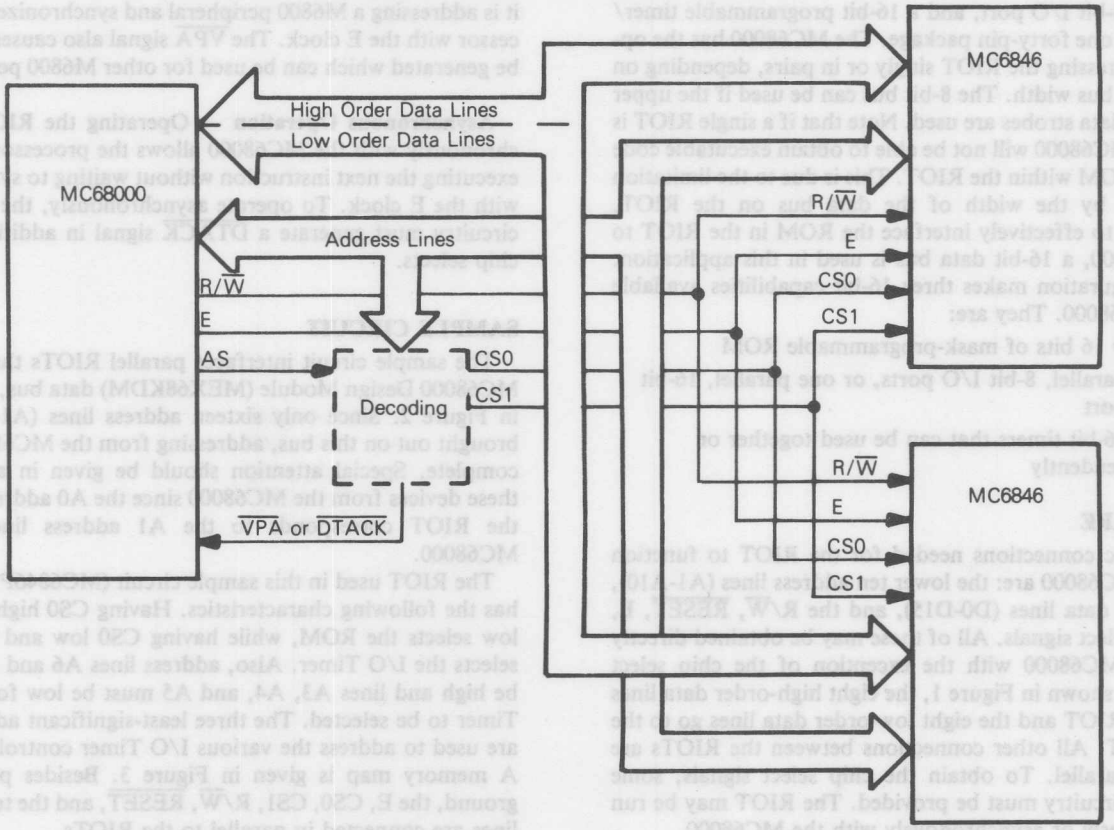

Figure 1. MC6846 to MC68000 Interface — Block Diagram

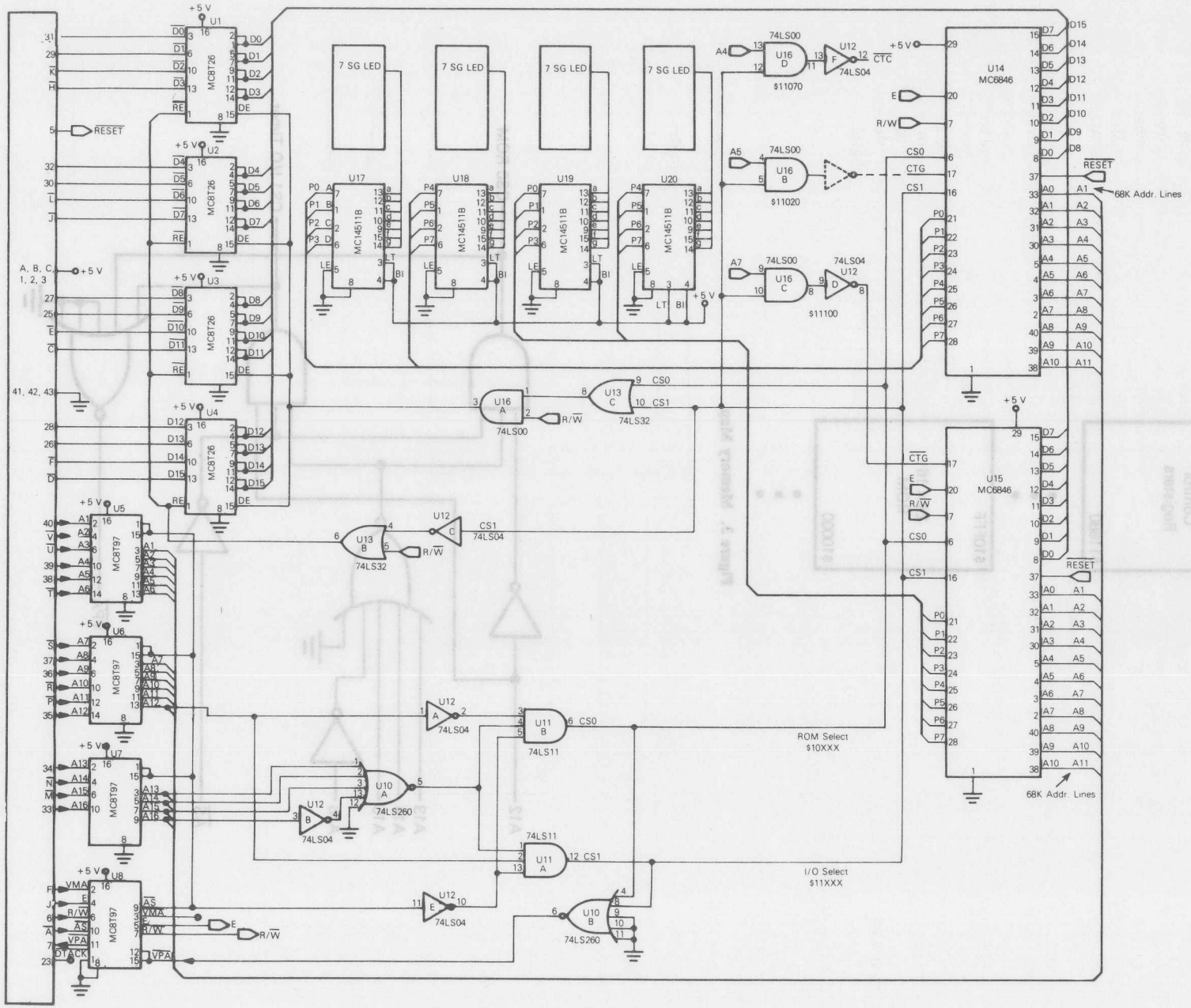

Figure 2. MC6846 to MC68000 Interface — Schematic

