

**32K**  
**Bytesaver**  
**Instruction**  
**Manual**

Five Dollars

CROMEMCO

32K BYTESAVER

Instruction Manual

CROMEMCO, INC.  
280 Bernardo Avenue  
Mountain View, CA 94040

Part No. 023-0002

Copyright 1978

September 1978

TABLE OF CONTENTS

Section 1:	INTRODUCTION .....	2
Section 2:	OPERATING INSTRUCTIONS .....	4
2.1	SWITCH OPTIONS--AN OVERVIEW .....	4
2.2	ADDRESSING THE 32K BYTESAVER .....	12
2.3	BOARD SELECT / CHIP SELECT .....	13
2.4	SHADOWING ROM SOCKET PAIRS .....	16
2.5	MEMORY BANKS .....	18
2.6	SELECT BANK 0 ON RESET OR POWER-ON-CLEAR ..	24
2.7	DIRECT MEMORY ACCESS .....	24
Section 3:	PROM PROGRAMMING INSTRUCTIONS .....	29
3.1	PROGRAMMING FROM RDOS OR Z-80 MONITOR .....	30
3.2	PROGRAMMING FROM 3K CONTROL BASIC .....	33
3.3	PROGRAMMING FROM Z-80 ASSEMBLY CODE .....	37
Section 4:	THEORY OF OPERATION .....	40
4.1	POWER SUPPLIES .....	40
4.2	ADDRESSING .....	41
4.3	MEMORY READ CYCLES .....	43
4.4	MEMORY WRITE CYCLES .....	44
4.5	DMA CYCLES .....	45
Section 5:	ASSEMBLY INSTRUCTIONS .....	47
5.1	ASSEMBLY STEPS .....	47
5.2	POWER LINE TESTING .....	51
	32K BYTESAVER PARTS LIST .....	53
	32K BYTESAVER PARTS LOCATION DIAGRAM .....	55
	APPLICATION NOTE .....	56
	32K BYTESAVER SCHEMATIC .....	57

Section 1  
INTRODUCTION

This manual provides assembly instructions, operating instructions and theory of operation for the Cromemco 32K BYTESAVER memory board.

The 32K BYTESAVER is an S-100 bus compatible, 32K-byte capacity, 2716-type EPROM memory board and programmer. The 32K BYTESAVER features:

- Independent operation as a 32K-byte ROM memory board.
- Independent operation as a 2716 PROM programmer.
- BANK SELECT allowing memory expansion beyond 64K-bytes.
- ROM SHADOWING allowing external memory to overlap portions of the 32K BYTESAVER's address space.
- Powerful DMA configuration options with DMA OVERRIDE.
- Fully buffered address lines.
- Digital count derived PROGRAM PULSES (no erratic one-shots).
- A separate memory protect switch for each ROM socket.
- All options switch selectable (no soldered jumper wires).

## 32K BYTESAVER INSTRUCTION MANUAL

This manual consists of four basic sections: Operating Instructions, PROM Programming Instructions, Theory of Operation and Assembly Instructions. If you purchased a 32K BYTESAVER kit, the Assembly Instructions provide step-by-step construction and initial test procedures. Section 2.1 of the Operating Instructions provides a 32K BYTESAVER operational overview for those who want to put the board quickly to use.

### TECHNICAL SPECIFICATIONS

#### 32K BYTESAVER PROM CARD

MEMORY CAPACITY:	32K BYTES
MEMORY TYPE:	INTEL 2716, TI 2516 OR EQUIVALENT*
MEMORY ACCESS TIME:	450 NANoseconds
WAIT STATES @ 2 MHZ:	NONE REQUIRED
WAIT STATES @ 4 MHZ:	ONE PER MACHINE CYCLE
BUS COMPATIBILITY:	S-100
POWER REQUIREMENTS:	+8 VOLTS AT 2.1 AMPS (MAX.)
OPERATING ENVIRONMENT:	0-55 DEGREES CENTIGRADE

NOTE: Texas Instrument's 2716 PROM is not equivalent to the Intel 2716 PROM, and thus it may not be used with the 32K BYTESAVER. The TI 2516 is equivalent to the Intel 2716, so it may be used with the 32K BYTESAVER.

Section 2

OPERATING INSTRUCTIONS

Operating the 32K BYTESAVER board involves inserting from one to sixteen 2716 PROM devices in sockets ROM 0 - ROM 15 (any sockets may be used or left unused), setting six switch groups to configure the board, plugging the board into a convenient S-100 bus slot, and then applying system power. To program a PROM, you will additionally need to run software described in Section 3, PROM PROGRAMMING INSTRUCTIONS.

2.1 SWITCH OPTIONS--AN OVERVIEW

The 32K BYTESAVER is configured by setting six switch groups located along the top edge of the board (see Figure 1). To provide an operational overview and for later quick reference, the function of each switch group is briefly explained in this section.

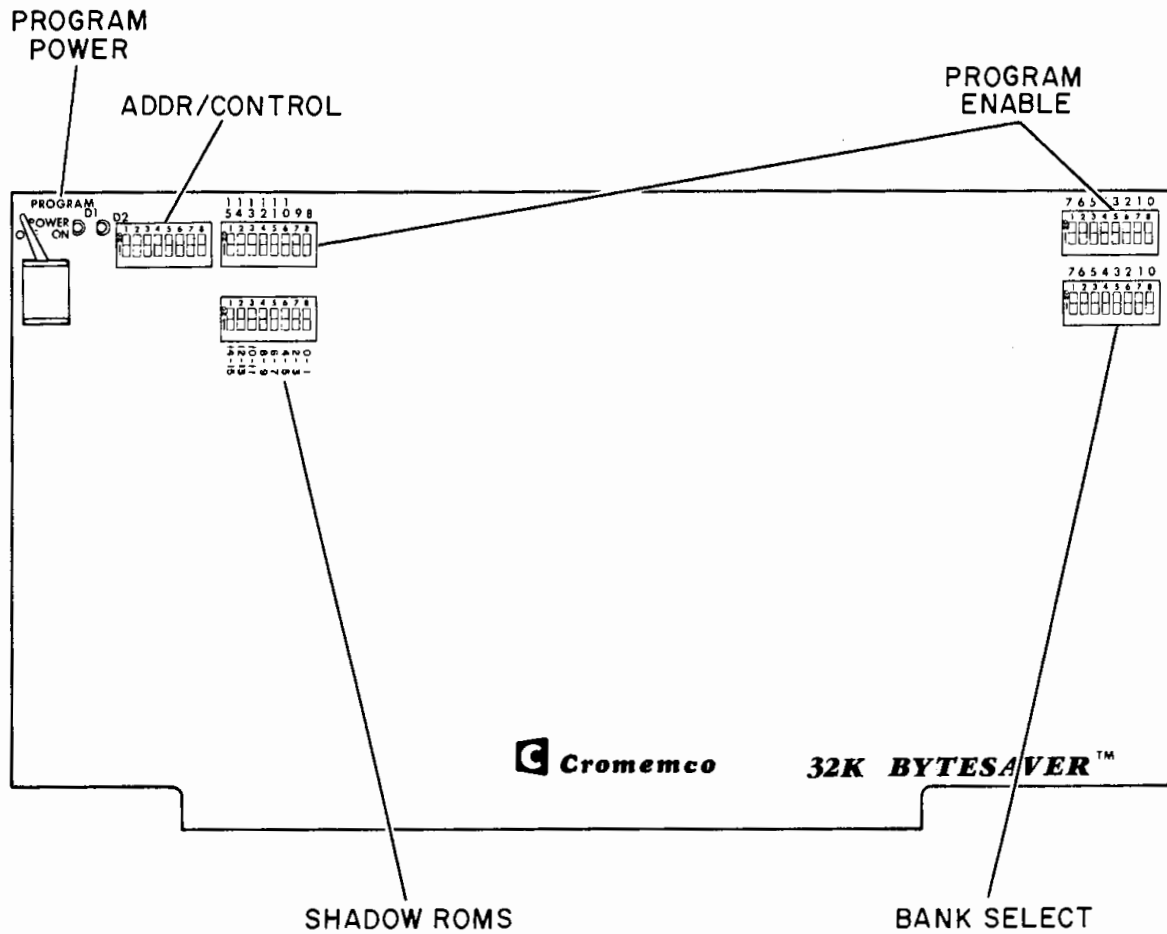


Figure 1: SWITCH LOCATIONS

\* PROGRAM POWER TOGGLE SWITCH

The PROGRAM POWER switch turns the +26 volt dc to dc power supply and the nearby red LED indicator ON and OFF. Position this switch ON before PROM programming; position it OFF when done to prevent inadvertent re-programming.

\* ADDR/CONTROL SWITCHES

The ADDR/CONTROL switches control several different functions (see Figure 2).

Switches 1,2 and 3 are not used.

The BANK ENABLE/DISABLE switch enables multiple 64K memory banks (bank 0 - bank 7) when ON, and disables multiple banks when OFF (normal direct 64K addressing).

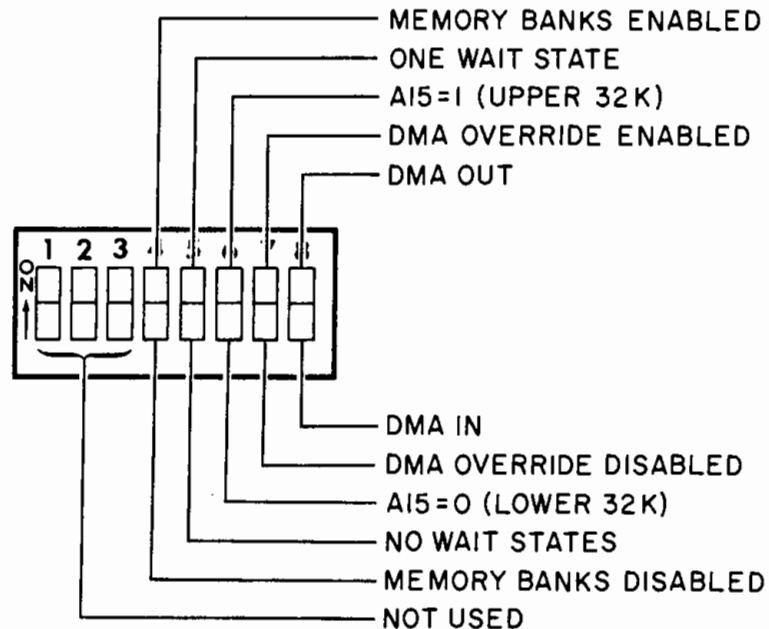


Figure 2: ADDR/CONTROL SWITCHES

The WAIT STATE switch is used to match the CPU cycle



## 32K BYTESAVER INSTRUCTION MANUAL

time to the 2716 PROM 450ns (max) memory access time. Positioning the WAIT STATE switch ON introduces one wait state per machine cycle; the OFF position introduces no wait states. When used in a Cromemco system with a ZPU running at 4 MHz, position the switch ON. The switch may be left OFF when operating at 2 MHz.

The A15 switch memory maps the 32K BYTESAVER board in the lower 32K half of the memory address space (0000H - 7FFFH) when in the OFF position (A15=0), and memory maps the board in the upper 32K half of memory address space (8000H - FFFFH) when in the ON position (A15=1).

The DMA ENABLE/DISABLE switch enables DMA OVERRIDE when ON and disables DMA OVERRIDE when OFF. For normal direct 64K DMA addressing, position the switch OFF. When performing DMA with memory banks enabled, turn the switch ON. The DMA IN/OUT switch is active only when DMA OVERRIDE is enabled. With DMA OVERRIDE enabled, the 32K BYTESAVER will respond directly to a DMA in the board's 16-bit address range by board enabling if DMA is IN, and by board disabling if DMA is OUT, regardless of current active memory bank status at the time. This feature effectively permits the user to define one board out of several stacked in different memory banks as the DMA board (the one with DMA IN), and the boards in other memory banks as non-DMA boards (the ones with DMA OUT).

\* PROGRAM ENABLE SWITCHES

The sixteen PROGRAM ENABLE switches individually enable and disable programming sockets ROM0 thru ROM15. An ON switch enables programming; an OFF switch inhibits programming. These switches may be alternately viewed as MEMORY PROTECT switches, preventing any memory write operations when in the OFF position.

To enable and disable socket programming, associate the board socket numbers (ROM0 - ROM15) with the numerals printed above the two switch DIPs on the p.c. board (15 to the far left, 0 to the far right).

\* BANK SELECT SWITCHES

The eight BANK SELECT switches map the 32K BYTESAVER into any combination of eight possible 64K byte memory banks (bank 0 - bank 7). Setting a BANK SELECT switch ON logically places the board in the correspondingly numbered memory bank; an OFF switch logically removes the board from a bank. Again, associate the bank number with the numerals printed above the BANK SELECT switches on the p.c. board, not the numerals on the DIP proper.

