BYTESAVER

INSTRUCTION MANUAL

• Bytesaver Assembly  • Bytesaver Parts List
• Bytemover Software  • 2708-2704 PROM Data
• Bytemover Assembly Listing

© Copyright 1977. All rights reserved.

Cromemco
inco rporated
Specialists in computers and peripherals
2400 CHARLESTON RD., MOUNTAIN VIEW, CA 94043
Introduction

The Cromemco Bytesaver is a read/write, non-volatile memory board, plug compatible with the Standard-100 (S-100) microcomputer bus. The Bytesaver has the capacity for eight 2708 U.V. erasable PROMs for a full 8K bytes of memory.

The Bytesaver contains an integral PROM programmer along with a DC-to-DC supply for generating the programming voltage. Programming is accomplished by a series of memory write operations to the PROM being programmed.

Cromemco provides the necessary programming software. Our Bytemover software, described later in this manual, allows convenient PROM programming using your computer's front panel sense switches to control the operation (e.g. to select one of the eight PROMs to be programmed). Programming can also be carried out using the Z-80 Monitor supplied with our Z-80 CPU card and our Z-80 microcomputer system.

Assembly Instructions

The Cromemco Bytesaver™ kit can be assembled in about one evening. All components are mounted on the component side of the p.c. board (the side with the printed legend) and soldered on the opposite side. Be sure to use high-quality, rosin core solder for the assembly and a fine-tipped, low-wattage soldering iron.

1. Solder the 10 14-pin IC sockets, the 6 16-pin IC sockets and the 8 24-pin sockets in position.

2. Solder the following 1/4 Watt resistors in position:

<table>
<thead>
<tr>
<th>R1</th>
<th>47K</th>
<th>yellow-violet-orange</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>10K</td>
<td>brown-black-orange</td>
</tr>
<tr>
<td>R3</td>
<td>180</td>
<td>brown-gray-brown</td>
</tr>
<tr>
<td>R4</td>
<td>1K</td>
<td>brown-black-red</td>
</tr>
<tr>
<td>R5</td>
<td>1.9K</td>
<td>white-brown-red</td>
</tr>
<tr>
<td>R6</td>
<td>1.5K</td>
<td>brown-green-red</td>
</tr>
<tr>
<td>R7</td>
<td>1.2K</td>
<td>brown-red-red</td>
</tr>
<tr>
<td>R8</td>
<td>47</td>
<td>yellow-violet-black</td>
</tr>
<tr>
<td>R9</td>
<td>1K</td>
<td>brown-black-red</td>
</tr>
<tr>
<td>R10</td>
<td>10</td>
<td>brown-black-black</td>
</tr>
<tr>
<td>R11</td>
<td>5.6K</td>
<td>green-blue-red</td>
</tr>
<tr>
<td>R12</td>
<td>5.6K</td>
<td>green-blue-red</td>
</tr>
<tr>
<td>R13</td>
<td>10K</td>
<td>brown-black-orange</td>
</tr>
<tr>
<td>R14</td>
<td>5.6K</td>
<td>green-blue-red</td>
</tr>
<tr>
<td>R15</td>
<td>180</td>
<td>brown-gray-brown</td>
</tr>
<tr>
<td>R16-R31</td>
<td>18K</td>
<td>brown-gray-orange</td>
</tr>
<tr>
<td>R32-R39</td>
<td>4.7K</td>
<td>yellow-violet-red</td>
</tr>
</tbody>
</table>

3. Next, install the 1N914 diodes. We recommend that no diode be installed in the diode position just below transistor Q0. Since we recommend that the PROM containing the Bytemover software be inserted in PROM position zero, installing this diode may allow accidental programming of this PROM.

When installing the diodes, be careful to orient them properly, by noting the position of the cathode (banded) end. Due to the close spacing of the holes in the p.c. board, all diodes should be mounted on end.

4. Install the 23 capacitors as shown on the p.c. board. Be careful to orient the electrolytic capacitors with the positive (+) end as shown on the board.

5. Solder the transistors in place taking care to orient them properly. Note that Q8 and Q9 are 2N3906 transistors and Q10 is a type MPS6660. All other transistors are type 2N3904.

6. Install the p.c. board switch (SW1) in the upper left corner of the p.c. board.

7. Install the Cromemco high-speed pulse transformer (Model XT8K) in position T1. Note that the leads are asymmetrically arranged so there is only one possible orientation.

8. Install IC14, the positive 12V regulator IC. Use a 6-32 by 1/4" screw and nut.

9. Initially install the heatsink in the upper right corner of the p.c. board by just starting the nuts on the 6-32 by 3/8" screws. Install IC12 and IC13 but be sure to place the insulating washer between IC13 and the heatsink. The nylon screw must be used to secure IC13. It is important that the screw be inserted from the p.c. board side so the screw head is against the foil side. Be aware that the insulating washer may have to be trimmed with a pair of scissors to clear the protrusions in the heatsink. Tighten the nuts on the screws in the heatsink assembly only after all the screws have been inserted. Take care that the leads of the voltage regulators do not come in contact with the sides of the openings in the heatsink. Although voltage regulators IC12, IC13 and IC14 may look similar, they are not interchangeable.