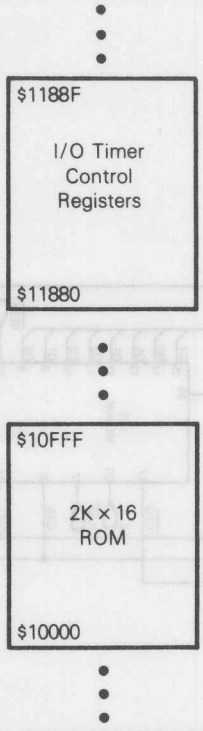

Figure 3. Memory Map

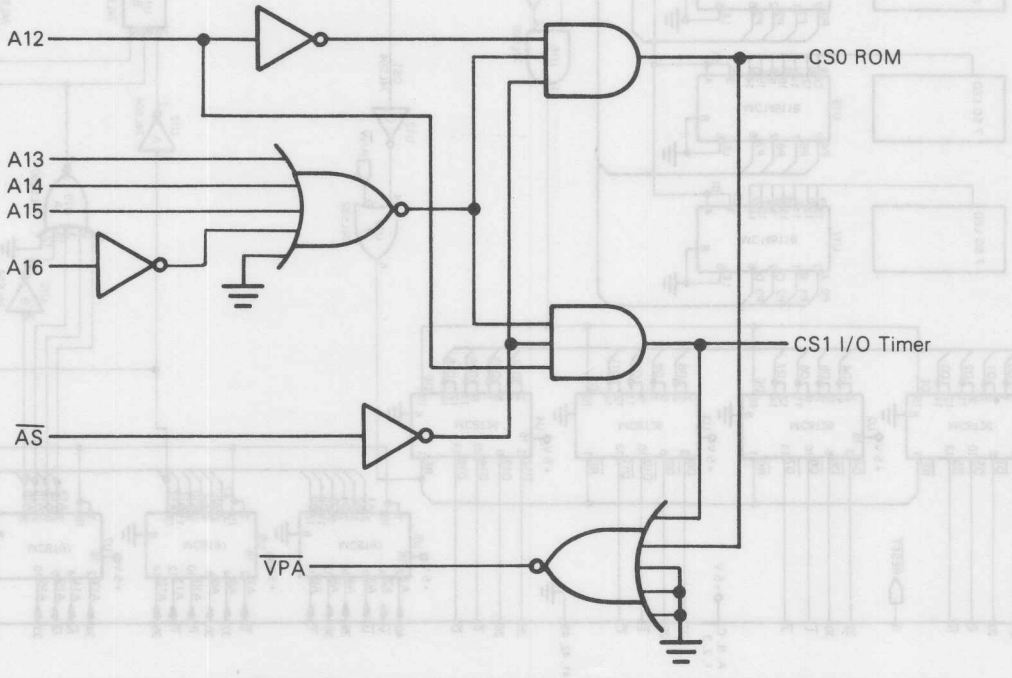

Figure 4. Decoding Circuitry (Synchronous Interface)

