Twenty Dollars

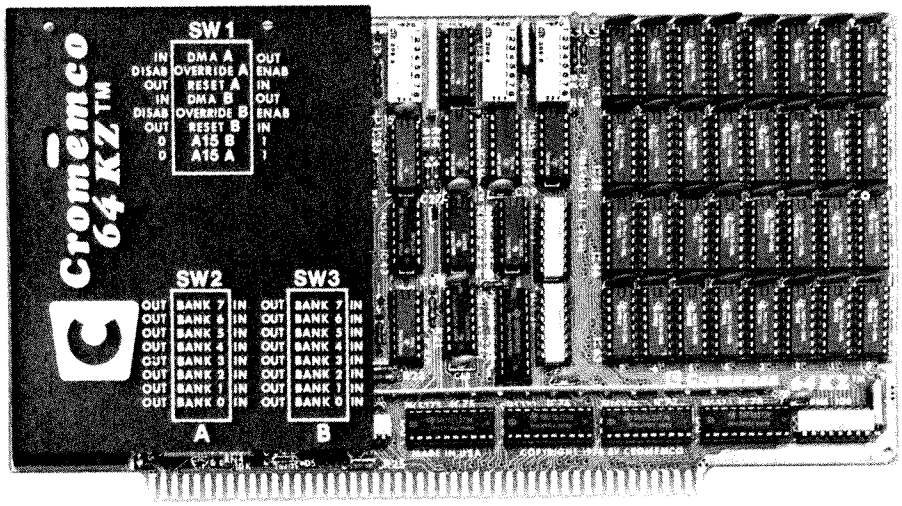
Cromemco

64KZ

Random Access Memory

Instruction Manual

64KZ
Random Access Memory



Part Number 023-0008

Copyright © 1979, 1981 By CROMEMCO, Inc.
All rights reserved.

Cromemco™
280 Bernardo Avenue
Mountain View, CA 94040

May 1981

This manual was produced on a Cromemco System Three computer utilizing a Cromemco HDD-22 Hard Disk Storage System running under the Cromemco Cromix ™ Operating System. The text was edited with the Cromemco Cromix Screen Editor.

The edited text (including imbedded typesetting codes) was transmitted over voice grade telephone lines. The transmission originated from a Cromemco system employing the Quadart and IOP boards utilizing the Cromemco Qmodem communications program. The transmission was received by a Shaffstall CEI interface and typeset on a Comp Edit phototypesetter.

Cromemco

Table of Contents

1. Introduction		Technical Specifications	1
2. Operating Instructions	2.1	Switch Options - An Overview	3
	2.2	Addressing the 64KZ	12
	2.3	Block and Bank Select	15
	2.4	Direct Memory Access	22
	2.5	Multi-User Basic Application	25
	2.6	64KZ Jumper Options	27
3. Theory of Operation		Theory of Operation	29
Appendix A		64KZ Parts List	35
Appendix B		64KZ Parts Location Diagram	37
Appendix C		Limited Warranty	39
		64KZ Schematic Diagram	41

Cromemco

Chapter 1 Introduction

The Cromemco 64KZ is an S-100 bus compatible 65,536 byte (64 Kbyte) read/write memory board. The 64KZ incorporates the TMS 4116-15 16K X 1 bit dynamic RAM chip to achieve its high memory density while maintaining a true 250 nsec (max) access time. This means the 64KZ reliably operates in 4 MHz Z80 systems with *absolutely no wait states.* The Cromemco 64KZ memory board offers the following outstanding and versatile features:

- 64 Kbytes of read/write memory on one S-100 memory board.
- A true access time of 250 nanoseconds (maximum).
- Z80 and 8080 CPU compatibility.
- Organization as two independent 32 Kbyte memory blocks: BLOCK A and BLOCK B.
- BANK SELECT allowing memory expansion beyond 64 Kbytes.
- Powerful DMA configuration options with DMA OVERRIDE.
- Automatic 64KZ enable or disable after a system RESET.
- All significant BLOCK A and BLOCK B options independently switch selectable.