\* SHADOW ROM SWITCHES

Each SHADOW ROM switch controls two ROM sockets (the rightmost controls ROM0 and ROM1, the leftmost controls ROM14 and ROM15) by placing both sockets in the memory map when the switch is OFF, and by removing both sockets ("floating" the board when either socket is addressed) from the memory map when ON. Placing a SHADOW ROM switch ON effectively creates a 4K-byte "hole" in the 32K BYTESAVER address space, into which another memory module may be mapped.

EXAMPLE 1

Suppose you have a 4MHz ZPU and you want your 32K BYTESAVER to reside in the upper 32K of memory. As a standard practice, you decide to program 2716 PROMs in socket ROM0 only; there are no other boards in the upper 32K of memory, so multiple memory banks are not required. Then, for memory read operation, the switch settings would be as shown in Figure 3.

To program a PROM in socket ROM0, all switch settings remain the same except the PROGRAM POWER switch which must be turned ON.

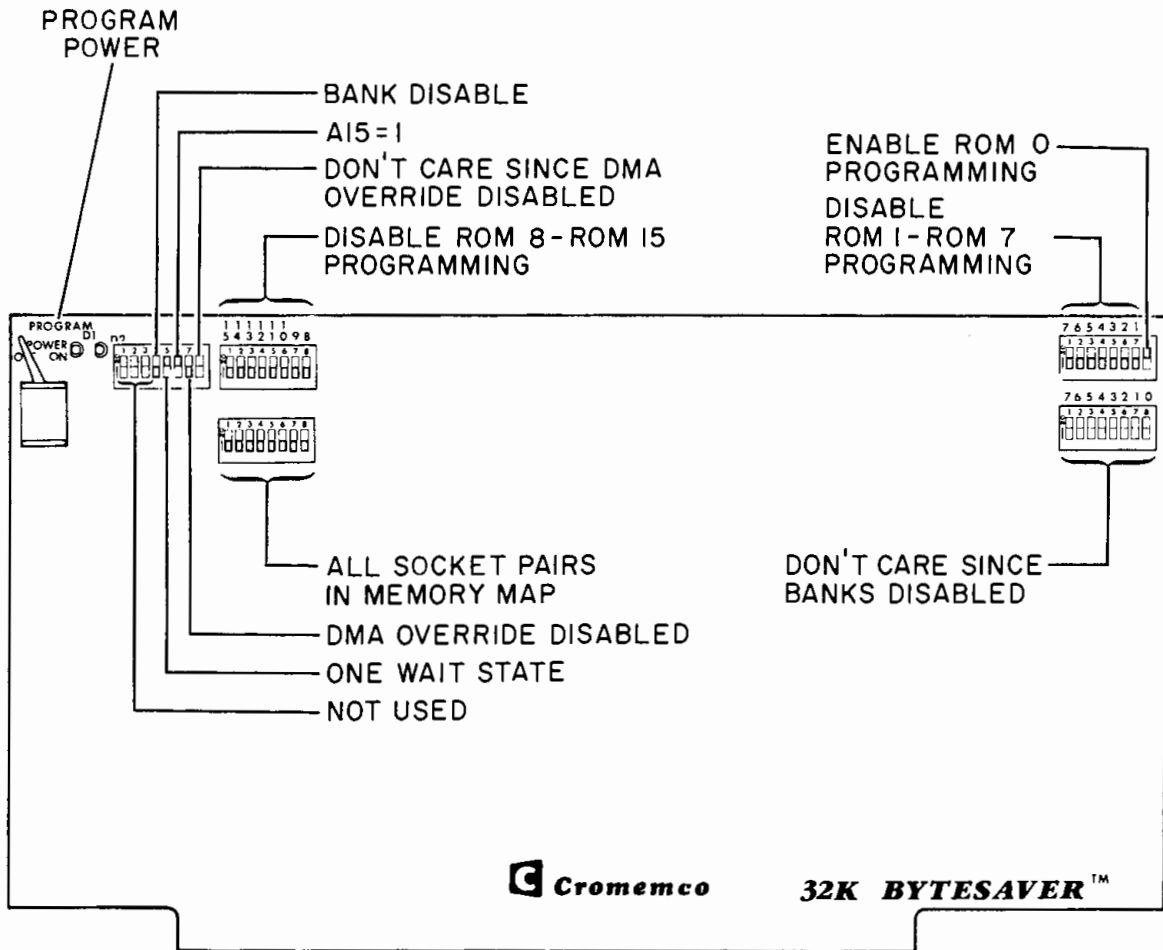


Figure 3: EXAMPLE 1 SWITCH SETTINGS

The following example uses all of the 32K BYTESAVER special features.

EXAMPLE 2

Suppose you have a 2 MHz system and you want the board to reside in the lower 32K of memory. The system has 4K of RAM at overlapping addresses 0000H - 0FFFH, so a 4K "hole" must be created in the 32K BYTESAVER memory map. You intend to program four PROMs at a time in sockets ROM12 - ROM15 (programming 8K blocks of source code to PROM) and a 16K RAM card also resides in the lower 32K of memory solely for DMA transfers (assume this card is assigned to memory bank 1).

You decide to assign the 32K BYTESAVER to memory bank 0 so it will be enabled on a system Power-On-Clear or RESET--see Section 2.6 for details. The appropriate 32K BYTESAVER switch settings for memory read operations are then shown in Figure 4.

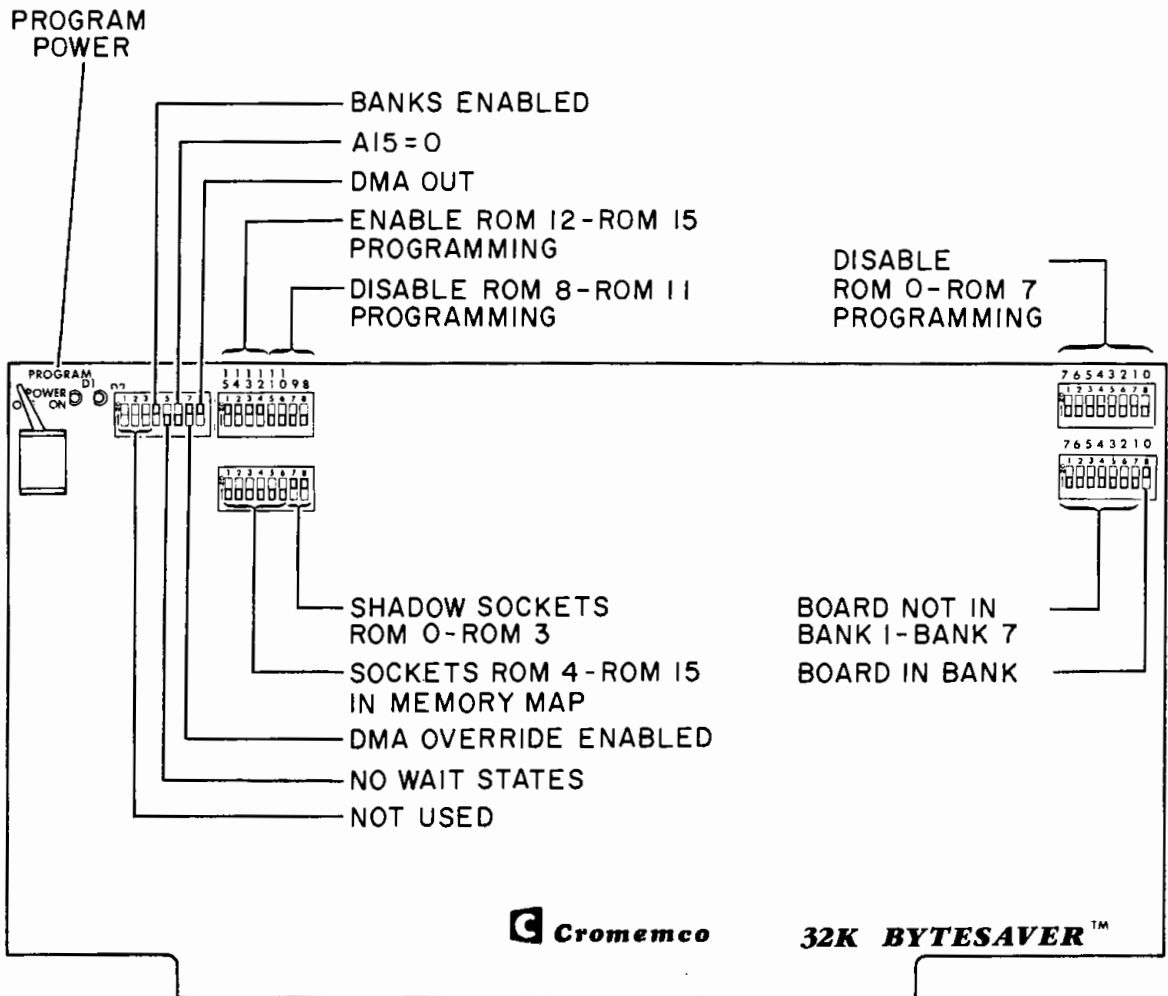


Figure 4: EXAMPLE 2 SWITCH SETTINGS

All 32K BYTESAVER special features and operational modes touched upon in this section are discussed in greater detail in the sections which immediately follow.

## 2.2 ADDRESSING THE 32K BYTESAVER

Addressing a byte on the 32K BYTESAVER involves four levels of selection: choosing a memory bank, a memory board, an IC chip, and finally choosing the byte-on-chip.

Memory banks are addressed by the CPU outputting a control word to an integral OUTPUT PORT 40H contained on each 32K BYTESAVER board. Board, chip and byte-on-chip are all decoded from the sixteen bit address sent out by the CPU on the S-100 bus.

Since the board capacity is 32K bytes, board select is generated by the high order address line A15. There are sixteen ROM sockets, so the next four high order address lines A11 - A14 are used to hardware generate chip enable (selecting ROM0 - ROM15), and the remaining eleven address lines A0 - A10 are used to address one-of-2,048 bytes on a 2716 PROM (see Figure 5).

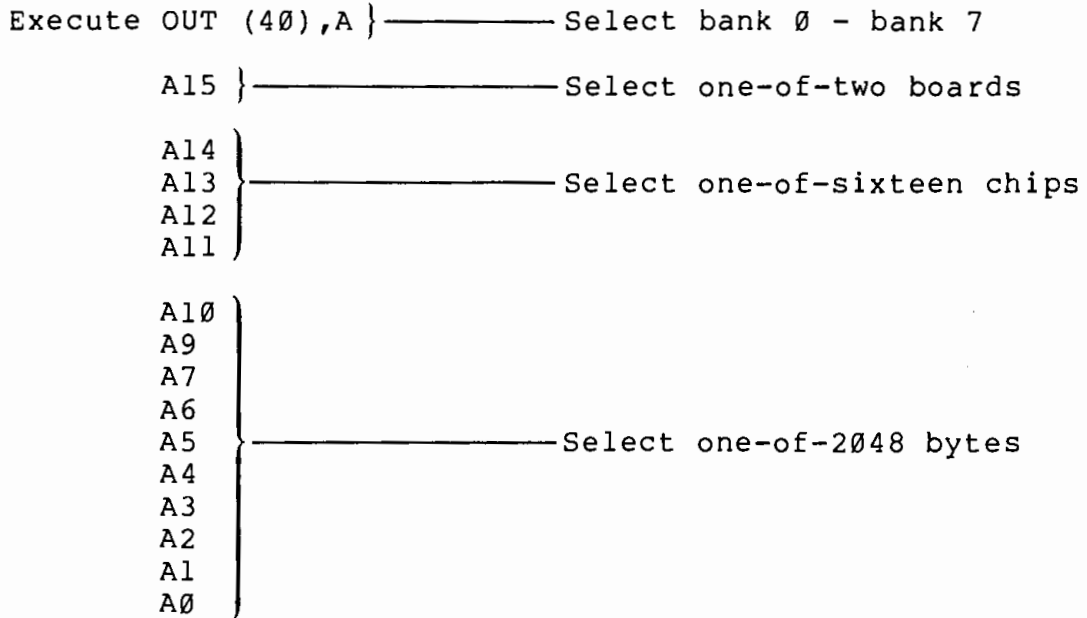


Figure 5: 32K BYTESAVER ADDRESSING

### 2.3 BOARD SELECT/CHIP SELECT

S-100 bus address line A15 is hardware compared to switch A15 in the ADDR/CONTROL switch group. Switch A15 ON corresponds to address line A15=1 (addresses 8000H - FFFFH); switch A15 OFF corresponds to address line A15=0 (addresses 0000H - 7FFFH). These two switch settings place the 32K BYTESAVER in the upper or lower 32K half of the 64K address space, respectively.