10. Install three jumper wires to select where the Bytesaver is to reside in memory. Each of the three high-order address lines (A15, A14 and A13) may be tied to either the corresponding "H" or "L" terminal. For example, in order for the Bytesaver to reside in the top 8K of memory, the three jumpers should be installed as shown below:

```
A15<--[O] HI
|[O] LO
|HI
|LO
|A14
```

This adjustment causes the Bytesaver to reside in the top 8K of the memory map.

11. Install the ICs in their sockets, being careful to orient pin 1 of each IC as shown by the small white dot on the p.c. board at each IC position. Install a PROM containing the Bytemover software in PROM position zero.

Your Bytesaver is now fully assembled. Detailed operating instructions are given in the Bytemover software section of this manual.

Notes

Interrupts: If you plan to have your computer respond to interrupts while executing a program stored in the Bytesaver memory, a small modification is required to the Bytesaver circuit. This modification is shown in Note 1 on the Bytesaver schematic.

Wait State: Should you wish to use low speed 2704s or 2708s (access times greater than 450 ns) in your Bytesaver, be aware there is a provision for a wait state. Simply insert the jumper wire between IC10 and IC11. No jumper need be inserted when using full speed PROMs. Jumper is also required for 4MHz operation.
Programming PROMs

PROM programming with the Bytesaver is carried out by a sequence of memory write operations to the Bytesaver card. A sequence of approximately 100 memory write operations is required to write the data into each location of every PROM that you wish to program. This sequence of memory write operations is carried out automatically when using the special instructions of Cromemco software. The Cromemco Z-80 Monitor, the Resident Operating System, and the 3K Control Basic Interpreter all have provision for PROM programming with the Bytesaver card. If your computer is equipped with front panel sense switches, our Bytemover software can be used for PROM programming. Operation of the Bytemover software is described in the next section.

Bytesaver Parts List

<table>
<thead>
<tr>
<th>Capacitors</th>
<th>Integrated Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1-C8</td>
<td>IC1</td>
</tr>
<tr>
<td>C9-C15</td>
<td>IC2</td>
</tr>
<tr>
<td>C16</td>
<td>IC3</td>
</tr>
<tr>
<td>C17</td>
<td>IC4</td>
</tr>
<tr>
<td>C18</td>
<td>IC5</td>
</tr>
<tr>
<td>C19</td>
<td>IC6</td>
</tr>
<tr>
<td>C20</td>
<td>IC7</td>
</tr>
<tr>
<td>C21-C23</td>
<td>IC8</td>
</tr>
<tr>
<td></td>
<td>IC9</td>
</tr>
<tr>
<td></td>
<td>IC10</td>
</tr>
<tr>
<td></td>
<td>IC11</td>
</tr>
<tr>
<td></td>
<td>IC12*</td>
</tr>
<tr>
<td></td>
<td>IC13*</td>
</tr>
<tr>
<td></td>
<td>IC14*</td>
</tr>
<tr>
<td></td>
<td>IC15</td>
</tr>
<tr>
<td></td>
<td>IC16</td>
</tr>
<tr>
<td></td>
<td>IC17</td>
</tr>
<tr>
<td></td>
<td>IC18</td>
</tr>
<tr>
<td></td>
<td>IC19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Diodes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D1-D19</td>
<td>.1N914 or 1N4531</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transistors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Q0-Q7</td>
<td>2N3904</td>
</tr>
<tr>
<td>Q8, Q9</td>
<td>2N3906</td>
</tr>
<tr>
<td>Q10, Q11, Q12</td>
<td>MF66560</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resistors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>47K</td>
</tr>
<tr>
<td>R2</td>
<td>10K</td>
</tr>
<tr>
<td>R3</td>
<td>180</td>
</tr>
<tr>
<td>R4</td>
<td>1K</td>
</tr>
<tr>
<td>R5</td>
<td>9.1K</td>
</tr>
<tr>
<td>R6</td>
<td>1.5K</td>
</tr>
<tr>
<td>R7</td>
<td>1.2K</td>
</tr>
<tr>
<td>R8</td>
<td>47</td>
</tr>
<tr>
<td>R9</td>
<td>1K</td>
</tr>
<tr>
<td>R10</td>
<td>10</td>
</tr>
<tr>
<td>R11</td>
<td>5.6K</td>
</tr>
<tr>
<td>R12</td>
<td>5.6K</td>
</tr>
<tr>
<td>R13</td>
<td>10K</td>
</tr>
<tr>
<td>R14</td>
<td>5.6K</td>
</tr>
<tr>
<td>R15</td>
<td>180</td>
</tr>
<tr>
<td>R16-R31</td>
<td>18K</td>
</tr>
<tr>
<td>R32-39</td>
<td>4.7K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integrated Circuits</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>.74123</td>
</tr>
<tr>
<td>IC2</td>
<td>.7474</td>
</tr>
<tr>
<td>IC3</td>
<td>.7402</td>
</tr>
<tr>
<td>IC4</td>
<td>.7406</td>
</tr>
<tr>
<td>IC5</td>
<td>.7406</td>
</tr>
<tr>
<td>IC6</td>
<td>.7402</td>
</tr>
<tr>
<td>IC7</td>
<td>.7406</td>
</tr>
<tr>
<td>IC8</td>
<td>.7442</td>
</tr>
<tr>
<td>IC9</td>
<td>.74L04</td>
</tr>
<tr>
<td>IC10</td>
<td>7410</td>
</tr>
<tr>
<td>IC11</td>
<td>74LS04</td>
</tr>
<tr>
<td>IC12*</td>
<td>340T-5.0 or 7805</td>
</tr>
<tr>
<td>IC13*</td>
<td>320T-5.0 or 7905</td>
</tr>
<tr>
<td>IC14*</td>
<td>340T-12 or 7812</td>
</tr>
<tr>
<td>IC15</td>
<td>7432 or 74LS32</td>
</tr>
<tr>
<td>IC16</td>
<td>74367</td>
</tr>
<tr>
<td>IC17</td>
<td>74367</td>
</tr>
<tr>
<td>IC18</td>
<td>74367</td>
</tr>
<tr>
<td>IC19</td>
<td>74367</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Miscellaneous</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SW1</td>
<td>p.c. board switch</td>
</tr>
<tr>
<td>T1</td>
<td>XT8K pulse transformer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sockets</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>14-pin</td>
</tr>
<tr>
<td>6</td>
<td>16-pin</td>
</tr>
<tr>
<td>8</td>
<td>24-pin</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hardware</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>6-32 by 3/8&quot; screws</td>
</tr>
<tr>
<td>1</td>
<td>6-32 by 1/4&quot; screw</td>
</tr>
<tr>
<td>1**</td>
<td>6-32 by 3/8&quot; nylon screw</td>
</tr>
<tr>
<td>5</td>
<td>6-32 nuts</td>
</tr>
<tr>
<td>1</td>
<td>Heatsink</td>
</tr>
<tr>
<td>1</td>
<td>Insulating washer</td>
</tr>
</tbody>
</table>