Technical Specifications CROMEMCO 64KZ MEMORY BOARD

MEMORY CAPACITY:	65,536 BYTES (64 KBYTES)
MEMORY TYPE:	TMS 4116-15, 16K X 1 DYNAMIC RAM
MEMORY ACCESS TIME:	250 NANOSECONDS (MAXIMUM)
WAIT STATES @ 2 MHz:	NONE REQUIRED
WAIT STATES @ 4 MHz:	NONE REQUIRED
BUS COMPATIBILITY:	S-100
POWER REQUIREMENTS:	+ 8 VOLTS @ 1.8 AMPERES (MAXIMUM)
	+18 VOLTS @ .45 AMPERES (MAXIMUM)
	−18 VOLTS @ .03 AMPERES (MAXIMUM)
OPERATING ENVIRONMENT:	0 - 55 DEGREES CELSIUS

Chapter 2 Operating Instructions

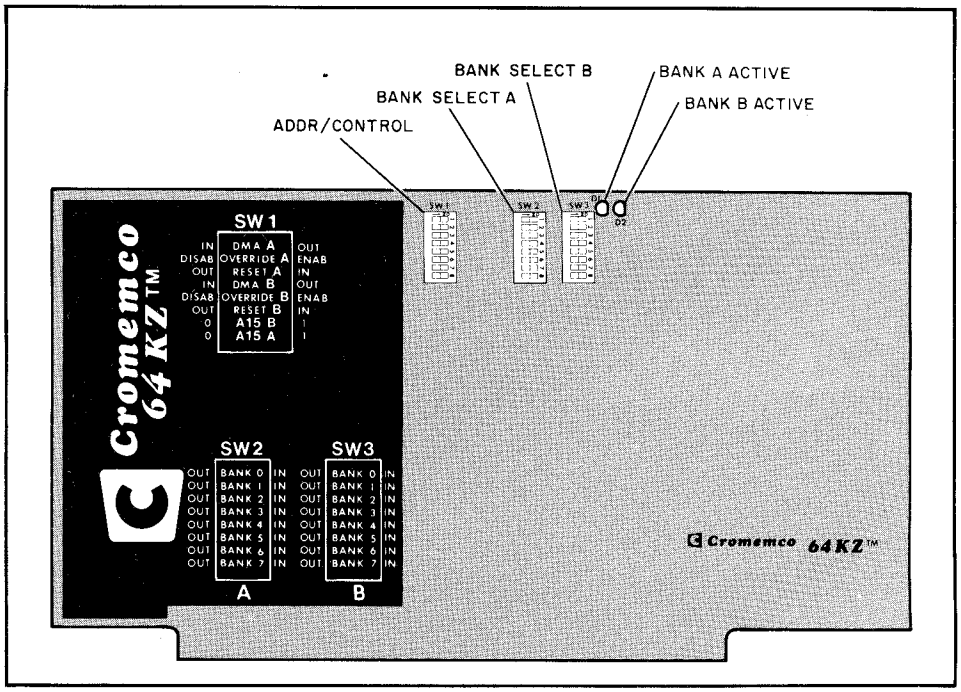
This chapter provides all of the information necessary to operate the 64KZ board in your system. The instructions begin with a 64KZ operational overview and summary of switch selectable options. The remainder of the instructions discusses each switch selectable 64KZ option from a system point of view. Careful reading of this chapter will enable you to custom fit the 64KZ to your specific system requirements.

2.1 Switch Options— An Overview

Before plugging the 64KZ into an empty S-100 bus slot, the board should be configured by setting three 64KZ switch groups. This section provides a brief description of each switch function, followed by three example configurations which typify the spectrum of 64KZ applications: a "stripped down" no options configuration; a 64KZ which shadows a ROM bootstrap loader program; and a 64KZ used in a Direct Memory Access DAZZLER® system.

The 64KZ board is functionally organized as two *independent* 32 Kbyte memory modules, referred to as **BLOCK A** and **BLOCK B.** BLOCK A and BLOCK B are configured by setting three switch groups located along the top edge of the board (see Figure 1). Quick reference switch legend artwork appears on the 64KZ heatsink assembly which summarizes each switch function. Those switch functions suffixed by the letter **A** pertain to memory BLOCK A; those suffixed with **B** to BLOCK B.