Each ROM socket (ROM0 - ROM15) spans a 2K-byte swath of memory. Address lines A11 - A14 feed two one-of-eight

decoders (IC6 and IC7 in the 32K BYTESAVER Schematic) to generate select signals for each ROM socket. The entire 64K address space may then be spanned by two 32K BYTESAVER boards. Figure 6 illustrates such an arrangement along with the address range spanned by each ROM socket.

EXAMPLE 3

Suppose you programmed two 2716 PROMs with Cromemco's Z-80 MONITOR and 3K Control BASIC. The Z-80 MONITOR spans addresses E000H - E3FFH, and Control BASIC spans E400H - EFFFH. To load these programs, you would then place the two programmed PROMs in sockets ROM12 and ROM13 on a 32K BYTESAVER assigned to 8000H - FFFFH with A15 = 1.



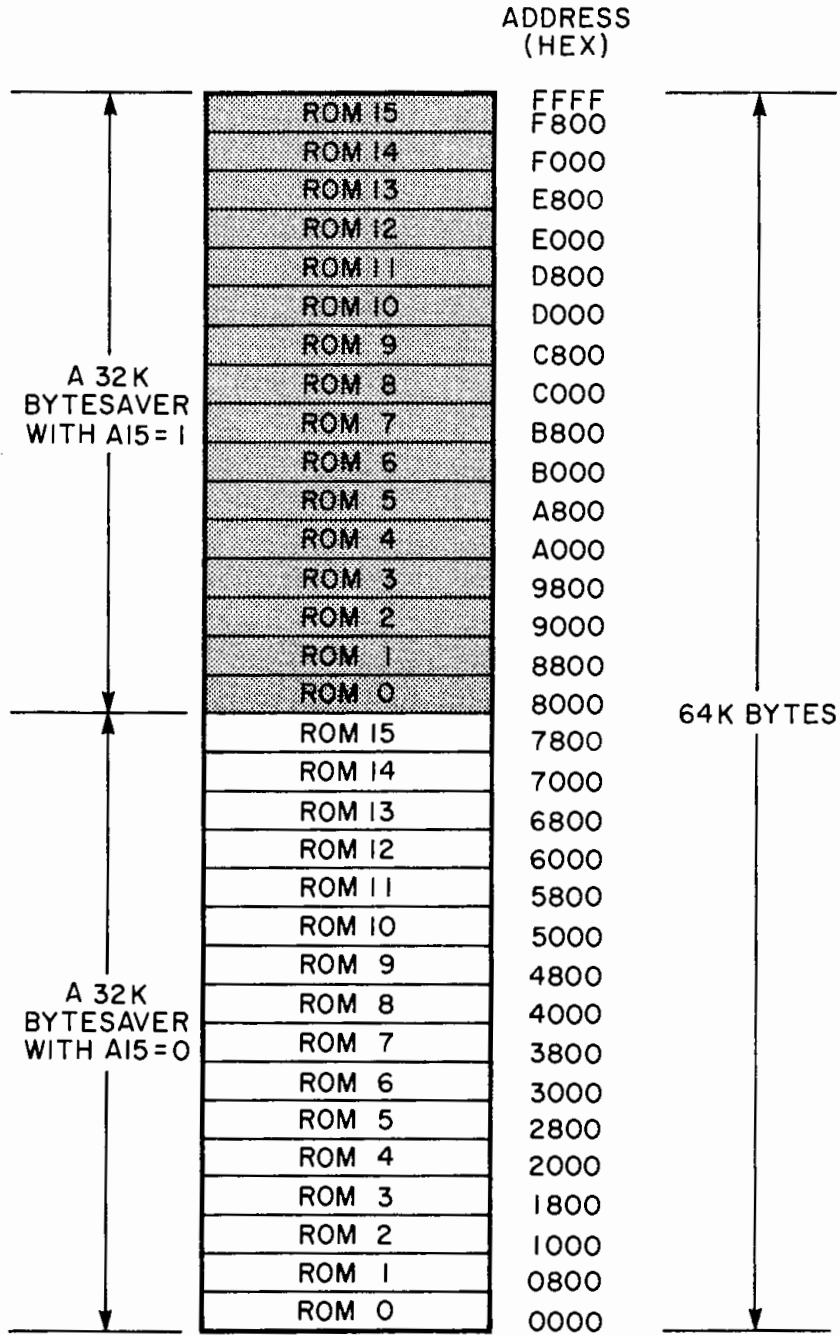


Figure 6: TWO 32K BYTESAVERS SPANNING THE 64K ADDRESS SPACE

2.4 SHADOWING ROM SOCKET PAIRS

A 32K BYTESAVER board spans one half of the CPU's 64K direct addressing range. The eight SHADOW ROM switches allow the user to remove pairs of ROM sockets thereby creating 4K-byte "holes" in the board's memory map which may then be "filled" with other memory modules. Each SHADOW ROM switch controls two vertically separated ROM sockets: the leftmost switch (number 1) controls the two leftmost sockets (ROM 14 and ROM15); the rightmost switch (number 8) controls the two rightmost sockets (ROM0 and ROM1). Positioning a switch ON completely removes a socket pair from the memory map; an OFF switch leaves the socket pair in the map.

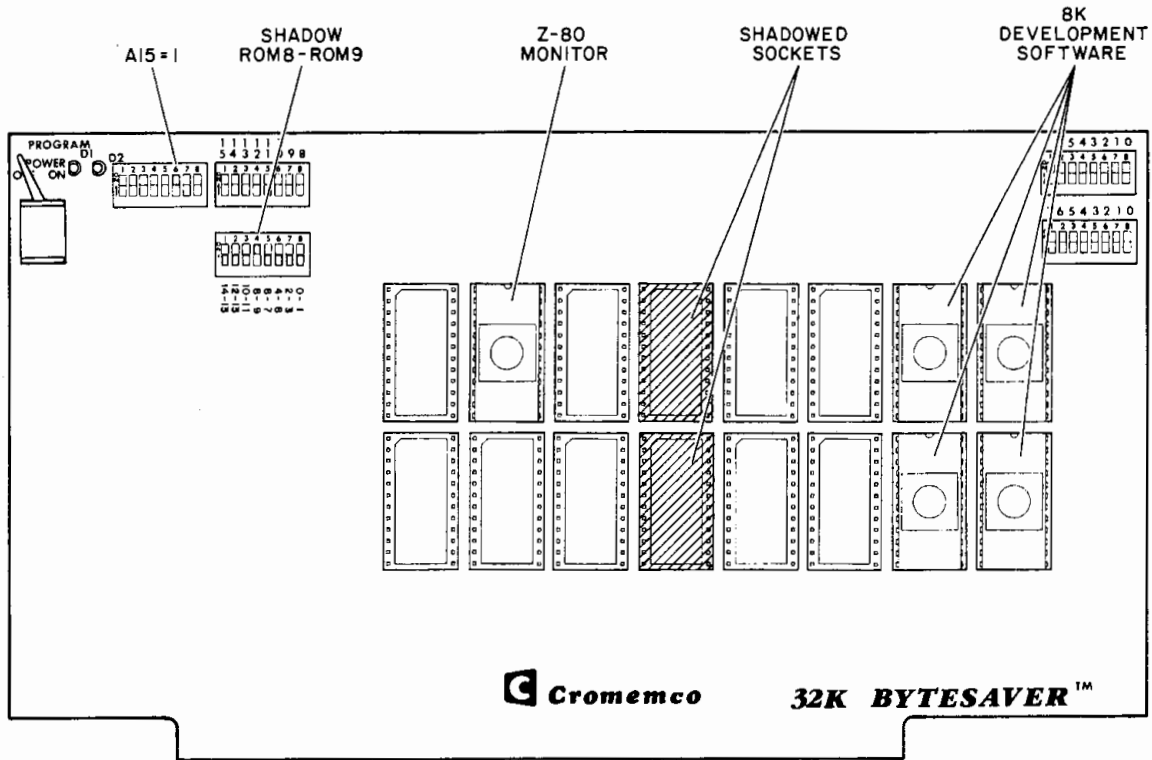
EXAMPLE 4

Suppose you have a Cromemco Disc Operating System (CDOS) with 32K of RAM spanning 0000H - 7FFFH, and a 32K BYTESAVER assigned to 8000H - FFFFH. The Cromemco Floppy Disc Controller board is factory shipped with the RDOS monitor program in ROM memory on the 4FDC board. The RDOS program spans addresses C000H - C3FFH, so you SHADOW sockets ROM8 and ROM9, leaving a hole at C000H - C7FFH which RDOS then partially fills.

Further assume you program one-half of a 2716 with Cromemco's Z-80 Monitor program (spanning E000H - E3FFH), and you program four 2716's with your own 8K-byte development system.

The required switch settings and the resulting memory map are shown in Figure 7.

# 32K BYTESAVER INSTRUCTION MANUAL



SOCKET	CONTENTS	ADDR
		FFFF
ROM 15	EMPTY	F800
ROM 14	EMPTY	F000
ROM 13	EMPTY	E800
ROM 12	USER DEFINED Z-80 MONITOR	E000
ROM 11	EMPTY	D800
ROM 10	EMPTY	D000
SHADOWED {	FLOATING 3K	D000
	RDOS ON 4FDC	C400 C000
ROM 7	EMPTY	C000
ROM 6	EMPTY	B800
ROM 5	EMPTY	B000
ROM 4	EMPTY	A800
ROM 3	8K BYTES	A000
ROM 2	OF	9800
ROM 1	DEVELOPMENT	9000
ROM 0	SOFTWARE	8800
		8000

Figure 7: EXAMPLE 4 SWITCH SETTINGS AND MEMORY MAP

Carefully note that two empty ROM sockets are not equivalent to SHADOWING the same two sockets. The 32K BYTESAVER will memory read data 0FFH from an unSHADOWED empty ROM socket, and as a result will actively drive all eight S-100 data bus lines DI0 - DI7 high. The 32K BYTESAVER tri-states its DI output lines when SHADOWED sockets are addressed, whether they are empty or not.

## 2.5 MEMORY BANKS

BANK SELECT is an optional board feature which effectively allows memory expansion beyond the CPU's 64K direct addressing range. This feature may be completely disabled by switch selecting BANK DISABLE in the ADDR/CONTROL switch group. When this is done, the eight BANK SELECT switch settings become irrelevant. In this mode the 32K BYTESAVER exists only in the upper or lower half of the CPU's 64K direct addressing range for memory read, PROM programming or DMA operations.

To enable memory banks, switch select BANK ENABLE in the ADDR/CONTROL switch group. When this is done, the 32K BYTESAVER is logically placed in one or more 64K-byte memory banks with the eight BANK SELECT switches, and bank addressing is software controlled by executing the OUT (40H),A (or equivalent) Z-80 instruction.

Memory may be stacked up to eight banks deep (see Figure 8). Positioning one or more BANK SELECT switches ON places a 32K BYTESAVER in each corresponding memory bank. On the other hand, positioning all switches OFF completely removes the board from the memory map (except possibly for DMA transfers--see Section 2.7).