Notes

*The three voltage regulator ICs (IC12, IC13 & IC14) may look physically similar, however they are not interchangeable. Each must be in the proper IC location.

**The nylon screw is used to secure IC13. It is important that the screw be inserted from the p.c. board side of the assembly so the head of the screw is against the foil side of the board.
BYTEMOVER SOFTWARE

Introduction

Cromemco Bytemover software is designed to be used with the Cromemco 8K Bytesaver described. When you purchase a Bytesaver with one 2708 PROM, the Bytemover software is preprogrammed in that PROM.

The PROM containing the Bytemover software is normally inserted into PROM location zero on the Bytesaver board.

The Bytemover software can be used to program a PROM in any of the PROM locations on the Bytesaver board. The Bytemover software can also be used to transfer programs from PROM to RAM.

The operation of the Bytemover software is controlled by setting front panel sense switches on any S-100 bus-compatible computer. However, to use the Bytemover software there must be at least one RAM board in the computer beginning at location zero in the memory map. Furthermore, this RAM board must be unprotected for proper execution of the Bytemover software.

Programming Partially Filled PROMs

Software can be loaded into a 2704 or 2708 in as small increments as you desire provided it is added to previously unused areas in that PROM.

This is done by first using Bytemover to move the current contents of the PROM down to RAM, adding the new software to an area of RAM which corresponds to the unused portion of the PROM and finally using Bytemover again to re-program the PROM with the new software.

Although the entire PROM must always be programmed, it never hurts to re-write the same information over again. If 0's are programmed at any time,

In general, it is OK to write a "1" over a "1", a "0" over a "0", or a "0" over a "1". But in order to write a "1" over a "0", the PROM must first be completely erased.

If the PROM to which you want to add software is PROM zero on the Bytesaver board, turn off the A.C. power to the computer and install a 1N914 diode just below Q0 (see step 3 of the Bytesaver assembly instructions).

Turn the power back on and move Bytemover down to RAM zero by following Example 1. Add the new software to an area of RAM which corresponds to an unused portion of PROM zero.

Re-program PROM zero by following Example 4 of this manual. Note that you need not erase the PROM to do this. Turn the computer power off and remove the 1N914 diode below Q0.

PROM Programming Time

The Bytesaver software supplied here is designed to program a PROM in approximately 30 seconds. We have found that this is generally a sufficiently long period of programming time. However—to be completely within the manufacturer's specifications—the PROM should be programmed for 2 to 3 minutes.

If you wish to program your PROMs for longer than 30 seconds, the Bytemover software may be easily modified. Simply change the contents of location 77H from 40H to 00H. Now you must manually time the programming operation and depress the stop switch at the end of the operation.