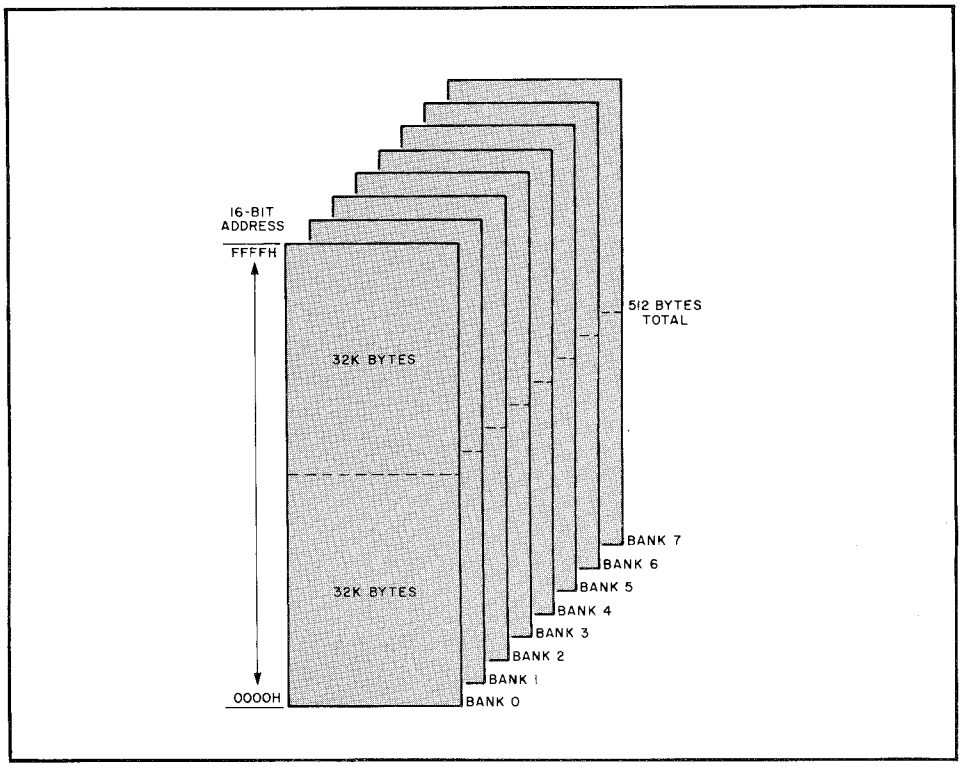
Figure 1
64KZ Switch Locations



Memory is further organized into eight 64 Kbyte memory **BANKs** (BANK 0 - BANK 7), allowing memory expansion up to 8 X 64 Kbytes = 512 Kbytes (see Figure 2). BLOCK A and BLOCK B are independently switch

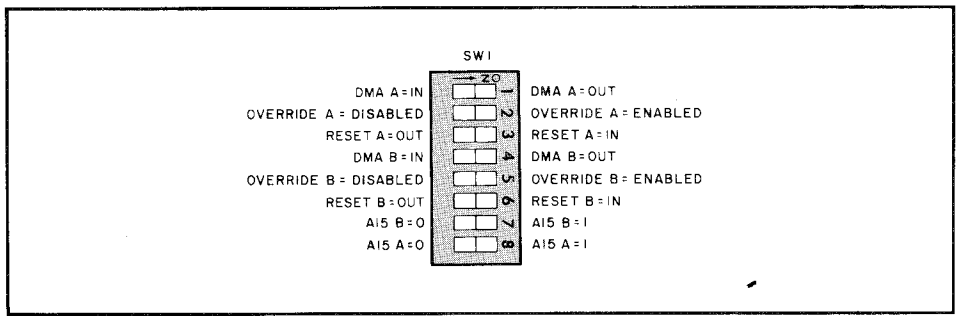
assigned to any combination of memory BANKs with the two BANK SELECT switch groups and memory BANKs are toggle activated/ deactivated under software control. Note that any switch in the three switch groups may safely be repositioned while power is applied to the 64KZ board.

Figure 2
64KZ Memory Banks



In the SW1 ADDR/CONTROL group, there are four switch functions which control BLOCK A/B: DMA IN/OUT; OVERRIDE ENABLE/ DISABLE; RESET IN/OUT and A15 0/1 (see Figure 3).

Figure 3
Addr/Control Switches



- The A15 Switches
  A15 = 0 maps BLOCK A/B into the lower half of the CPU's 64K direct addressing range (0000H - 7FFFH).

  A15 = 1 maps BLOCK A/B into the upper half of the CPU's 64K direct addressing range (8000H - FFFFH).

- The Reset Switches
  Setting RESET = OUT unconditionally disables BLOCK A/B after a system RESET or Power On Clear (P.O.C.), regardless of the BLOCK's current BANK active/ inactive status.

  Setting RESET = IN unconditionally enables BLOCK A/B after a system RESET or P.O.C., regardless of the BLOCK's current BANK active/ inactive status.