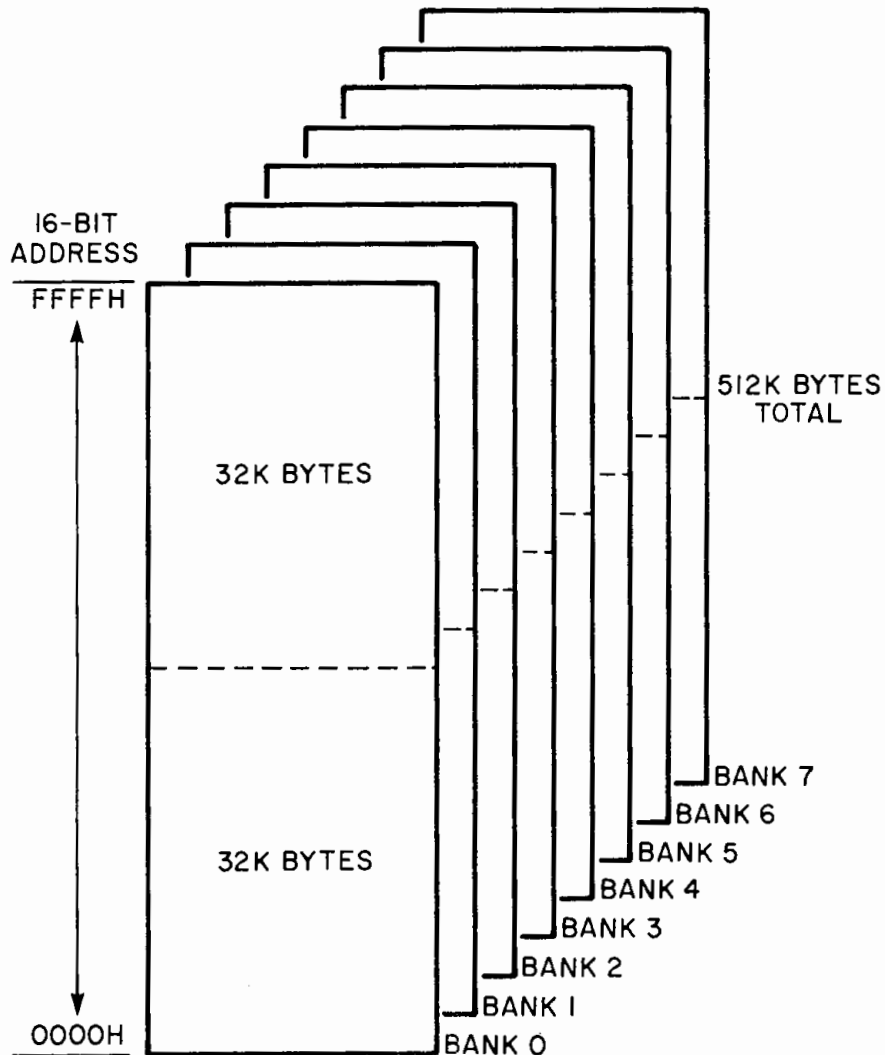


Figure 8: THE MEMORY MAP WITH MULTIPLE MEMORY BANKS

As stated above, memory banks are activated and deactivated under software control. Each 32K BYTESAVER contains an integral OUTPUT PORT 40H which latches the bits of the control byte output to it by the CPU. Each set bit (logic 1) enables its corresponding memory bank, and each reset bit (logic 0) disables its bank. Control byte bit 7 (MSB) controls memory bank 7, bit 6 controls memory bank 6, etc.

If the 32K BYTESAVER is switch mapped into any of the banks activated by the control byte (logical OR), the board responds when addressed and thus is placed "in" the memory map. When this condition occurs, the green LED indicator lights. Conversely, if the 32K BYTESAVER is switch mapped into no bank activated by the output control byte, the board will not respond when addressed and thus is "out" of the memory map. When a control byte inactivates the board, the green LED indicator goes out, and more specifically, the board responds by tri-stating (floating) all of its output lines. This behavior allows two or more memory boards with BANK SELECT to occupy the same or overlapping 16-bit address space but in different memory banks, provided only one board is memory bank active at a time, and all other boards are inactive. Memory bank conflicts may result if:

a) Two or more address overlapping memory boards are switch

32K BYTESAVER INSTRUCTION MANUAL

assigned to the same memory bank, or

b) Two or more 16-bit address overlapping memory boards assigned to disjoint memory banks are simultaneously activated by the same control byte.

EXAMPLE 5

Suppose two 32K BYTESAVERS are both mapped into the upper 32K of memory (A15=1), and their memory bank switches are set as shown in Figure 9. The resulting memory map is then shown in Figure 10.

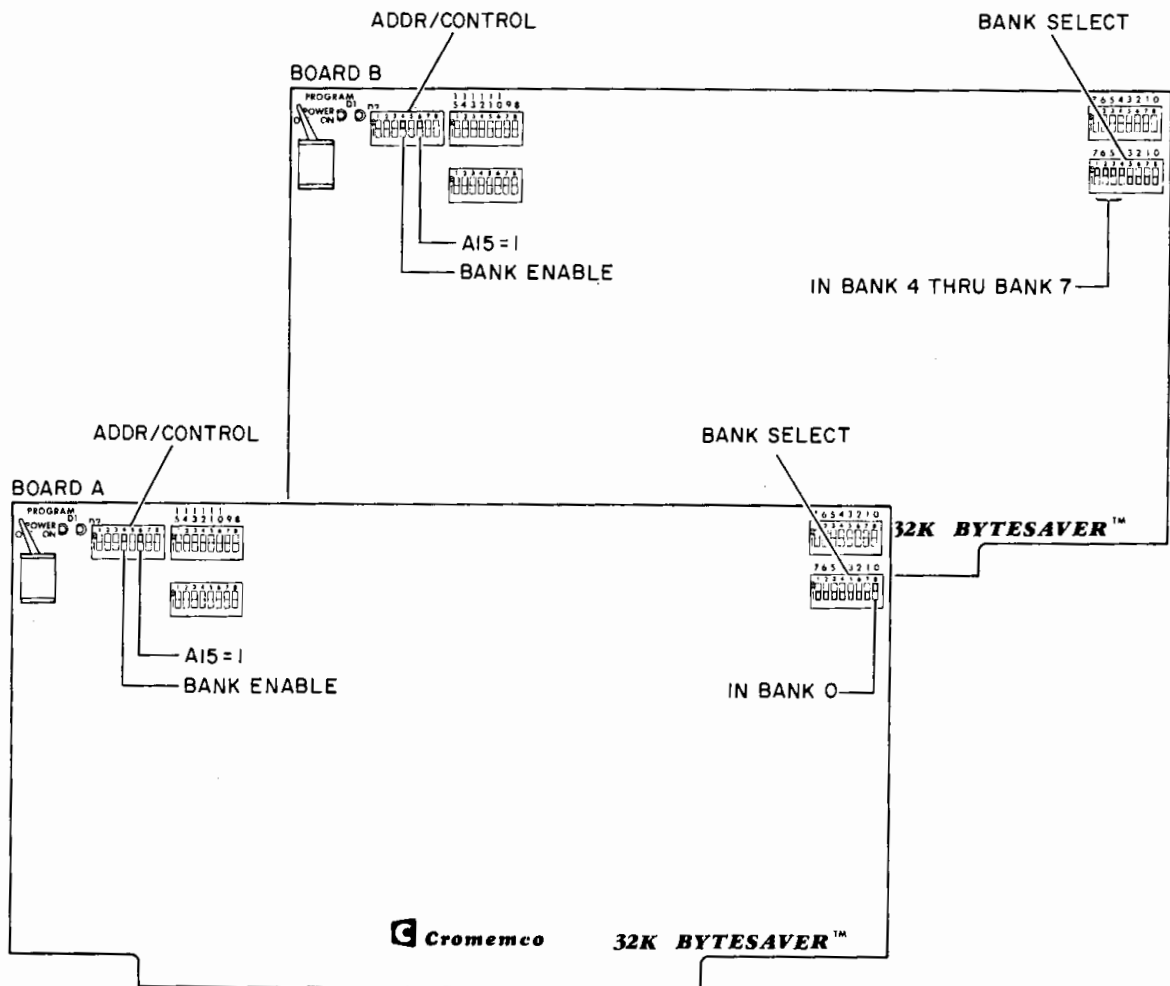


Figure 9: EXAMPLE 5 SWITCH SETTINGS

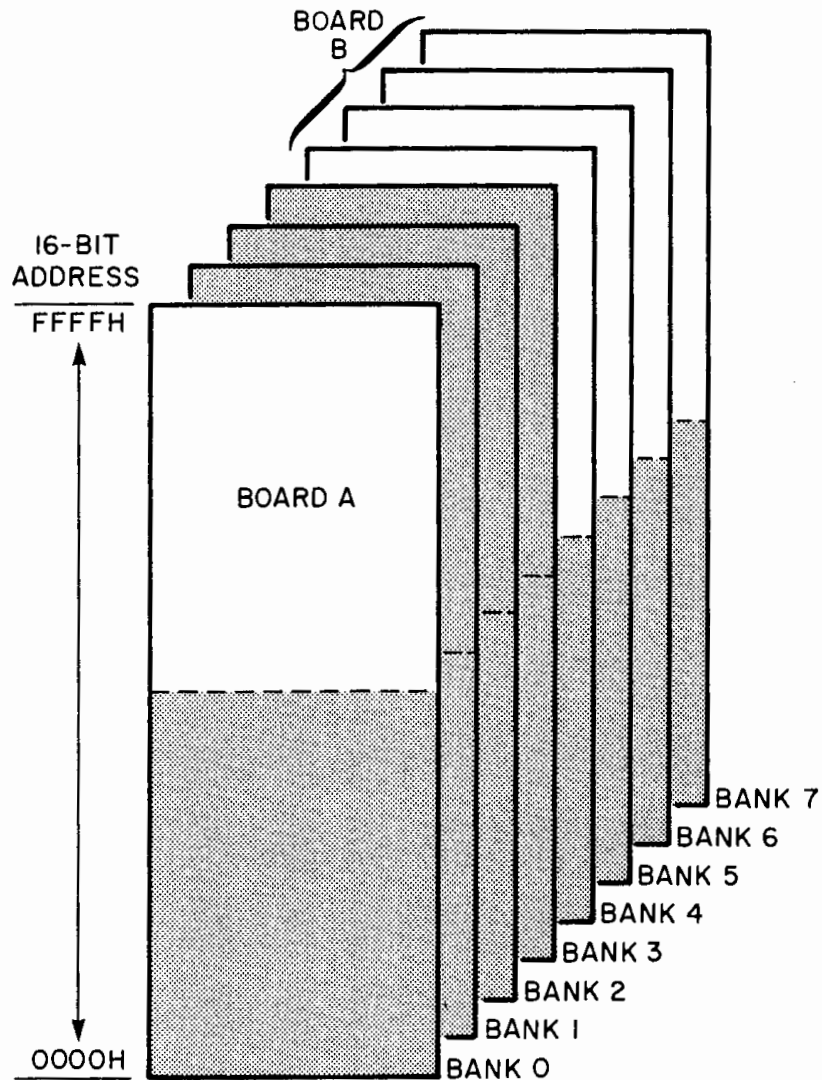


Figure 10: EXAMPLE 5 MEMORY MAP

To continue the same example, the sample programs below illustrate how to memory bank enable and disable the two boards.



## 32K BYTESAVER INSTRUCTION MANUAL

\* Executing the instructions below activates memory banks 2 and 3, and de-activates all other memory banks. The instructions then place both board A and board B in inactive memory banks (both boards inaccessible).

ADDR	OBJECT	MNEMONIC	COMMENT
0000	3E0C	LD A,00001100B	;LOAD 0000 1100 INTO REG. A
0002	D340	OUT (40H),A	;OUTPUT CONTROL BYTE TO PORT 40H
0004	--	--	;NEXT INSTRUCTION

\* Executing the instructions below simultaneously activates both boards A and B, and thus is illegal.

ADDR	OBJECT	MNEMONIC	COMMENT
0000	3E81	LD A,10000001B	;LOAD 1000 0001 INTO REG. A
0002	D340	OUT (40H),A	;OUTPUT CONTROL BYTE TO PORT 40H
0004	--	--	;NEXT INSTRUCTION

\* Executing the instructions below places board A in an active memory bank, and board B in an inactive memory bank (board A available for memory read, PROM programming and DMA transfers; board B inaccessible).

ADDR	OBJECT	MNEMONIC	COMMENT
0000	3E01	LD A,00000001B	;LOAD 0000 0001 INTO REG. A
0002	D340	OUT (40H),A	;OUTPUT CONTROL BYTE TO PORT 40H
0004	--	--	;NEXT INSTRUCTION

\* Executing the instructions below places board A in an inactive memory bank and board B in an active memory bank (board A inaccessible; board B available for memory read, PROM programming and DMA transfers).

ADDR	OBJECT	MNEMONIC	COMMENT
0000	3E60	LD A,01100000B	;LOAD 0110 0000 INTO REG. A
0002	D340	OUT (40H),A	;OUTPUT CONTROL BYTE TO PORT 40H
0004	--	--	;NEXT INSTRUCTION

## 2.6 SELECT BANK 0 ON RESET OR POWER-ON-CLEAR

When system power is first applied, or after a subsequent system RESET, the 32K BYTESAVER will respond in one of two different ways. If BANK SELECT is DISABLED, the board will remain "in" the memory map in the CPU's 64K-byte direct addressing range.

If BANK SELECT is ENABLED, memory bank 0 is automatically hardware activated by a system RESET or a power-on clear (POC), and banks 1 through 7 are de-activated. Thus, a RESET or a POC to the boards in Example 5 would activate board A, and de-activate board B.

## 2.7 DIRECT MEMORY ACCESS

A device may request direct memory access to the 32K BYTESAVER by asserting the S-100 bus line pHOLD low. The CPU grants the request by driving line pHLDA (hold acknowledge) high. When control line pHLDA is high, the device then may directly drive the S-100 bus address lines and control lines (which are tri-stated during DMA transfers when pHLDA is high), and use the data bus lines for reading or writing without CPU intervention. The device may then transfer data at a rate limited only by the memory access time.

The general features of a DMA transfer are then:

- \* Fast asynchronous read or write access to memory.
- \* The DMA device should not be responsible for many overhead tasks (such as memory bank switching) to keep the memory access as quick as possible.
- \* The access is direct--no CPU intervention to slow the transfer.
- \* The DMA device must be capable of controlling and driving the tri-stated address, data and control busses.