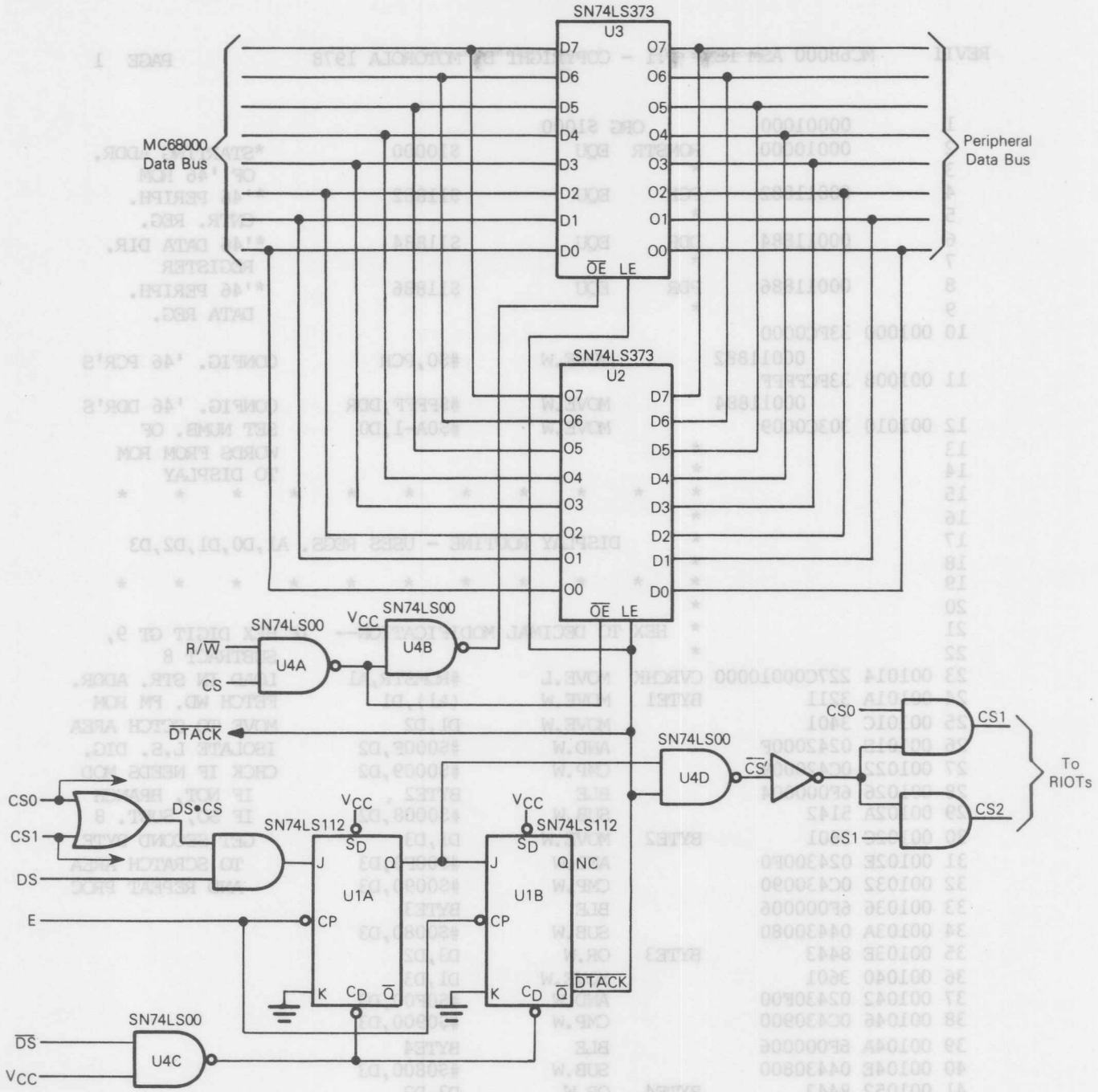

Figure 5. Asynchronous Interface Circuitry

```
REVII     MC68000 ASM REV= 1.1 - COPYRIGHT BY MOTOROLA 1978          PAGE  1


  1        00001000          ORG $1000
  2        00010000     ROMSTR  EQU          $10000          *STARTING ADDR.
  3                     *                                     OF '46 ROM
  4        00011882     PCR     EQU          $11882          *'46 PERIPH.
  5                     *                                     CNTR. REG.
  6        00011884     DDR     EQU          $11884          *'46 DATA DIR.
  7                     *                                     REGISTER
  8        00011886     PDR     EQU          $11886          *'46 PERIPH.
  9                     *                                     DATA REG.
 10 001000 33FC0000
               00011882          MOVE.W       #$0,PCR         CONFIG. '46 PCR'S
 11 001008 33FCFFFF
               00011884          MOVE.W       #$FFFF,DDR      CONFIG. '46 DDR'S
 12 001010 303C0009              MOVE.W       #$0A-1,D0       SET NUMB. OF
 13                    *                                     WORDS FROM ROM
 14                    *                                     TO DISPLAY
 15                    *    *    *    *    *    *    *    *    *    *    *
 16                    *
 17                    *       DISPLAY ROUTINE - USES REGS. A1,D0,D1,D2,D3
 18                    *
 19                    *    *    *    *    *    *    *    *    *    *    *
 20                    *
 21                    *  HEX TO DECIMAL MODIFICATION--  IF HEX DIGIT GT 9,
 22                    *                                   SUBTRACT 8
 23 001014 227C00010000 CVRCHK  MOVE.L       #ROMSTR,A1      LOAD IN STR. ADDR.
 24 00101A 3211         BYTE1   MOVE.W       (A1),D1         FETCH WD. FM ROM
 25 00101C 3401                 MOVE.W       D1,D2           MOVE TO SCTCH AREA
 26 00101E 0242000F             AND.W        #$000F,D2       ISOLATE L.S. DIG.
 27 001022 0C420009             CMP.W        #$0009,D2       CHCK IF NEEDS MOD
 28 001026 6F000004             BLE          BYTE2             IF NOT, BRANCH
 29 00102A 5142                 SUB.W        #$0008,D2         IF SO, SUBT. 8
 30 00102C 3601         BYTE2   MOVE.W       D1,D3           GET SECOND BYTE