- The Override and DMA Switches
  The OVERRIDE and DMA switches control the 64KZ's Direct Memory Access response. Both of these switch settings are irrelevant for non-DMA transfers. In this case, BLOCK A/B is always accessible for read/write access when properly addressed (the correct 16-bit address **and** the correct memory BANK address).

  During DMA cycles, setting OVERRIDE = DISABLED again makes BLOCK A/B always available for read/write access when properly addressed (the correct 16-bit address **and** the correct memory BANK address). This mode may correctly be thought of as normal direct addressing (no DMA vectoring), and since OVERRIDE = DISABLED, switch setting DMA IN/OUT is irrelevant.

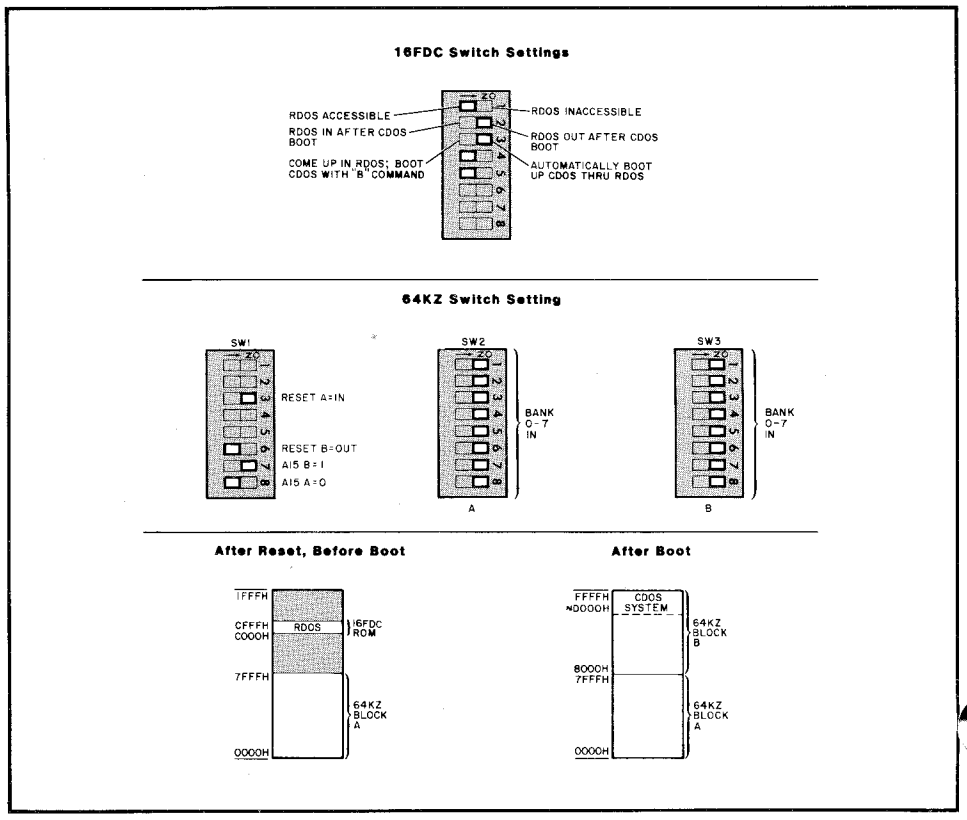
  Setting OVERRIDE = ENABLED effectively collapses all memory BANK boundaries during DMA (memory BANK boundaries are OVERRIDDEN) and board enabling or disabling is contingent upon whether DMA = IN or DMA = OUT. In this case, BLOCK A/B enables for DMA read/write access when 16-bit addressed if DMA = IN; and BLOCK A/B automatically disables when DMA = OUT, thereby allowing DMA vectoring to memory with DMA = IN.

- The Bank Select Switches and Indicators
  Switch groups SW2 and SW3 assign BLOCKs A and B to any combination of memory BANKs (none, one, several or all). Switch group SW2 controls BLOCK A and SW3 controls BLOCK B (see Figure 4).

Switch settings on the 16FDC board allow the user options of completely disabling or enabling RDOS-II, removing or retaining RDOS-II in the memory map after a CDOS boot, automatically *coming up* in RDOS-II (assuming a Cromemco 3102 terminal) and booting CDOS only when the direct command **B** (**B**oot) is issued, or automatically executing the RDOS-II boot routine. Note that if a terminal other than the Cromemco 3102 is used, several carriage **RETURN**s must be entered before RDOS-II will respond.