In line with this general philosophy, the 32K BYTESAVER's DMA response behavior is controlled by two switches in the ADDR/CONTROL switch group; DMA OVERRIDE and DMA IN/OUT. There are four possible switch setting combinations; each is tabulated and discussed below.

Table 1

DMA OVERRIDE SWITCH	DMA IN/OUT SWITCH	32K BYTESAVER RESPONSE
DISABLED	IN or OUT	Board enables when correctly addressed for either DMA or non-DMA transfers.
ENABLED	OUT	Board enables when correctly addressed for non-DMA transfers (normal operation); board disables during any system DMA transfer.
ENABLED	IN	Board enables when correctly addressed for non-DMA transfers; board enables when the DMA device addresses the board's assigned 32K block of memory, regardless of which banks were active before the DMA request.

The first table entry indicates the board behavior with DMA OVERRIDE DISABLED (note that in this case the DMA IN/OUT

switch setting is irrelevant). Here, the key phrase is "correctly addressed"; the 32K BYTESAVER will respond for memory read, write (PROM programming) or DMA transfers only when it is in an active memory bank (if multiple memory banks are enabled), and the S-100 bus address falls within the board's assigned 32K block of memory. The board in effect does not differentiate between a DMA data transfer and a normal read/write cycle in any way.

The 32K BYTESAVER does differentiate between DMA and non-DMA transfers with DMA OVERRIDE ENABLED, as shown in the last two table entries. A typical application demonstrating how DMA OVERRIDE works is shown in Figure 11.

Here, two 32K BYTESAVERS are assigned to the same 16-bit address space with the A15 switch. Board A is assigned to memory bank 0, and board B to memory bank 1 (any other Cromemco memory boards with BANK SELECT and DMA OVERRIDE could also be used in the example). For non-DMA transfers, both boards are available for read/write operations when correctly addressed (board A is in memory bank 0 at 8000H - FFFFH and board B is in memory bank 1 at 8000H - FFFFH).

32K BYTESAVER INSTRUCTION MANUAL

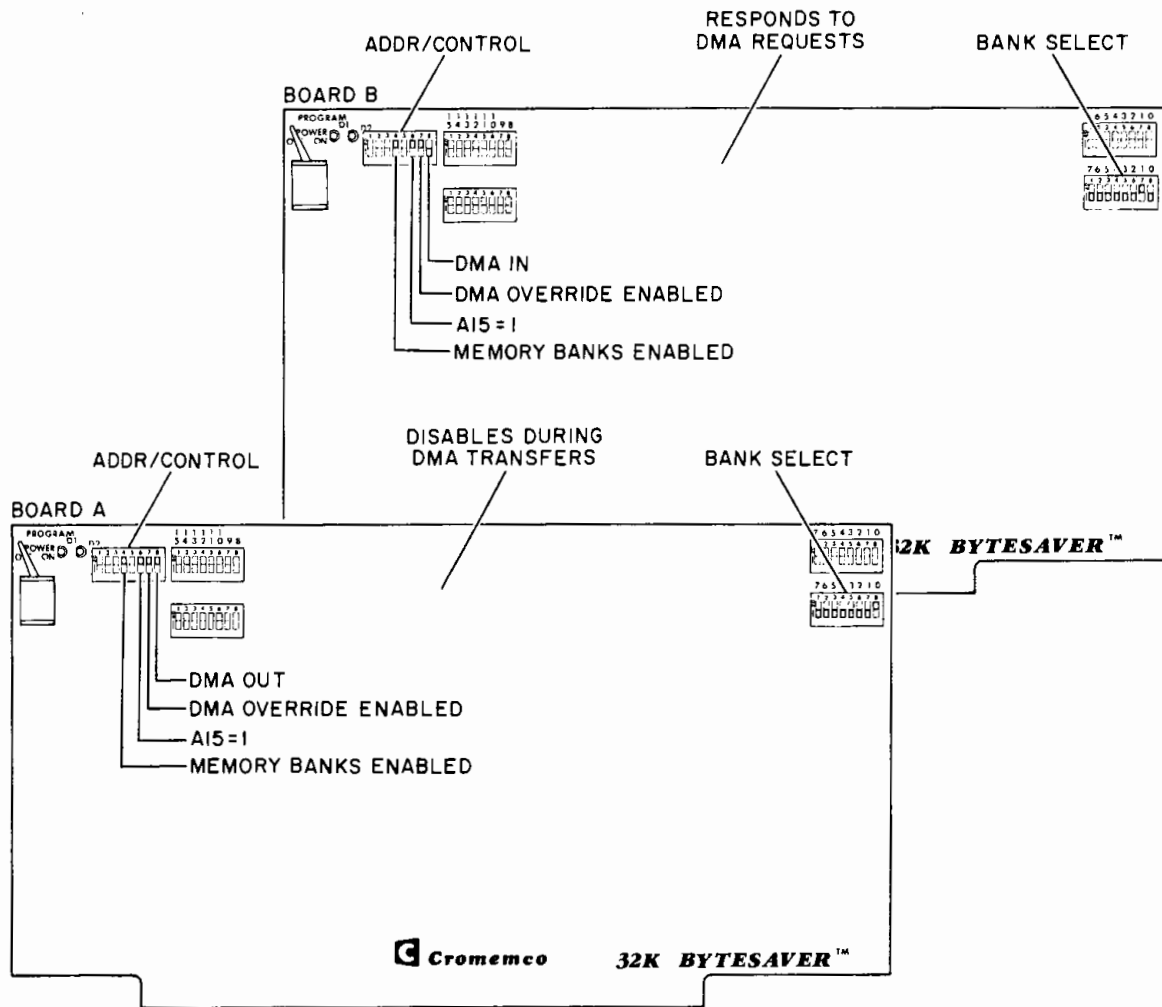


Figure 11: DMA OVERRIDE EXAMPLE CONFIGURATION

When the CPU grants an asynchronous DMA request by driving the pHLDA line high, board A automatically disables and board B enables when the S-100 bus address is in the range 8000H - FFFFH, regardless of which board was in an active memory bank before the request.

Thus, the DMA OVERRIDE feature is seen as a means of overriding logical memory bank boundaries during a DMA transfer. This provides a fast way of vectoring the DMA device to the DMA board (the one with DMA IN) and disabling

## 32K BYTESAVER INSTRUCTION MANUAL

all non-DMA boards (the ones with DMA OUT) without burdening the DMA device with any overhead memory bank switching responsibilities.

It should be noted that after the DMA transfer is completed, both 32K BYTESAVERS revert back to the same memory bank status which existed before the DMA transfer.

Section 3

PROM PROGRAMMING INSTRUCTIONS

The 2716 is a 16,384-bit, ultraviolet light erasable and electrically programmable read-only memory chip. The chip is erased, thereby forcing all bits to the logic 1 state, by exposing the chip's transparent quartz window to intense ultraviolet radiation. Consult the 2716 manufacturer's literature for detailed erasure procedures.

The 2716 is programmed one byte at a time. The bytes may be programmed individually, sequentially, or in random order (contrasting with earlier EPROMs which must be programmed sequentially). The bits in a byte are programmed by selectively changing logic 1 (erased) bits to the logic 0 state. Note that bits may not be programmatically changed from the logic 0 back to the logic 1 state--only complete EPROM erasure can force this transition.

To program a 2716, insert a 2716 into a 32K BYTESAVER socket with the system power OFF, turn ON the PROGRAM POWER switch and PROGRAM ENABLE the ROM socket (never insert or remove the board or board parts while system power is ON). The 32K BYTESAVER hardware then permits you to program a 2716 byte by merely performing a memory write to the target location (or by executing any one of several Cromemco system commands described in the following sections). The 32K

BYTESAVER senses the memory write cycle, forces the CPU to an idle state via the PRDY line until byte programming is complete, properly drives the 2716 with the target address and the program data byte, then applies a digitally timed PROGRAM PULSE. Programming one byte takes approximately 50 msec, thus the entire 2716 may be programmed in approximately  $2,048 \times 50$  msec or about 100 seconds.

Specific 2716 programming examples appear in the next three sections. Section 3.1 illustrates how to program 2716s using Cromemco's RDOS and Z-80 MONITOR system commands, section 3.2 discusses programming using 3K Control BASIC, and section 3.3 deals with programming 2716s from Z-80 Assembly Language code.

### 3.1 PROGRAMMING FROM RDOS OR Z-80 MONITOR

The 2716 bytes may be programmed individually, sequentially or in random order as mentioned above. Thus, the 2716 may be properly thought of as a true read(fast)/write(slow) memory chip, provided a write cycle never attempts to change a logic 0 bit to logic 1 bit. All system commands which move, substitute or display memory may then be used to program and verify 2716s. System commands which memory read the 2716 (e.g., display memory) will be executed at full system speed, but the system commands which memory write to the 2716 (e.g., move memory) will slow



significantly since a write cycle to the 2716 takes approximately 50 msec instead of 750 nsec (with a 4 MHz clock).

It is assumed that a reader of this section is familiar with either Cromemco's RDOS or Z-80 MONITOR programs. In both examples which follow, it is also assumed that:

- a) An erased 2716 has been inserted in socket ROM15 on a 32K BYTESAVER which has been mapped into the upper 32K of memory (A15=1), thus the erased 2716 spans addresses F800H - FFFFH.
- b) The PROGRAM POWER switch has been positioned ON.
- c) Socket ROM15 has been PROGRAM ENABLED.
- d) The user is in either RDOS or Z-80 MONITOR and has just been prompted for a new command.

EXAMPLE 6

To program the entire 2716 with source code residing at 400H - 0BFFH, you would issue the move memory command:

```
M 400 BFF F800<CR>
```

or

```
M 400 S800 F800<CR>
```

where <CR> stands for pressing the RETURN key.

Any discrepancies between the source code and the

programmed 2716 will be printed out. If there are discrepancies, you should again turn ON the PROGRAM POWER switch, and re-program the 2716 with the move memory command as shown above, or use the substitute memory command illustrated in the next example.

#### EXAMPLE 7

Assume you want to program 2716 addresses FC00H, FC01H, FC02H and FC03H with data bytes B8H, B9H, BAH and BBH respectively. Since the 2716 is assumed erased, the current contents of each of these locations is FFH. This task is most easily accomplished using the substitute memory command as illustrated below. In this example, the characters you type are underlined.

```
SM FC00<CR>
```

```
FC00:  FF.B8  FF.B9  FF.BA  FF.BB  FF.<CR>
```

If you then wanted to verify that the four bytes properly programmed the 2716, you would issue a display memory command:

```
DM FC00 FC03<CR>
```

or

```
DM FC00 S4<CR>
```

and the response should be:

```
FC00:  B8 B9 BA BB
```

The Z-80 MONITOR program supports a PROGRAM command which is designed to program 2708-type EPROMs (RDOS does not support this command). While in principle the command will program 2716's, in practice it will take hours to execute due to the large number of passes required to program the 2708.

3.2 PROGRAMMING FROM 3K CONTROL BASIC

3K Control BASIC (CB) program text may be stored in 2716 PROM for subsequent loading and execution by issuing the SAVE command. The SAVE command is normally used for moving the CB program text area to RAM memory, but since the 2716 is functionally equivalent to a read(fast)/write(slow) RAM, the SAVE command may also be used to program 2716 PROMs.

To SAVE a CB program in a 2716 PROM:

- a) Determine the length of the CB program text using the CB SIZE function value.
- b) PROGRAM ENABLE sockets containing erased 2716 PROMs.
- c) Turn the PROGRAM POWER switch ON.
- d) Issue a SAVE ppp command where "ppp" is the 2716 PROM starting "page" address.
- e) After receiving a CB message indicating successful programming, turn the PROGRAM POWER switch OFF.

3K Control BASIC logically partitions memory into "pages", where 1 page = 256 bytes. Pages 0 and 1 (0000H - 01FFH) are not used by CB; pages 2 and 3 (0200H - 03FFH) are used for variables, the input buffer and the stack; pages 4 thru 31 (0400H - 1FFFH) are normally used for CB program text and arrays; and pages 32 on (2000H - end of user RAM) are normally used to save CB program files (see Figure 12).