Step By Step Instructions

1. Before using the Bytesaver, you must install three jumper wires to set the location of Bytesaver in memory. This adjustment is shown in Figure 1. The assembled Bytesaver comes with A13, A14 and A15 each tied to the corresponding HI pad to position the board at the very top of memory. In the following instructions it is assumed this is the jumper connection used.

2. Turn off all power to the computer and plug in the Bytesaver board.
3. Be sure the program power on the Bytesaver is turned off (program power switch in the down position).
4. Turn on the computer. Raise the reset switch, the stop switch and then raise the reset switch once again to initialize the computer.
5. Raise address switches A15, A14 and A13. All other address switches should be in the down position.
6. Raise the examine switch. You are now examining the contents of the first byte of PROM in PROM location zero of the Bytesaver memory board (memory location 340 000). If the PROM supplied with your Bytesaver is in this PROM location, the data lights will read “061,” the first byte of the Bytemover program.

Example 1: Transfer the Bytemover program from PROM to RAM beginning at location zero in RAM.

1. Raise the reset switch.
2. Depress the unprotect switch (on the Altair front panel).
4. Now set the sense switches for the task to be done, referring to Figure 2.

<table>
<thead>
<tr>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
<th>A10</th>
<th>A9</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>Down</td>
<td>Down</td>
<td>Down</td>
<td>Down</td>
<td>Down</td>
<td>Down</td>
<td>Down</td>
</tr>
</tbody>
</table>

This table shows the sense switches required for the above transfer.

The Bytemover program is now transferred to location zero in RAM.
5. Push the run switch. In less than one second, the contents of PROM will be transferred to RAM. The contents of PROM are unaffected by this operation.

6. Raise the stop switch.

7. Raise the reset switch. Note that the data lights read “061”.

Example 2: Program a 2708 PROM inserted in PROM location one. This PROM is to be programmed with the contents of the first 1K bytes of RAM beginning at location zero in memory. The Bytesaver software is still in the PROM installed at PROM location zero on the Bytesaver board.

1. Raise the reset switch.

2. Depress the unprotect switch (on the Altair front panel).


4. Raise the protect switch on the Bytesaver board (i.e. program power switch to the on position). The protect light on the front panel should go off when this switch is raised.

5. Now set the sense switches for the task to be done:

<table>
<thead>
<tr>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>Up</td>
<td>MSB</td>
<td>LSB</td>
<td>MSB</td>
</tr>
<tr>
<td>To program a PROM</td>
<td>For a 1K transfer</td>
<td>PROM address location in increments of 1K from the PROM in which Bytesaver is stored</td>
<td>Selection of RAM address in 1K increments</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 2: Function of the sense switches in Bytesaver.

6. Push the run switch. Note that panel light A9 is blinking at a rate of about twice per second. When this light stops blinking, the PROM programming is complete.

7. Raise the stop switch.

Example 3: Altair 8K BASIC can be stored in seven 2708 PROMs. Given that these seven PROMs are in PROM locations one through seven of the Bytesaver board, 8K BASIC can easily be transferred into RAM using the following procedure:

1. Raise the reset switch.

2. Depress the unprotect switch (on the Altair front panel).


4. Now set the sense switches for the task to be done:

<table>
<thead>
<tr>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
<th>A10</th>
<th>A9</th>
<th>A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down</td>
<td>Up</td>
<td>Down</td>
<td>Up</td>
<td>Down</td>
<td>Up</td>
<td>Down</td>
<td>Down</td>
</tr>
<tr>
<td>to transfer from PROM to RAM</td>
<td>for a 7K transfer</td>
<td>To begin transfer from the PROM 1K higher in memory than the Bytesaver program</td>
<td>All down for storage to begin at location zero in RAM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Push the run switch. In less than one second BASIC will be loaded into RAM (it sure beats paper tape!). Now raise the stop switch.

Example 4: If you do not have Bytesaver in PROM, you can program a PROM with Bytesaver that is stored in RAM. The Bytesaver software (see listing) must first be loaded into RAM beginning at location zero in memory. The Bytesaver software can then be burned into a PROM using the following procedure:

1. Raise the reset switch.

2. Depress the unprotect switch (on the Altair front panel).

3. Insert an erased PROM into PROM location zero.

4. Examine location 000 240 in memory.

5. Raise the program power switch on the Bytesaver board.

6. Set the sense switches with A15, A14 and A13 up.

7. Push the run switch. When light A9 stops blinking, the programming is complete. The INTE light will be on.

8. Turn off PROM program power by depressing the switch on the Bytesaver.
FIGURE 3: Bytesaver physical layout.

Erasing PROMS: The 2704 and 2708 PROMs are erased by shining intense U.V. light through their quartz windows. One such U.V. source is available for $125 from Prometrics, 5345 North Kedzie Av., Chicago, IL 60625.