Assuming the 16FDC and 64KZ switch settings shown in Figure 6, CDOS automatically boots up after each system RESET or P.O.C. when the console RETURN key is depressed several times. The rationale behind the switch settings follows. First, BLOCK B, which is assigned to the upper 32K of memory, must disable after a RESET or P.O.C. to avoid a conflict with the shadowed RDOS-II ROM program located at C000H - CFFFH. This is accomplished by setting 64KZ switch RESET B = OUT. 16FDC switch number 2 must be positioned ON to disable RDOS-II after boot (RDOS-II is disabled at the same time memory BANK 0 is enabled). 16FDC switch number 3 is positioned ON to automatically start the RDOS-II boot routine after several RETURN key strokes, and finally, 64KZ switch BLOCK B, BANK 0 = IN to activate the upper 32 Kbytes of read/write memory as RDOS-II is disabled. The resulting memory map after CDOS has been loaded into 64KZ memory is shown in Figure 6.

Figure 6
Example 2 Boot CDOS Configuration



The next example illustrates the 64KZ **DMA OVERRIDE** feature used in conjunction with a Cromemco DAZZLER® color television interface.

Example 3

Assume in addition to the CDOS boot option discussed in EXAMPLE 2, you desire to operate the 64KZ in a CDOS system which contains a Cromemco DAZZLER graphics board set. The DAZZLER provides a general purpose interface between your computer and a color TV receiver. The DAZZLER uses high-speed Direct Memory Access (DMA) to read the *picture memory area* of the host computer, and translate the information into a color TV signal (for further details, reference Cromemco's DAZZLER Instruction Manual, part number 023-0003).

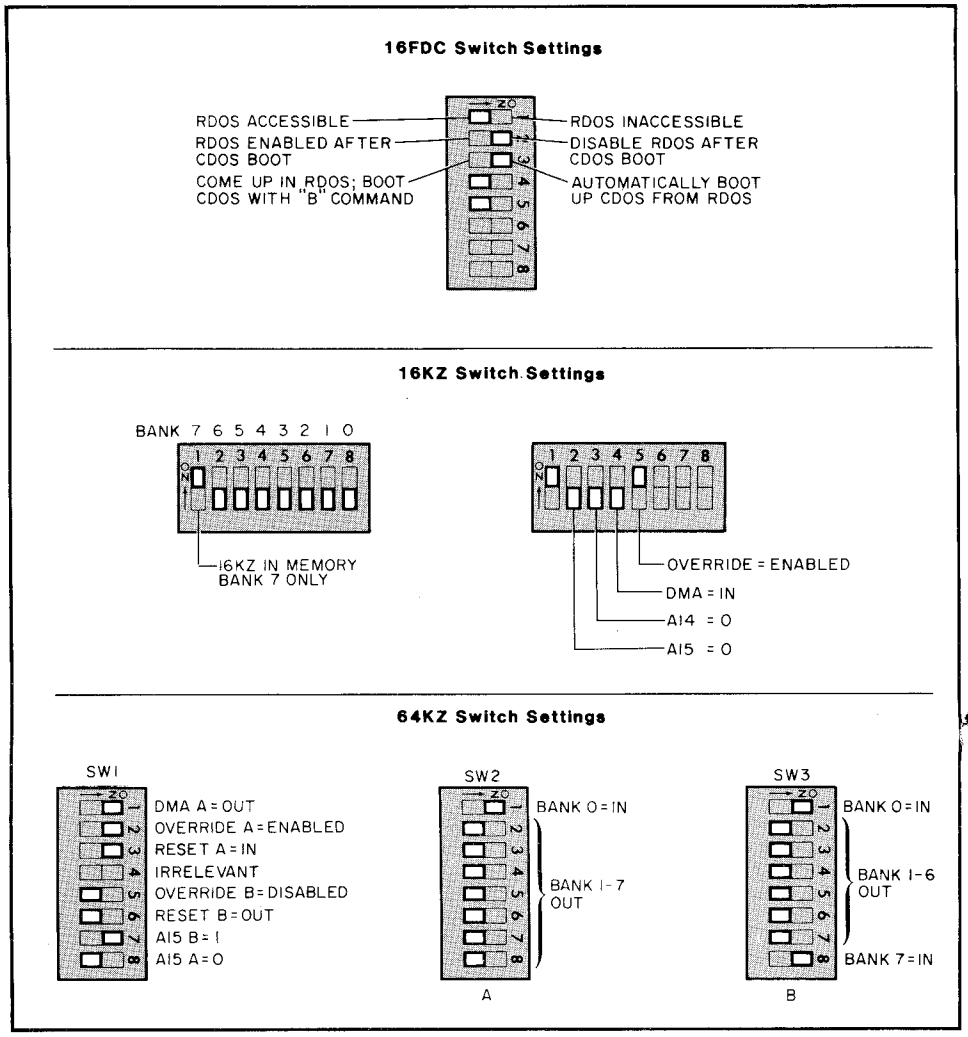
The DAZZLER can display either a 512 byte picture, or a X4 resolution 2 Kbyte picture. For the purpose of this example, assume you choose to use a Cromemco 16KZ read/write memory board to store the TV picture, a program located in BLOCK A of a 64KZ to perform the picture processing, and a program in BLOCK B of the same 64KZ to handle all I/O transfers between the picture processing program and picture memory. The 16KZ board, like the 64KZ, features memory BANK SELECT and DMA OVERRIDE. The 16KZ's memory capacity is 16 Kbytes, allowing eight separate X4 resolution 2 Kbyte pictures to be simultaneously stored on the board (this capacity might be useful in generating animated TV displays where one picture is displayed while the other seven are being processed for subsequent display).