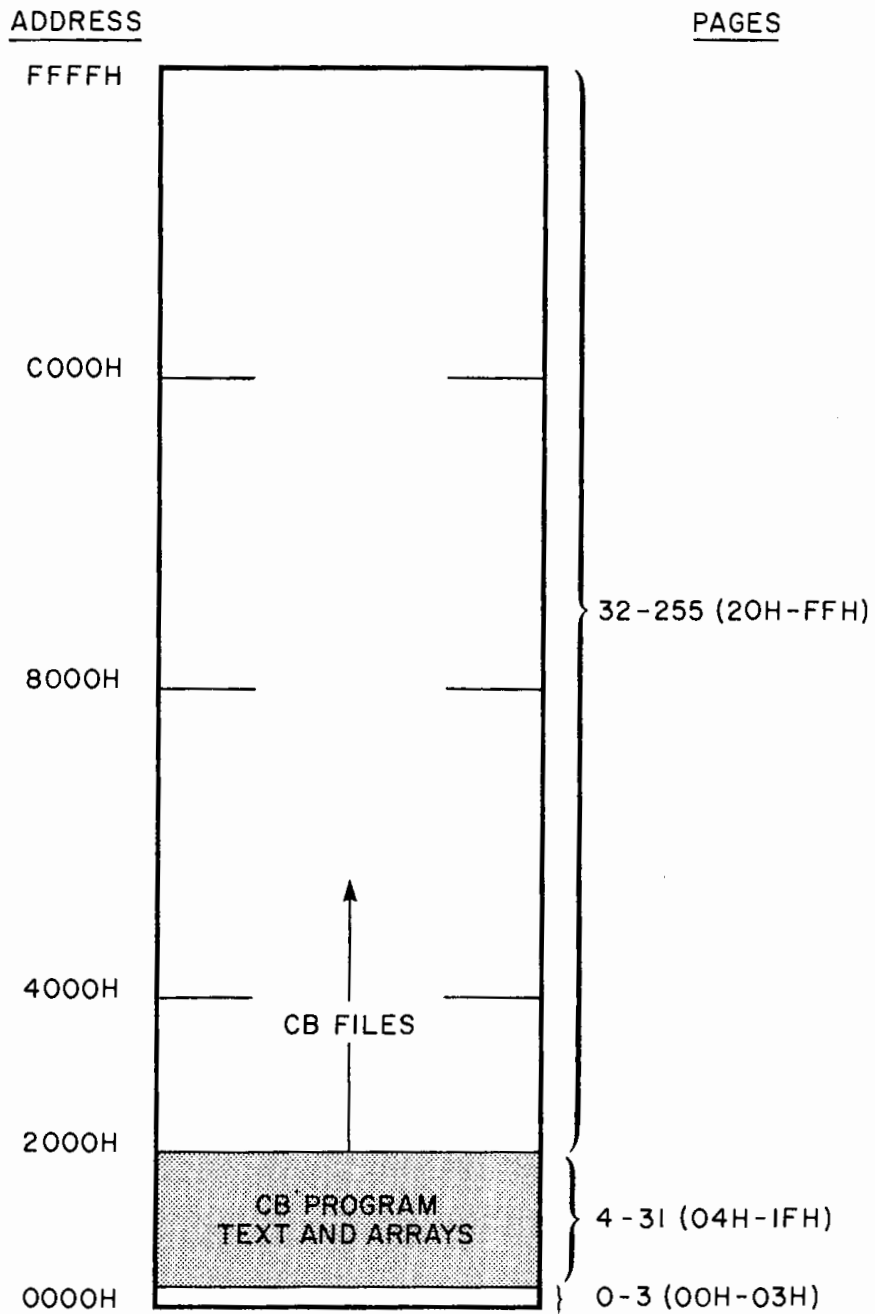


Figure 12: CONTROL BASIC MEMORY MAP

The SAVE ppp command is used to store the CB program text as a file for later execution, and the LOAD ppp command

brings the SAVED file back into the text area for editing.

The page number arguments of the SAVE, RUN and LOAD commands are specified in decimal. For the SAVE command, the page argument is the starting address of an erased 2716 PROM. Sufficient erased PROM should start at this address to contain all of the CB program text. If the CB text does not fill more than 7 pages of an 8 page capacity 2k-byte 2716, the unprogrammed 2716 pages may be filled later with CB program text or other data.

To determine the CB program text length, first clear the text area with the NEW command, then execute the CB program shown below which evaluates and outputs the SIZE function value. The SIZE function evaluates to the number of bytes allocated to, but left unused, by the CB program text.

```
>1 PRINT SIZE
>2 STOP
>RUN
  7142
```

The output (7,142 decimal in this example) gives the size of the unfilled CB text buffer. This number should be recorded for later reference. The size of the unfilled test buffer may be changed using the LOCK ppp command (see Cromemco's 3K Control BASIC Instruction Manual).

To determine the length of any CB program, load the same two statements at the beginning of your program (leave line numbers 1 and 2 free for this purpose), then RUN the combined program resulting in an output like that shown below:

```
>RUN
 5938
```

The program text length is the difference of these two numbers, or,  $7142 - 5938 = 1204$  bytes = 4.7 pages. A 4.7 page program will then occupy 5 of the 2716's 8 page capacity, leaving 3 pages free for other CB program text.

#### EXAMPLE 8

Suppose you wanted to store a 2,500 (decimal) byte CB program in 2716 PROM. Since 2,500 bytes = 9.8 pages, two 2716 PROMs are needed to store the text. Assume two erased PROMs occupy PROGRAM ENABLED sockets ROM14 and ROM15 on a 32K BYTESAVER assigned to the upper 32K of memory (A15=1). The PROMs then reside at F000H - FFFFH, or pages 240 - 255 decimal. You decide to place the CB program "towards the front" of the PROMs on pages 240 - 249, leaving the last six pages 250 - 255 free for later programming.

You would then turn the PROGRAM POWER switch ON, and issue the command:

```
>SAVE 240
```

The programming time is approximately 13 seconds/page, so after about 130 seconds, you would see the message:

```
SAVED ON PAGE %F0 TO %F9
```

```
OK
>
```

if the programmed PROMs verified correctly, or

SORRY

OK  
>

if they did not. If the PROMs do not verify, you may try to re-program them with another SAVE 240 command without damage to the devices, or erase them and try again. Turn OFF the PROGRAM POWER switch after programming.

Assume now you move the two PROMs to 32K BYTESAVER sockets ROM0 and ROM1 (8000H - 8FFFH or pages 128 - 143) for running. Sockets ROM0 and ROM1 should be PROGRAM DISABLED to prevent inadvertent re-programming. To run the program, issue the command:

>RUN 128

Or, to bring the program into the text area for editing, type:

>LOAD 128

3K Control BASIC also supports a 2708-type PROM programming command--EPROM ppp. While in principle this command will also program 2716 devices, in practice it should not be used since it would take hours to execute.

### 3.3 PROGRAMMING FROM Z-80 ASSEMBLY CODE

Moving individual bytes or blocks of system memory to 2716 EPROM is most easily accomplished using RDOS or Z-80 MONITOR commands (see Section 3.1), but there may be instances where it is desirable to program 2716s during the execution of your own Z-80 assembly language program. This section discusses a relocatable Z-80 assembly language example program which may easily be modified to meet your specific requirements.

EXAMPLE 9

Assume you want to program and verify a 2716 with source code located in system memory at 1000H - 13FFH. This represents 1K-bytes of data, and you decide to load the data in the upper half of a 2716 residing in socket ROM15 on a 32K BYTESAVER assigned to the upper 32K of memory. Thus, the 2716 resides at F800H - FFFFH, and the source code will program the upper half of this region, or FC00H - FFFFH. The program is written as a subroutine which is called with the source code starting address in HL, the number of source code bytes in BC and the EPROM starting address in DE. The subroutine returns with (A) = 00H indicating successful verification; (A) = FFH if not.

```

;
;           MAIN PROGRAM
;
;   This sample main program CALLs subroutine
;   PGM2716 which does the 2716 PROM programming.
;
      (1000) SOURCE: EQU      1000H      ;SOURCE CODE START ADDR
      (03FF) SWATH: EQU      3FFH       ;LENGTH OF SOURCE CODE
      (FC00) EPROM: EQU      0FC00H     ;EPROM PROGRAMMING START
0000     210010 MAIN: LD      HL,SOURCE ;LOAD SOURCE START ADDR
0003     01FF03          LD      BC,SWATH ;LOAD SOURCE LENGTH
0006     1100FC          LD      DE,EPROM ;LOAD PROGRAM START ADDR
0009     CD0D00          CALL    PGM2716 ;PROGRAM THE PROM
000C     76              HALT         ;END OF SAMPLE MAIN

```



```

;
;           SUBROUTINE PGM2716
;
; This subroutine programs 2716 EPROMs occupying PROGRAM
; ENABLED sockets on a Cromemco 32K BYTESAVER memory board.
; Switch the PROGRAM POWER switch ON before running program,
; turn it OFF after execution.
;   CALL subroutine with:   (HL)=source code starting address
;                           (BC)=source code length
;                           (DE)=EPROM programming start address
; RETURNS with:           (A)=00H if successful verification
;                           (A)=FFH if any byte does not verify
;                           (BC), (DE) and (HL) changed
;
000D    C5      PGM2716: PUSH    BC           ;SAVE SWATH ON STACK
000E    EDB0           LDIR           ;BLOCK MOVE SOURCE TO EPROM
0010    C1           POP     BC           ;RESTORE SWATH TO BC
0011    2B           DEC     HL           ;ADDR LAST SOURCE BYTE
0012    1B      VERIFY: DEC     DE           ;ADDR LAST EPROM BYTE
0013    1A           LD      A,(DE)       ;EPROM BYTE TO ACC.
0014    EDA9           CPD            ;COMPARE (A) TO (HL)
0016    E21E00        JP      PO,EXIT     ;THRU VERIFY IF (BC)=00H
0019    28F7           JR      Z,VERIFY   ;NOT THRU: NEXT BYTE
001B    3EFF      ERROR: LD      A,-1      ;ERROR: LOAD (A) WITH
001D    C9           RET                ;0FFH AND RETURN
001E    3E00      EXIT:  LD      A,0      ;SUCCESS: LOAD (A) WITH
0020    C9           RET                ;00H AND RETURN

```

The sample program consists of a main segment which CALLS the EPROM programming subroutine. Your main program must define register values (BC), (DE) and (HL) before calling PGM2716. For simplicity, the main and subroutine programs are assembled with a 0000H starting address. The code may be re-located anywhere else in memory provided the absolute jump instruction "JP PO,EXIT" is re-assembled to point to the new EXIT point. It is the user's responsibility to test Reg. A for a successful verification after the RETURN and take the appropriate action.

Section 4

THEORY OF OPERATION

This section gives a summary discussion of the 32K BYTESAVER at the component level. The user may find the discussion useful for troubleshooting the board, or just for gaining a fuller understanding of the board's features. The discussion is functionally divided into five categories: power supplies, addressing, memory read cycles, memory write cycles and DMA cycles.

4.1 POWER SUPPLIES

There are three major power supply lines on the 32K BYTESAVER; two regulated +5 volt lines and a Zener regulated +26 volt line.

The +5 volt lines are derived from the S-100 bus unregulated +8 volt line with two 7805 IC voltage regulators. Each +5 volt line drives approximately one-half of the board loads. One of the regulated +5 volt lines is dc to dc converted to +26 volts by relaxation oscillator Q1, Q2, Q3, T1 and associated circuitry. The converter is turned ON and OFF by the PROGRAM POWER switch, which also controls the red LED indicator. The +26 volt line is used to supply the 2716 programming voltage.

If +26 volts is not present across C3 when the PROGRAM POWER switch is ON, check the collector of Q1 for a 0 volt/+17 volt square wave running at between 100 and 200 KHz. If no switching voltage is present, check Q1. If Q1 is good, then check the other converter components Q2, Q3, D5 (a 25 volt 3% Zener diode), D6 and T1 for opens or shorts.

#### 4.2 ADDRESSING

High order address line A15 is compared to switch A15 in the ADDR/CONTROL group at [IC21 pins 9 and 10]. The comparison output is logically ANDed with S-100 bus signals and board signal [IC16 pin 8] to yield the important node signal BOARD ENABLE which logically equals the following Boolean expression:

$$\text{BOARD ENABLE} = (\text{A15} = \text{switch A15}) \text{ AND } (\overline{\text{sINTA}}) \text{ AND } (\overline{\text{sOUT}}) \text{ AND } (\overline{\text{sINP}}) \text{ AND } (\overline{\text{MEMR DISABLE}}) \text{ AND } (\overline{\text{IC16 PIN 8}}) \text{ AND } (\overline{\text{SWO OR PROGRAM POWER}}) \text{ AND } (\overline{\text{THE ADDRESSED ROM NOT SHADOWED}}).$$

The board will be enabled when the logic expression above is true, or evaluates to logic 1. The line [IC16 pin 8] will be low when the board is in an active memory bank, or has DMA IN during a DMA cycle.

Address lines A11 - A14 feed two one-of-eight decoders IC6 and IC7 which generate chip select signals for each of

the sixteen ROM sockets. Buffered address lines A0 - A9 parallel feed all 2716s to finally select the byte-on-chip.