 31 00102E 024300F0             AND.W        #$00F0,D3         TO SCRATCH AREA
 32 001032 0C430090             CMP.W        #$0090,D3         AND REPEAT PROC
 33 001036 6F000006             BLE          BYTE3
 34 00103A 04430080             SUB.W        #$0080,D3
 35 00103E 8443         BYTE3   OR.W         D3,D2
 36 001040 3601                 MOVE.W       D1,D3
 37 001042 02430F00             AND.W        #$0F00,D3
 38 001046 0C430900             CMP.W        #$0900,D3
 39 00104A 6F000006             BLE          BYTE4
 40 00104E 04430800             SUB.W        #$0800,D3
 41 001052 8443         BYTE4   OR.W         D3,D2
 42 001054 2601                 MOVE.L       D1,D3         USE LONG WD. FOR MSB.
 43 001056 02830000F000         AND.L        #$F000,D3
 44 00105C 0C8300009000         CMP.L        #$9000,D3
 45 001062 6F000006             BLE          DONE
 46 001066 04438000             SUB.W        #$8000,D3
 47 00106A 8443         DONE    OR.W         D3,D2           DISPYD DATA  IN D2
 48 00106C 33C200011886         MOVE.W       D2,PDR          WRITE DATA TO PIA
 49 001072 2E3C0003D090         MOVE.L      #250000,D7       DLY FOR 5 SEC
 50 001078 5387         DLY     SUB.L        #1,D7
 51 00107A 6AFC                 BPL          DLY
 52 00107C 51C8FF9C             DBRA         D0,BYTE1        GO AGN IF <> 10
 53                    *
```

Figure 6. Program Listing

```
REVII     MC68000 ASM REV= 1.1 - COPYRIGHT BY MOTOROLA 1978          PAGE  2

 54                     *         BACK TO MACSBUG
 55                     *
 56 001080 4E4F                TRAP         15
 57 001082 0000                DC.W          0
 58                     *
 59                     * THE PIA ADRESSES ARE   $11882   PCR
 60                     *                        $11884   DDR
 61                     *                        $11886   TO WRITE TO THE DISPL
 62                          END

****** TOTAL ERRORS   0--   0


SYMBOL TABLE

BYTE1       00101A BYTE2       00102C BYTE3       00103E BYTE4       001052
CVRCHK      001014 DDR         011884 DLY         001078 DONE        00106A
PCR         011882 PDR         011886 ROMSTR      010000
```

Figure 6. Program Listing (Concluded)

MOTOROLA *Semiconductor Products Inc.*
Printed in Switzerland