With the 16FDC, 16KZ, and 64KZ switch settings illustrated in Figure 7, there are five active areas in the system memory map (see Figure 8).

RDOS-II ROM ON THE 16FDC BOARD: This 4 Kbyte memory module automatically boots CDOS from diskette (loads the CDOS program into roughly the uppermost 12 to 15 K of memory in BLOCK A, BANK 0) after a system RESET or a P.O.C. (assuming a Cromemco 3102 terminal) and disables after CDOS is loaded.

64KZ RAM BLOCK A: This 32 Kbyte memory module exists in memory BANK 0 only, spanning addresses 0000H - 7FFFH. The module enables after a system RESET or a P.O.C., and disables during all DMA cycles. Initially, RDOS-II loads the bootstrap program at starting address 0080H which then transfers the CDOS system program to the uppermost 12 to 15K of memory. Subsequently, BLOCK A would be loaded with a DAZZLER picture processing program from diskette.

Figure 7
Example 3 Switch Settings

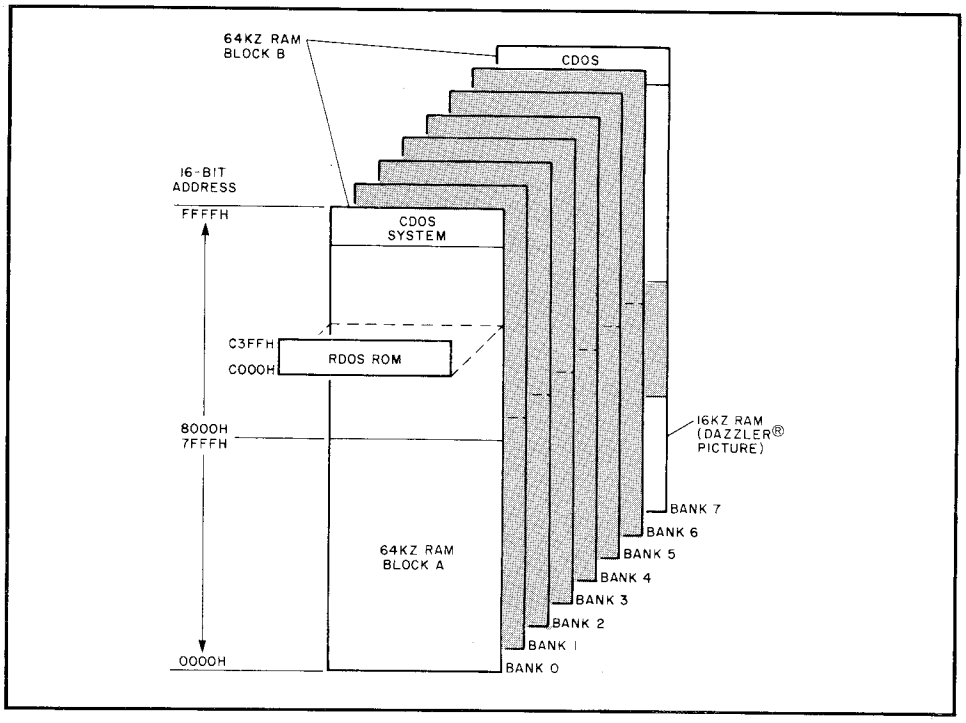


64KZ RAM BLOCK B: This 32 Kbyte memory module exists in memory BANK 0 **and** BANK 7 spanning addresses 8000H - FFFFH. The module disables after a system RESET or a P.O.C. to avoid a conflict with the shadowed RDOS-II ROM program. BLOCK B is activated in BANK 0 at the same time that RDOS-II is disabled, and its uppermost 12 to 15 K is then loaded with the CDOS system program. Subsequently, BLOCK B is loaded from diskette with an I/O driver program which links the DAZZLER picture memory with the picture processing program in BLOCK A. In this example, BLOCK B is available for DMA and non-DMA read/write access in both BANKs 0 and 7.