Negative true chip select signals from IC6 and IC7 are logically ANDed by pairs (IC5 and IC8), and the SHADOW ROM switches parallel connect all of these ANDed outputs to node BOARD ENABLE. If either of the socket pair is chip selected with a low level from IC6 or IC7, the AND output will go low; and if the corresponding SHADOW ROM switch is closed (ON), BOARD ENABLE will go low thus disabling the board.

The board is mapped into an active or inactive memory bank by outputting a control word to output port 40H. Port address 40H is decoded from address lines A0 - A7 by the wire-ORed 7405's IC13 and IC19. These outputs are logically ANDed with control signals sOUT,  $\overline{\text{pWR}}$  and switch BANK ENABLE to yield node signal BANK SELECT ENABLE = (sOUT) AND ( $\overline{\text{pWR}}$ ) AND (BANK ENABLE) AND (A0 - A7 = 40H). A high BANK SELECT ENABLE strobes the D-type flip flop [IC15 pin 3] (it is also strobed on a Power-On Clear and a system RESET), and if the D-input at [IC15 pin 2] is low, the board is mapped into an active memory bank and the green LED indicator lights. If the D-input is high when strobed, the board is mapped into an inactive memory bank and the LED goes out.

The bits of the control byte output to port 40H are inverted by IC14 and IC18, and those bits selected with the

BANK SELECT switches are logically ANDed to drive the D-input of [IC15 pin 2]. Any logic 1 control bit output to port 40H which is switch connected to [IC14 pin 11] will then place the board in an active memory bank.

#### 4.3 MEMORY READ CYCLES

The CPU begins a memory read cycle by placing the memory address on the S-100 address bus A0 - A15 and by asserting the SMEMR control line high. After sampling the pRDY line to ascertain whether the memory is ready to supply the data byte, the CPU strobes the data from the data in bus DI0 - DI7 with a momentary high transition on the pDBIN line if memory is ready, or the CPU re-samples the pRDY line one clock cycle later if memory is not ready.

If no wait states are selected with the WAIT switch in the ADDR/ CONTROL group, the pRDY line to the CPU stays high; if one wait state is selected with the switch, flip flop output [IC15 pin 8] goes low during pSYNC at the beginning of the read cycle (forcing pRDY = low), then goes high again one machine cycle later when pSYNC = low.

When the CPU asserts pDBIN high to strobe the read data from the DI bus, this signal is logically ANDed with BOARD ENABLE to yield signal READ ENABLE = (pDBIN) AND (BOARD ENABLE). READ ENABLE in turn enables tri-state drivers IC27

which place the 2716 data byte onto the DI bus.

#### 4.4 MEMORY WRITE CYCLES

The CPU begins a memory write cycle to the 32K BYTESAVER by first placing the byte address on the S-100 address bus A0 - A15 and asserting control signals  $\overline{sW0}$  low and MEM WRITE high. The CPU then places the data byte on the data out bus D00 - D07, asserts the  $\overline{pWR}$  line low and samples the pRDY line. If the pRDY line is low indicating memory has not latched the data byte, the CPU waits an integral number of clock cycles with stable address, data and control signals until the pRDY line again goes high. After the pRDY line is sampled high, program execution resumes.

The coincidence of MEM WRITE and BOARD ENABLE clears PROGRAM PULSE counters IC28 and IC29. These dual 4-bit counters are then driven by the 2 MHz CLOCK line, and they, along with one-of-ten decoder IC30 and D-type flip flop IC31 generate a digitally counted 50 msec PROGRAM PULSE which feeds the 2716 PROG input during programming operations. The coincidence of MEM WRITE and BOARD ENABLE also enables tri-state drivers IC20 placing the contents of the data out bus D00 - D07 at the 2716 output pins D0 - D7. While the data byte drives the 2716 data lines and the 50 msec PROGRAM PULSE is high, the pRDY line is held low (thus forcing the CPU to wait) until the byte is completely programmed. During this

time, the PROGRAM POWER switch should be ON forcing +26 volts at the 2716 V(pp) input rather than the memory read +5 volt level.

The PROGRAM ENABLE switches selectively enable or disable the PROGRAM PULSE to each ROM socket. A closed (ON) switch enables the PROGRAM PULSE to its socket; an open (OFF) switch holds the 2716 PROG input at logic 0 thus inhibiting PROM programming.

#### 4.5 DMA CYCLES

The CPU acknowledges a DMA request by asserting the pHLDA line high. The system address, data out and control busses are then tri-stated allowing the DMA device to control these lines. The pHLDA signal to the 32K BYTESAVER is logically gated with switch settings DMA OVERRIDE ENABLE/DISABLE and DMA IN/OUT resulting in signal [IC16 pin 8] which ultimately either enables or disables the board thru node signal BOARD ENABLE. To enable the board, [IC16 pin 8] must be low. The table below shows the relationship among the DMA switches and signal [IC16 pin 8]. In the table, "Q" is output [IC15 pin 5] which is low when the board exists in an active memory bank (when the green LED is lit).

TABLE 2

DMA OVERRIDE	DMA IN/OUT	IC16 PIN 8 (A LOW ENABLES THE BOARD)
DISABLED (OFF)	OUT (ON)	Q
DISABLED (OFF)	IN (OFF)	Q
ENABLED (ON)	OUT (ON)	(pHLDA) OR (Q)
ENABLED (ON)	IN (OFF)	(pHLDA) AND (Q)

From the table it is seen that when DMA OVERRIDE is disabled, the board must be placed in an active memory bank to be accessible, and the DMA IN/OUT switch setting is irrelevant. When DMA OVERRIDE is ENABLED, a DMA cycle will disable the board if DMA is OUT (since pHLDA is high); a DMA cycle will enable the board if DMA is IN regardless of the Q output state (since  $\overline{\text{pHLDA}}$  is low). Thus, with DMA OVERRIDE enabled, a board with DMA OUT disappears during DMA transfers, and a board with DMA IN is available across memory bank boundaries.



Section 5

ASSEMBLY INSTRUCTIONS

If you purchased a 32K BYTESAVER kit, you will find assembly to be straight-forward provided you follow the instructions below.

Before beginning assembly, verify you have all kit parts by referring to the Parts List at the end of this manual. Please fill out and return the Missing Parts form to your authorized Cromemco dealer if any parts are damaged or missing.

5.1 ASSEMBLY STEPS

All parts are inserted from the component side of the board (with the white printed legend), and all soldering is done from the opposite side.

Be sure to use a high quality rosin core solder (DO NOT USE ACID CORE SOLDER), and a fine tipped low-wattage (25 W or less) soldering iron.

The printed legend on the component side of the board shows the exact location and orientation of each component. Check off each instruction when completed.

( ) Solder in position all 1/4-watt resistors:

R1	180	BROWN-GREY-BROWN
R2	180	BROWN-GREY-BROWN
R3	47	YELLOW-VIOLET-BLACK
R4	100K	BROWN-BLACK-YELLOW
R5	1K	BROWN-BLACK-RED
R6	33K	ORANGE-ORANGE-ORANGE
R7	560	GREEN-BLUE-BROWN
R8	560	GREEN-BLUE-BROWN
R9	1K	BROWN-BLACK-RED
R10	1K	BROWN-BLACK-RED
R11	560	GREEN-BLUE-BROWN
R12	560	GREEN-BLUE-BROWN

( ) Solder in position forty-seven IC sockets.

( ) Solder five SIP resistor networks RN1-RN5 in place. The arrow tips printed on the circuit board point to SIP pin 1; align arrows with numerals "1" printed on SIP packages.

( ) Install polarized capacitors C3, C9, C10 and C24. C3 has a 50V rating; the others a 20V rating. WHEN INSTALLING THE 10 uF POLARIZED CAPACITORS MAKE CERTAIN THAT THE "+" END OF THE CAPACITOR IS ALIGNED WITH THE "+" PRINTED ON THE P.C. BOARD.

( ) Install the remaining twenty-seven capacitors. Note that C4's value is .01 uF, not .1 uF. Also note that on some boards, capacitor values C27 and C28 are mis-labeled; see application note at the end of this manual.

( ) Install transformer T1.

## 32K BYTESAVER INSTRUCTION MANUAL

( ) Install transistors Q1, Q2 and Q3. Align transistor flat sides with outline drawings. Note that on some boards, the outline drawing for Q1 mis-represents the transistor outline; see application note at the end of this manual.

( ) Install light emitting diodes D1 and D2. Align LED flat sides with outline drawings. Bend the LED leads at right angles so that the LEDs face the top of the board when installed.

( ) Install three 1N4148 diodes D3, D4 and D6. Align the bands on the diodes with the bands on the outline drawings. Install 1N5253 Zener diode D5 in the same way.

( ) Install five 8-pole DIP switches. The arrows on the switch packages indicating the ON position should point towards the top of the board.

( ) Install the SPDT toggle switch.

( ) Install the heatsink and voltage regulators IC32 and IC33. Position each regulator so the exposed metallic side makes contact with the heat sink. Make sure the regulator legs do NOT make contact with the metallic heat sink.

( ) Install all ICs in their correct sockets (see Important Note below). The arrow tips printed on the circuit board

point to IC pin 1 (see Figure 13). Also note that on some boards, IC23 is mis-labeled; see application note at the end of this manual.

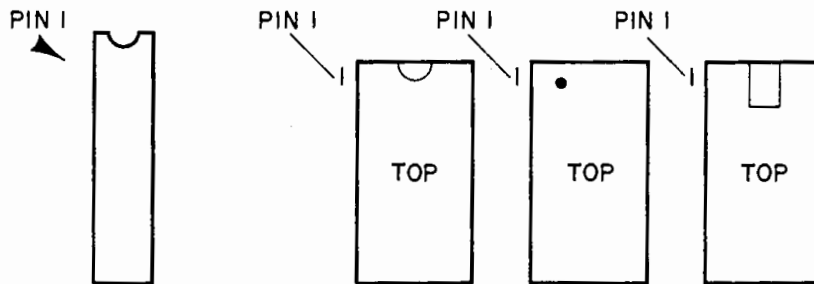


Figure 13: IC PIN 1 POSITION

IMPORTANT NOTE

The most common assembly faults are bent under IC legs. To avoid this problem, first bend the IC legs to closely match the IC socket span. Then "rock" the IC into its socket with a gentle end-to-end pressure. Visually inspect the legs after insertion by looking beneath the device.

This completes the construction of the Cromemco 32K BYTESAVER board. Carefully inspect your work before proceeding. Take particular care to see that there are no inadvertent solder bridges between pads and/or adjacent foil areas. It is a good practice to scrub the solder side of the

board clean with a fluorocarbon solution to remove any fine metallic particles which may be imbedded in the rosin residue.

## 5.2 POWER LINE TESTING

Follow the next procedure to verify that no short circuits exist between the board power lines, or between the power lines and ground. If testing indicates a short circuit, the connection must be found and removed.

With the board disconnected from the S-100 bus, connect an ohmmeter, on the R x 1 or lowest full-scale setting, across C24. The exact resistance reading is not important (it depends heavily on the ohmmeter design), but it should be several ohms or greater. If the reading indicates zero or a fraction of an ohm, a short exists between the +8 volt line and ground. Remove the short and verify by re-testing. Reverse the ohmmeter leads and again verify a non-zero ohm condition.

Test one +5 volt line as above by placing the ohmmeter leads across C9. Test the other +5 volt line across C10. Test the +26 volt line across C3. In each case, observe a non-zero ohm condition.

Now, place the ohmmeter between the "+" end of polarized

## 32K BYTESAVER INSTRUCTION MANUAL

capacitor C24 and the "+" end of C9; repeat with C24 and C10; repeat with C9 and C10. Verify a non-zero ohm condition in each case.

Follow the next procedure to verify that proper power supply voltages are present when the 32K BYTESAVER is plugged into an S-100 bus slot.

First, turn the system power OFF. Install the 32K BYTESAVER in an S-100 slot which permits access to the component side of the board with a VOM. NEVER insert or remove the board or board parts with the system power ON.

Turn the system power ON. The 32K BYTESAVER switch settings are unimportant for this test. Carefully measure the voltage across C9 and verify 5.0 volts with the same polarity as capacitor C9. Carefully measure the voltage across C10 and verify 5.0 volts with C10's polarity. Turn the PROGRAM POWER switch ON and carefully measure the voltage across C3. Verify between 25.0 and 26.5 volts with C3's polarity.

This completes the preliminary testing of the 32K BYTESAVER board.