Bytemover 3.1 Octal Listing

```
061 000 000 301 321 056 311 363 345 345 000 000 000 061 004 000
315 000 000 061 002 000 341 061 004 000 325 305 371 016 000 131
151 333 377 127 346 007 007 007 107 172 346 070 017 000 147 071
056 000 172 353 346 200 017 017 306 055 041 000 000 157 071 351
371 041 013 000 071 353 371 353 021 000 000 073 361 002 003 023
172 346 004 007 007 007 205 157 351 000 000 076 126 205 157 351
000 151 174 140 371 147 056 153 001 000 000 073 361 022 023 003
170 376 374 077 037 037 346 100 056 175 205 157 351 056 153 170
346 004 007 007 007 205 157 351 000 000 000 174 041 000 374 071
371 041 000 374 031 353 147 056 153 170 346 370 306 010 107 351
333 377 107 346 340 036 000 113 127 170 346 037 107 147 056 140
351 306 032 157 333 377 346 100 017 017 205 157 351 174 041 000
374 071 371 056 315 147 351 000 000 000 000 373 351 174 041 000
374 031 353 056 361 147 001 000 000 351 000 326 220 157 172 306
004 127 376 070 077 076 000 037 205 157 351 000 000 373 351 351
351 073 361 353 276 353 027 346 001 057 074 205 157 073 073 361
057 353 206 353 306 007 077 027 346 001 057 074 205 157 003 023
170 346 004 057 074 205 157 351 000 000 000 000 000 000 000 000
```
0000 | 0000 * BYTEMOVER (T.M.) SOFTWARE FOR
0000 | 0001 * CROMEMCO 8K BYTESAVER (T.M.)
0000 | 0002 * VERSION 3.1
0000 | 0003 * SELF-RELOCATING SOFTWARE LOCATABLE AT ANY
0000 | 0004 * 1024 BYTE (1K) BOUNDARY IN MEMORY
0000 | 0009 * ROUTINE TO FIND ONESELF IN MEMORY
0000 | 0010 SP EQU 6
0000 | 0019 * DEFINE FIRST 4 BYTES IN MEMORY AS STACK
0000 31 00 00 | 0020 LXI SP, 0
0003 | 0029 * SAVE FIRST FOUR BYTES IN REGISTERS
0003 C1 | 0030 POP B
0004 D1 | 0040 POP D
0005 | 0049 * REPLACE BYTE 0 WITH A 'RETURN'
0005 2E C9 | 0050 MVI L, 0C9H
0007 F3 | 0051 DI
000B E5 | 0060 PUSH H
0009 E5 | 0070 PUSH H
000A 00 | 0080 NOP
000B 00 | 0081 NOP
000C 00 | 0082 NOP
000D 31 04 00 | 0090 LXI SP, 4
0010 CD 00 00 | 0100 CALL 0
0013 | 0101 * ROM LOCATION NOW IN BYTE 3
0013 31 02 00 | 0110 LXI SP, 2
0016 E1 | 0120 POP H
0017 | 0129 * RETURN BYTES 0-3
0017 31 04 00 | 0130 LXI SP, 4
001A D5 | 0140 PUSH D
001B C5 | 0150 PUSH B
001C | 0159 * STORE ROM LOCATION IN SP
001C F9 | 0160 SPHL
001D 0E 00 | 0170 MVI C, 0
001F 59 | 0180 MOV E, C
0020 69 | 0190 MOV L, C
0021 | 0199 * INPUT SENSE SW COMMANDS
0021 DB FF | 0200 IN 255
0023 57 | 0210 MOV D, A
0024 | 0219 * STRIP RAM ADDRESS
0024 E6 07 | 0220 ANI 7
0026 07 | 0230 RLC
0027 07 | 0240 RLC
0028 | 0249 * STORE RAM ADDRESS IN BC
0028 47— | 0250 MOV B, A
0029 7A | 0260 MOV A, D
002A | 0269 * STRIP ROM ADDRESS
002A E6 38 | 0270 ANI 56
002C OF | 0280 RRC
002D 00 | 0290 NOP
002E 67 | 0300 MOV H, A
002F 39 | 0310 DAD SP
0030 2E 00 | 0320 MVI L, 0
0032 7A | 0330 MOV A, D
0033 EB | 0340 XCHG
0034 | 0341 * ADDRESS OF ROM BEING PROCESSED IN DE
0034 | 0349 * BRANCH TO TRANSFER OR PROGRAM ROUTINE
ROUTINE TO TRANSFER ROM TO RAM