16KZ RAM (DAZZLER PICTURE): This 16 Kbyte memory module exists in memory BANK 7 only, spanning addresses 0000H - 3FFFH. The module automatically disables on a system RESET or P.O.C. (memory BANK 0 = OUT controls this function on the 16KZ board). Any DMA read/write operation to an address in the range 0000H - 3FFFH

automatically vectors to the 16KZ board since OVERRIDE = ENABLED and DMA = IN (and disables 64KZ BLOCK A since DMA A = OUT). The 16KZ is also available for non-DMA read/write access when correctly addressed in memory BANK 7.

Figure 8
Example 3 Memory Map



The overall system operation will sequence as follows (refer to Figure 9):

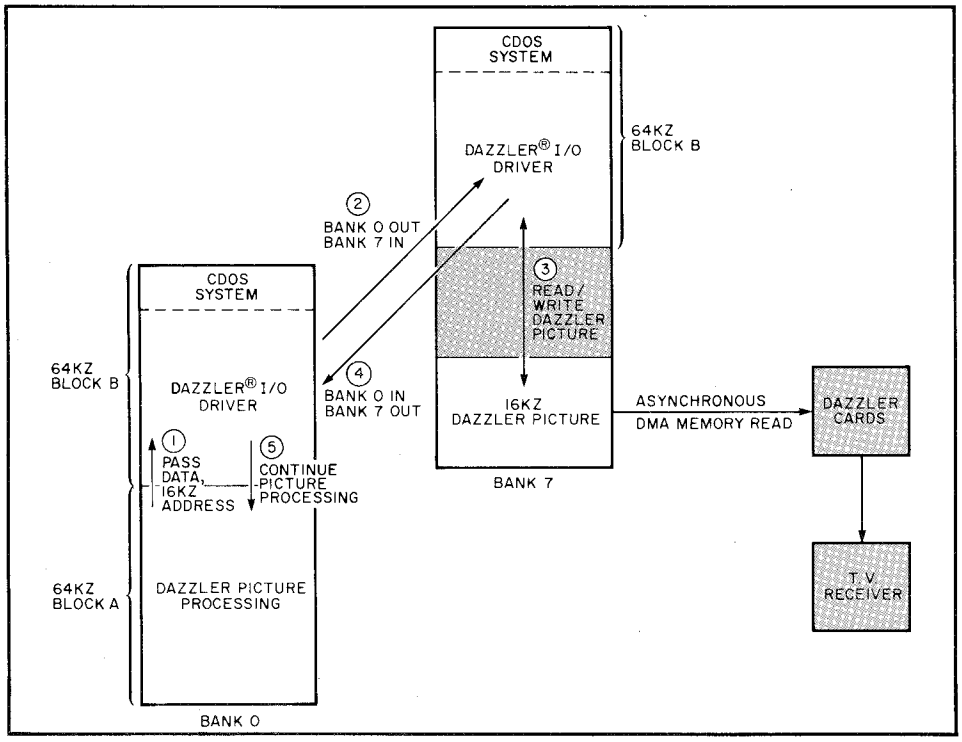
1. The picture processing program in BLOCK A, BANK 0 will generate a read/write request to the DAZZLER picture memory and pass the data and picture address to the I/O driver program in BLOCK B, BANK 0.

2. The I/O driver program will then switch memory BANK 7 IN and memory BANK 0 OUT by outputting the appropriate control word to output ports 40H on both the 64KZ and the 16KZ boards. This enables BLOCK B and the 16KZ picture memory, and disables BLOCK A. Notice that BLOCK B **must** be in BANK 7 to complete the data transfer; otherwise the CPU will lose contact with the I/O driver program in BLOCK B after the BANK switch.