32K BYTESAVER PARTS LIST

<u>Integrated Circuits Part No.</u>			<u>Resistors</u>		<u>Part No.</u>
IC1	74LS02	010-0068	R1	180	001-0009
IC2	74LS32	010-0058	R2	180	001-0009
IC3	74LS02	010-0068	R3	47	001-0003
IC4	74LS32	010-0058	R4	100K	001-0039
IC5	74LS09	010-0110	R5	1K	001-0018
IC6	74LS42	010-0057	R6	33K	001-0035
IC7	74LS42	010-0057	R7	560	001-0015
IC8	74LS09	010-0110	R8	560	001-0015
IC9	74LS32	010-0058	R9	1K	001-0018
IC10	74LS02	010-0068	R10	1K	001-0018
IC11	74LS32	010-0058	R11	560	001-0015
IC12	74LS02	010-0068	R12	560	001-0015
IC13	74LS05	010-0065	<u>Capacitors</u>		
IC14	74LS05	010-0065	C1	.1 uF	004-0030
IC15	74LS74	010-0055	C2	680 pF	004-0020
IC16	74LS00	010-0069	C3	10 uF @ 50V	004-0031
IC17	74LS33	010-0099	C4	.01 uF	004-0026
IC18	74LS05	010-0065	C5-C8	.1 uF	004-0030
IC19	74LS05	010-0065	C9	10 uF @ 20V	004-0030
IC20	74LS244	010-0100	C10	10 uF @ 20V	004-0030
IC21	74LS86	010-0052	C11	220 pF	004-0013
IC22	74LS33	010-0099	C12	47 pF	004-0005
IC23	74LS00	010-0069	C13-C21	.1 uF	004-0030
IC24	74LS04	010-0066	C22	47 pF	004-0005
IC25	74LS244	010-0100	C23	220 pF	004-0013
IC26	74367	010-0080	C24	10 uF @ 20V	004-0032
IC27	74LS244	010-0100	C25	.1 uF	004-0030
IC28	74393	010-0078	C26	47 pF	004-0005
IC29	74393	010-0078	C27	.1 uF	004-0030
IC30	74LS42	010-0057	C28	47 pF	004-0005
IC31	74LS74	010-0055	C29-C31	.1 uF	004-0030
IC32	7805	012-0001	<u>Diodes/Transistors</u>		
IC33	7805	012-0001	Q1	2N4013	009-0003
<u>Resistor Networks</u>			Q2	2N3906	009-0002
RN1	4.7K, 10 PIN	003-0014	Q3	2N3904	009-0001
RN2	1K, 8 PIN	003-0007	D1	TIL-209, RED	008-0019
RN3	1K, 8 PIN	003-0007	D2	TIL-211, GRN	008-0020
RN4	4.7K, 10 PIN	003-0014	D3-4	1N4148	008-0002
RN5	1K, 8 PIN	003-0007	D5	1N5253 (3%)	008-0010
			D6	1N4148	008-0002

32K BYTESAVER PARTS LIST

<u>Miscellaneous</u>	<u>Part No.</u>	<u>Documentation</u>	<u>Part No.</u>
PC BOARD	020-0004		
T1 XT8K XFMR	014-0001	32K BYTESAVER	
1-SPDT TOGGLE SW.	013-0000	INSTRUCTION	
5-8 POLE DIP SW.	013-0002	MANUAL	023-0002
1-HEATSINK	021-0017		
4-SCREWS 6-32	015-0006		
4-NUTS 6-32	015-0013		
1-P.C. BOARD	020-0004		
24-IC SOCKETS			
14 PIN	017-0001		
4-IC SOCKETS			
16 PIN	017-0002		
3-IC SOCKETS			
20 PIN	017-0004		
16-IC SOCKETS			
24 PIN	017-0005		





APPLICATION NOTE

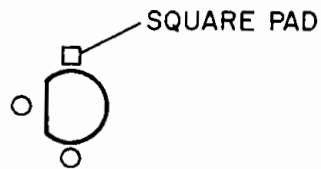
On some 32K BYTESAVER boards, there are four errors in the silk screened component legend. They are:

a) C27's value is labeled "47"; it should read ".1" .

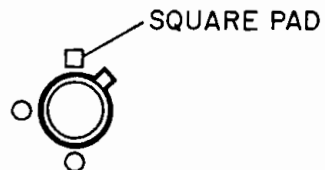
b) C28's value is labeled ".1"; it should read "47".

c) IC22 is labeled "74LS00"; it should be labeled "74LS33".

d) Q1's outline is shown as;



it should appear as:



WARRANTY

Your factory built 32K 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 they are returned to Cromemco. No other warranty is expressed or implied. We are not liable for consequential damages.

Should your factory built 32K BYTESAVER fail after the warranty period, it will be repaired for a fixed fee, provided it is returned to Cromemco. We reserve the right to refuse to repair any product which has, in our opinion, been subjected to abnormal electrical or mechanical abuse. The service fee is currently \$70.00, and is subject to change without notice.

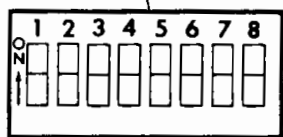
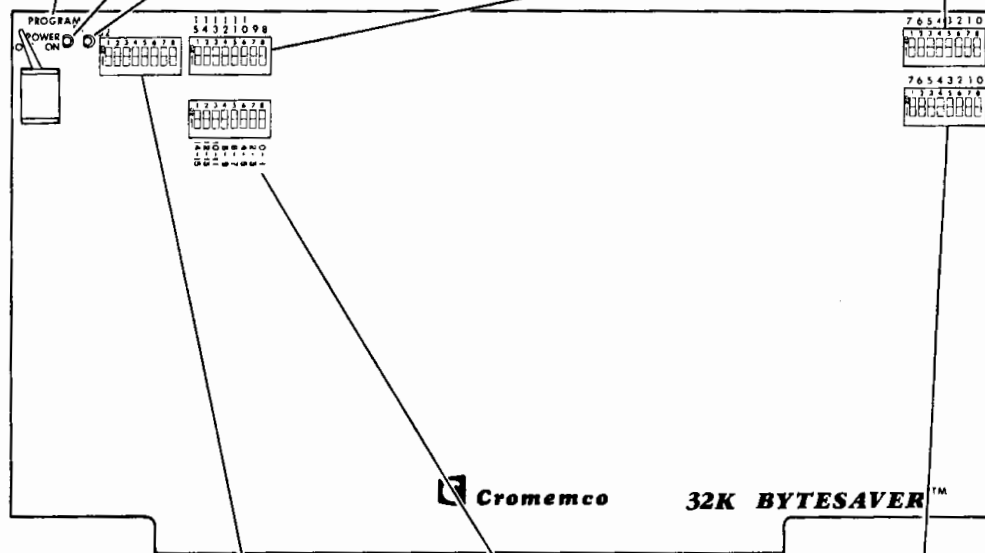
Your assembled 32K BYTESAVER kit will be repaired for a fixed fee, provided it is returned to Cromemco. We reserve the right to refuse repair of any kit which has not, in our opinion, been assembled in a workmanlike manner, or has been subjected to abnormal electrical or mechanical abuse. Payment of the service fee must accompany the returned merchandise. The service fee for kit repair is currently \$70.00, and is subject to change without notice.

PROGRAM POWER  
 ON = TURN ON +26V SUPPLY  
 OFF = TURN OFF +26V SUPPLY

PROGRAM POWER INDICATOR  
 ON = PROGRAM POWER ON  
 OFF = PROGRAM POWER OFF

BANK SELECT INDICATOR  
 ON = BOARD IN ACTIVE BANK  
 OFF = BOARD NOT IN ACTIVE BANK

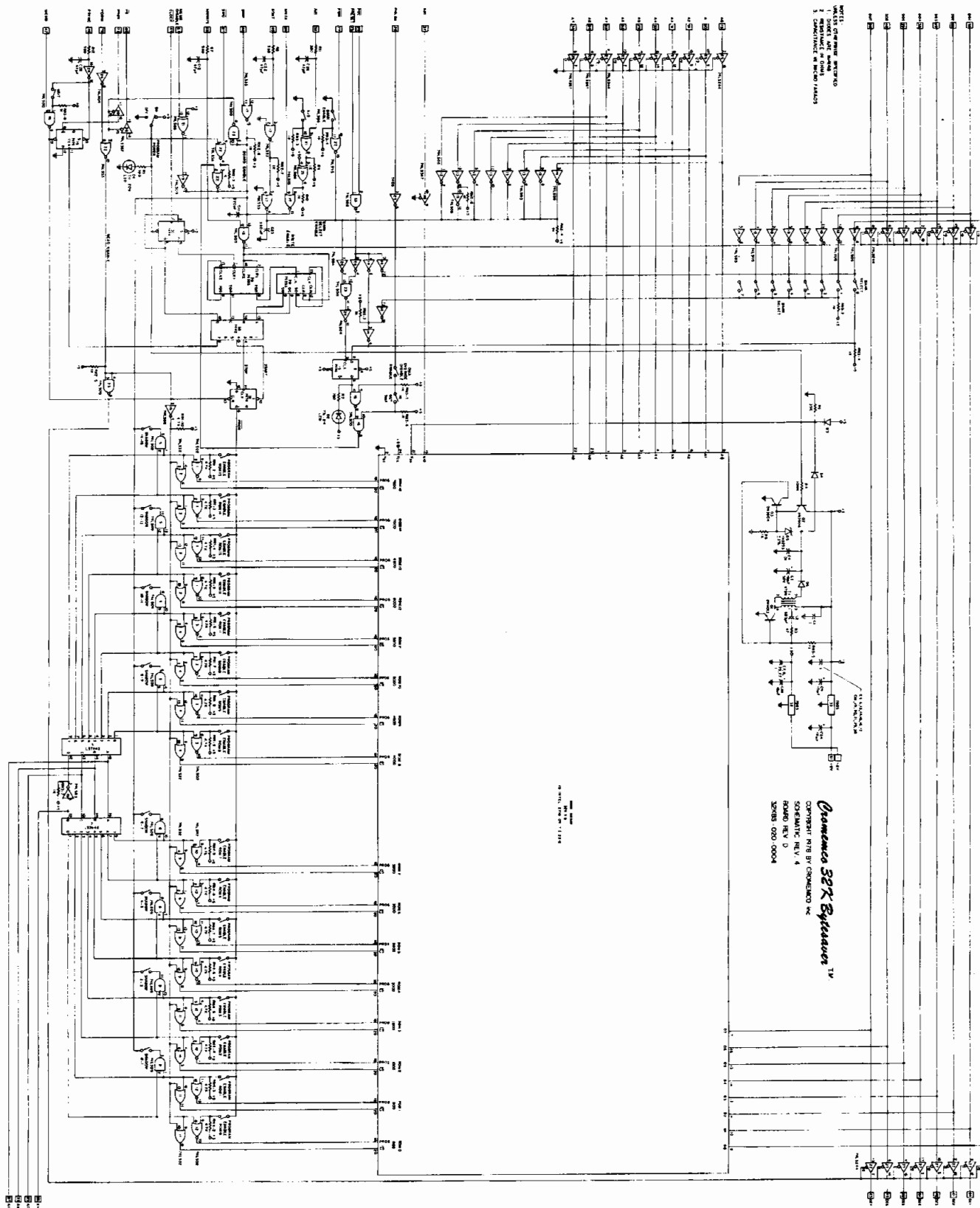
PROGRAM ENABLE ROM SOCKET  
 ON = ENABLE SOCKET PROGRAMMING  
 OFF = DISABLE SOCKET PROGRAMMING



BANK SELECT  
 ON = LOGICALLY PLACE BOARD IN BANK n  
 OFF = BOARD LOGICALLY NOT IN BANK n

SHADOW SOCKET PAIR  
 ON = REMOVE SOCKET PAIR FROM MEMORY MAP  
 OFF = SOCKET PAIR IN MEMORY MAP

<u>SWITCH NUMBER</u>	<u>FUNCTION</u>
1 - 3.	NOT USED
4.	BANK ENABLED/DISABLED ON = ENABLED OFF = DISABLED
5.	WAIT STATE ON = ONE WAIT STATE OFF = NO WAIT STATES
6.	A15 ON = (A15=1) OFF = (A15=0)
7.	DMA OVERRIDE ON = ENABLED OFF = DISABLED
8.	DMA IN/OUT ON = DMA OUT OFF = DMA IN



100111 GROUNDING SCHEMATIC  
 1 SOCKET UNIT MODEL 44  
 2 CHASSIS GROUNDING SCHEMATIC  
 3 COMPONENTS IN RED ON LABELS

**Zenith 32K System TV**  
 COPYRIGHT 1978 BY ZENITH INC.  
 SCHEMATIC REV. 4  
 READ REV. 0  
 ZENITH 000-0004

100111 GROUNDING SCHEMATIC  
 1 SOCKET UNIT MODEL 44  
 2 CHASSIS GROUNDING SCHEMATIC  
 3 COMPONENTS IN RED ON LABELS