SPHL

ADI 45

LXI H, 0

MOV L, A

DAD SP

PCHL

XP

ROUTINE TO PROGRAM ROM

MOV A, D

RLC

ADD L

MOV L, A

PCHL

* ROUTINE TO PROGRAM ROM

MOV H, A

MVI L, 107

INX H

DCX SP

XCHG

LXI H, 11

PCHL

STACK CONTAINS ROM LOCATION

XCHG H&L CONTAIN LOOP ADDRESS

LXI D, 0

* START OF TRANSFER LOOP

DCX SP

INX D

MOV A, D

ANI 4

RLC

MOV L, A

PCHL

* ROUTINE TO TRANSFER ROM TO RAM

SPHL

MOV L, A

PCHL

STACK CONTAINS ROM LOCATION

XCHG H&L CONTAIN LOOP ADDRESS

LXI D, 0

* INCENTRAM ROM ADDRESS

DCX SP

* MOVE DATA FROM ROM TO RAM

MOV A, D

RLC

ADD L

MOV L, A

PCHL

* INCREMENT RAM ADDRESS

INX B

DCX SP

* USE STAX AND POP 6 (PSW)
006C 1099 * TO MOVE DATA FROM ROM TO RAM
006C F1 1100 POP 6
006D 12 1110 STAX D
006E 1119 * INCREMENT ROM ADDRESS
006E 13 1120 INX D
006F 1129 * INCREMENT BYTE COUNT
006F 03 1130 INX B
0070 1138 * B STORES TWO CONSTANTS
0070 1139 * # COMPLETE PASSES & IN ROM CNT
0070 7B 1140 MOV A,B
0071 1149 * # PASSES = 32 ?
0071 FE FC 1150 CPI 252
0073 3F 1160 CMC
0074 1F 1170 RAR
0075 1F 1180 RAR
0076 1198 * SET 64 TO 0 FOR TWO MINUTE TIMER VERSION
0076 E6 40 1200 ANI 64
0078 1201 * A=64 IF COMPLETED 32 PASSES
0078 2E 7D 1205 MVI L, 7DH
007A 85 1210 ADD L
007B 6F 1220 MOV L,A
007C E9 1225 PCHL
007D 2E 6B 1226 MVI L, 6BH
007F 7B 1230 MOV A,B
0080 E6 04 1240 ANI 4
0080 1241 * A=4 IF END OF 1024 BYTE PASS
0080 07 1250 RLC
0083 07 1260 RLC
0084 07 1270 RLC
0085 85 1280 ADD L
0086 6F 1290 MOV L,A
0087 1291 * GO BACK TO 1090 UNLESS OVERFLOW
0087 1292 * THEN GO TO 1380 FOR
0087 1293 * ADDRESS SUBTRACTION
0087 1294 * OR 2135 FOR QILTS
0087 E9 1300 PCHL
0088 00 1350 NOP
0089 00 1360 NOP
008A 00 1370 NOP
008B 1378 * ANOTHER PROGRAM PASS TO BE DONE
008B 1379 * ADJUST ROM AND RAM ADDRESSES
008B 7C 1380 MOV A,H
008C 21 00 FC 1390 LXI H, 64512
008F 1399 * SUBTRACT 1024 FROM ROM ADDRESS
008F 39 1400 DAD H
0090 F9 1410 SPHL
0091 21 00 FC 1420 LXI H, 64512
0094 1429 * SUBTRACT 1024 FROM RAM ADDRESS
0094 19 1430 DAD D
0095 EB 1440 XCHG
0096 67 1450 MOV H,A
0097 2E 6B 1460 MVI L, 107
0099 7B 1470 MOV A,B
009A E6 FB 1480 ANI 248
009C 1489 * INCREMENT PASS CONTER BY ONE
009C C6 0B 1490 ADI 8
BYTEMOVER ASSEMBLY LISTING

MOV 8, A

* GO BACK TO 1090

PCHL

ROUTINE TO LOAD BYEMOVER INTO ROM IN 255

MOV B, A

ANI 224

MVI E, 0

MOV C, E

MOV D, A

ANI 31

MOV B, A

MOV H, A

MOV L, 96

MOV L, A

IN 255

ANI 64

RRC

RRC

ADD L

PCHL

* CHECK FOR 7K TRANSFER OF ROM TO RAM

ADI lAH

MOV L, A

PCHL

* PROGRAMMER VERIFICATION ROUTINE

MOV A, H

LXI H, 64512

DAD SP

SPHL

MVI L, OCDH

MOV H, A

LXI B, 0

PCHL

* ROM TO RAM TRANSFER STOP ROUTINE

SUI 90H

MOV L, A

* PROGRAMMER VERIFICATION ROUTINE

MOV A, H

LXI H, 64512

DAD 0

XCHG

MVI L, OF1H

MOV H, A

LXI B, 0

PCHL

SUI 90H

MOV L, A
BYTEMOVER ASSEMBLY LISTING

OAD 7A 2650 MOV A, D
00DF C6 04 2660 ADI 4
00E1 57 2670 MOV D, A
00E2 FE 38 2680 CPI 56
00E4 3F 2685 CMC
00E5 3E 00 2690 MVI A, 0
00E7 1F 2700 RAR
00E8 85 2710 ADD L
00E9 6F 2720 MOV L, A
00EA E9 2730 PCHL
00EB 2879 * ROM PROGRAMMER STOP ROUTINE
00EC 00 2880 NOP
00ED FB 2881 NOP
00EE E9 2885 EI
00EF E9 2890 PCHL
00F0 E9 2900 PCHL
00F1 2918 * PROGRAMMER VERIFICATION ROUTINE
00F1 2919 * PART 3
00F1 3B 2920 DCX SP
00F2 F1 2930 POP 6
00F3 EB 2940 XCHG
00F4 2949 * COMPARE FOR GREATER
00F4 BE 2950 CMP M
00F5 EB 2960 XCHG
00F6 17 2970 RAL
00F7 E6 01 3000 ANI 1
00F9 2F 3010 CMA
00FA 3C 3011 INR A
00FB 85 3015 ADD L
00FC 6F 3020 MOV L, A
00FD 3B 3030 DCX SP
00FE 3B 3040 DCX SP
00FF 3050 * COMPARE FOR LESSER
00FF F1 3055 POP 6
0100 2F 3056 CMA
0101 EB 3058 XCHG
0102 86 3059 ADD M
0103 EB 3060 XCHG
0104 C6 07 3061 ADI A, 1
0106 3F 3065 CMC
0107 17 3070 RAL
0108 E6 01 3090 ANI 1
010A 2F 3100 CMA
010B 3C 3101 INR A
010C 85 3105 ADD L
010D 6F 3110 MOV L, A
010E 03 3130 INX B
010F 13 3140 INX D
0110 7B 3150 MOV A, B
0111 E6 04 3180 ANI 4
0113 2F 3190 CMA
0114 3C 3191 INR A
0115 85 3195 ADD L
0116 6F 3200 MOV L, A
0117 E9 3210 PCHL
PIN CONFIGURATIONS

PIN NAMES

BLOCK DIAGRAM

READ OPERATION

DC & Operating Characteristics

\( T_A = 0^\circ \text{C} \) to \( 70^\circ \text{C} \), \( V_{CC} = +5V \pm 5\% \), \( V_{DD} = +12V \pm 5\% \), \( V_{BB} = -5V \pm 5\% \), \( V_{SS} = 0V \), Unless Otherwise Noted.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter Description</th>
<th>Min.</th>
<th>Typ.[1]</th>
<th>Max.</th>
<th>Unit</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{LI} )</td>
<td>Address and Chip Select Input Load Current</td>
<td></td>
<td>10</td>
<td>( \mu A )</td>
<td></td>
<td>( V_{IN} = 5.25V )</td>
</tr>
<tr>
<td>( I_{LO} )</td>
<td>Output Leakage Current</td>
<td></td>
<td>10</td>
<td>( \mu A )</td>
<td></td>
<td>( V_{OUT} = 5.25V, \overline{CS}/WE = 5V )</td>
</tr>
<tr>
<td>( I_{DD} )</td>
<td>( V_{DD} ) Supply Current</td>
<td>50</td>
<td>65</td>
<td>mA</td>
<td></td>
<td>Worst Case Supply Currents:</td>
</tr>
<tr>
<td>( I_{CC} )</td>
<td>( V_{CC} ) Supply Current</td>
<td>6</td>
<td>10</td>
<td>mA</td>
<td></td>
<td>All Inputs High</td>
</tr>
<tr>
<td>( I_{BB} )</td>
<td>( V_{BB} ) Supply Current</td>
<td>30</td>
<td>45</td>
<td>mA</td>
<td></td>
<td>( \overline{CS}/WE = 5V; T_A = 0^\circ \text{C} )</td>
</tr>
<tr>
<td>( V_{IL} )</td>
<td>Input Low Voltage</td>
<td>( V_{SS} )</td>
<td>0.65</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{IH} )</td>
<td>Input High Voltage</td>
<td>3.0</td>
<td>( V_{CC}+1 )</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{OL} )</td>
<td>Output Low Voltage</td>
<td></td>
<td>0.45</td>
<td>V</td>
<td></td>
<td>( I_{OL} = 1.6mA )</td>
</tr>
<tr>
<td>( V_{OH1} )</td>
<td>Output High Voltage</td>
<td></td>
<td>3.7</td>
<td>V</td>
<td></td>
<td>( I_{OH} = -100\mu A )</td>
</tr>
<tr>
<td>( V_{OH2} )</td>
<td>Output High Voltage</td>
<td></td>
<td>2.4</td>
<td>V</td>
<td></td>
<td>( I_{OH} = -1mA )</td>
</tr>
<tr>
<td>( P_D )</td>
<td>Power Dissipation</td>
<td></td>
<td>800</td>
<td>mW</td>
<td></td>
<td>( T_A = 70^\circ \text{C} )</td>
</tr>
</tbody>
</table>

NOTES:

1. Typical values are for \( T_A = 25^\circ \text{C} \) and nominal supply voltages.
2. The program input (Pin 18) may be tied to \( V_{SS} \) or \( V_{CC} \) during the read mode.
Waveforms

(Logic levels and timing reference levels same as in the Read Mode unless noted otherwise.)

Program Mode

CS/WE = +12V

Read/Program/Read Transitions
**AC Characteristics**

$T_A = 0^\circ C$ to $70^\circ C$, $V_{CC} = +5V \pm 5\%$, $V_{DD} = +12V \pm 5\%$, $V_{BB} = -5V \pm 5\%$, $V_{SS} = 0V$, Unless Otherwise Noted.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{ACC}$</td>
<td>Address to Output Delay</td>
<td>280</td>
<td>450</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{CO}$</td>
<td>Chip Select to Output Delay</td>
<td>120</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{DF}$</td>
<td>Chip De-Select to Output Float</td>
<td>0</td>
<td>120</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>$t_{OH}$</td>
<td>Address to Output Hold</td>
<td>0</td>
<td></td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

**Capacitance**

$T_A = 25^\circ C$, $f = 1MHz$

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{IN}$</td>
<td>Input Capacitance</td>
<td>4</td>
<td>6</td>
<td>pF</td>
<td>$V_{IN}=0V$</td>
</tr>
<tr>
<td>$C_{OUT}$</td>
<td>Output Capacitance</td>
<td>8</td>
<td>12</td>
<td>pF</td>
<td>$V_{OUT}=0V$</td>
</tr>
</tbody>
</table>

Note 1. This parameter is periodically sampled and not 100% tested.

**Typical Characteristics**

(Nominal supply voltages unless otherwise noted):

- **Output Sink Current vs. Output Voltage**
- **Access Time vs. Load Capacitance**
- **Range of Supply Currents vs. Temperature**
**PROGRAMMING OPERATION**

**Description**

Initially, and after each erasure, all bits of the 2708/2704 are in the "1" state (Output High). Information is introduced by selectively programming "0" into the desired bit locations.

The circuit is set up for programming operation by raising the CS/WE input (Pin 20) to +12V. The word address is selected in the same manner as in the read mode. Data to be programmed are presented, 8-bits in parallel, to the data output lines (O1-O8). Logic levels for address and data lines and the supply voltages are the same as for the read mode. After address and data set up one program pulse (Vp) per address is applied to the program input (Pin 18). One pass through all addresses to be programmed is defined as a program loop. The number of loops (N) required is a function of the program pulse width (tpw) according to:

\[ N \times tpw \geq 100 \text{ ms} \]

For program verification, program loops may be alternated as shown on page 12.

**Program Characteristics**

\( T_A = 25^\circ C, V_{CC} = +5V \pm 5\%, V_{DD} = +12V \pm 5\%, V_{BB} = -5V \pm 5\%, V_{SS} = 0V, \overline{CS}/\overline{WE} = +12V, \) Unless Otherwise Noted.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{AS} )</td>
<td>Address Setup Time</td>
<td>10</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{CSS} )</td>
<td>CS/WE Setup Time</td>
<td>10</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{DS} )</td>
<td>Data Setup Time</td>
<td>10</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{AH} )</td>
<td>Address Hold Time</td>
<td>1</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{CH} )</td>
<td>CS/WE Hold Time</td>
<td>.5</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{OH} )</td>
<td>Data Hold Time</td>
<td>1</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{DF} )</td>
<td>Chip Deselect to Output Float Delay</td>
<td>0</td>
<td>120</td>
<td></td>
<td>( ns )</td>
</tr>
<tr>
<td>( t_{PR} )</td>
<td>Program To Read Delay</td>
<td>10</td>
<td></td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{PW} )</td>
<td>Program Pulse Width</td>
<td>.1</td>
<td>1.0</td>
<td></td>
<td>( ms )</td>
</tr>
<tr>
<td>( t_{PR} )</td>
<td>Program Pulse Rise Time</td>
<td>.5</td>
<td>2.0</td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( t_{PF} )</td>
<td>Program Pulse Fall Time</td>
<td>.5</td>
<td>2.0</td>
<td></td>
<td>( \mu s )</td>
</tr>
<tr>
<td>( I_p )</td>
<td>Programming Current</td>
<td>10</td>
<td>20</td>
<td></td>
<td>( mA )</td>
</tr>
<tr>
<td>( V_p )</td>
<td>Program Pulse Amplitude</td>
<td>25</td>
<td>27</td>
<td></td>
<td>( V )</td>
</tr>
</tbody>
</table>

This manual produced by Laurel Publications (408) 353-3609.
Your factory-built Bytesaver is warranted against defects in materials and workmanship for a period of 90 days from the date of delivery. We will repair or replace products that prove to be defective during the warranty period provided that they are returned to Cromemco. No other warranty is expressed or implied. We are not liable for consequential damages.

Should your factory-built Bytesaver fail after the warranty period it will be repaired, provided that it is returned to Cromemco, for a fixed service fee. We reserve the right to refuse to repair any product that in our opinion has been subject to abnormal electrical or mechanical abuse. The service fee is currently $35 and is subject to change.

Your assembled Bytesaver kit will be repaired, provided that it is returned to Cromemco, for a fixed service fee. We reserve the right to refuse repair of any kit that in our opinion has not been assembled in a workmanlike manner or has been subject to abnormal electrical or mechanical abuse. Payment of the service fee must accompany the returned merchandise. The service fee is currently $35 and is subject to change.
NOTE:
1. DASHED MODIFICATION FOR ROM RESIDENT INTERRUPT 
   OPERATION (NOT INCLUDED ON STANDARD CARD)
2. ALL DIODES ARE IDENTICAL (IN914 OR EQUIV.)