3. The I/O driver program then transfers data to or from 16KZ DAZZLER memory.

4. After the data transfer, BANK 0 is switched back IN and BANK 7 is switched OUT. Notice that the I/O driver program in BLOCK B, BANKs 0 and 7 provides program continuity between BANKs 0 and 7 since BLOCK B is mapped into both BANKs.

Figure 9
Example 3 Dazzler Data Flow



5. The I/O driver program then passes control back to BLOCK A for further picture processing.

6. Asynchronously with the above sequential events (1-5), the DAZZLER interface cards periodically request and receive Direct Memory Access to the 16KZ memory board. All CPU processing is suspended until each DMA transfer is completed. During each DMA, the 16KZ board unconditionally enables for DAZZLER memory read cycles, and BLOCK A, BANK 0 unconditionally disables to preclude a data bus conflict. After the DMA transfer is completed, the CPU resumes processing from the point of interruption.

2.2 Addressing the 64KZ

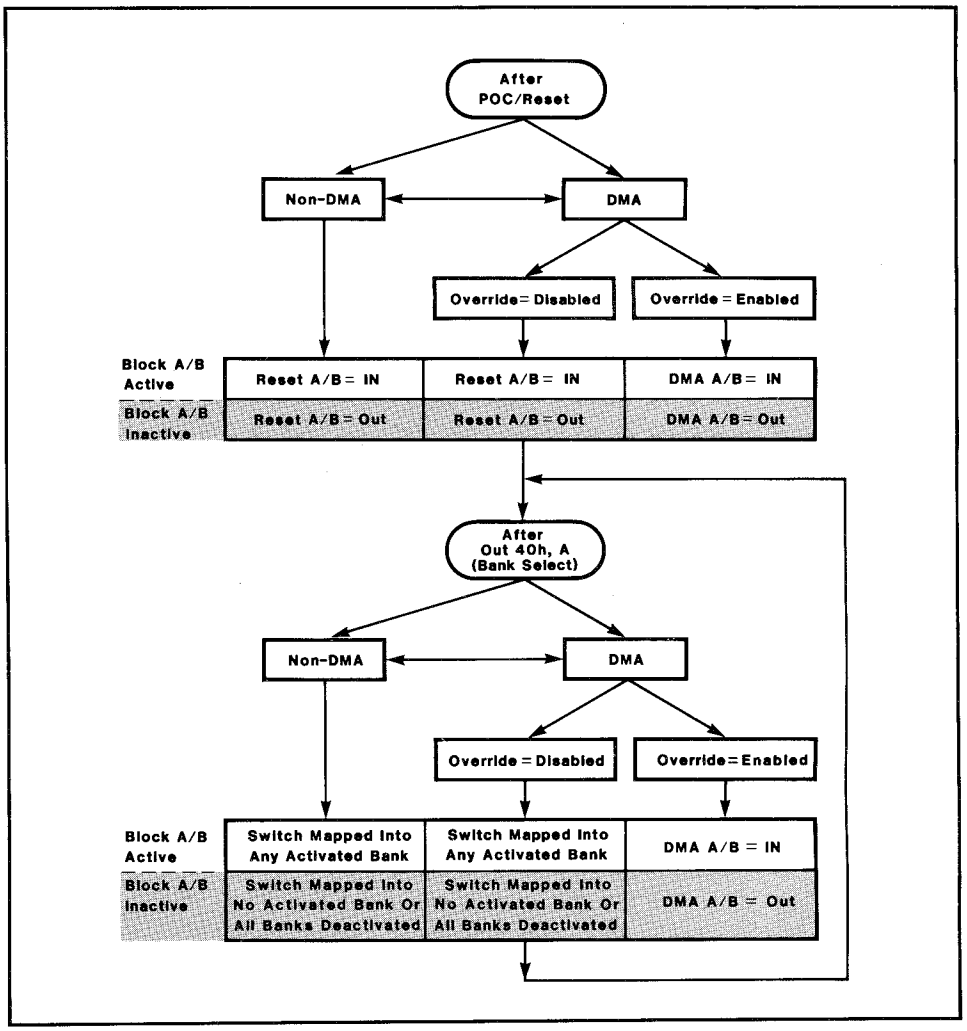
The 64KZ may properly be thought of as two independent 32 Kbyte RAM modules (BLOCK A & BLOCK B) on one memory board. To read/write a byte from either BLOCK, the memory byte must be addressed by driving S-100 bus lines A0 - A15; additionally the target memory BLOCK **must also be enabled or active.** An enabled memory BLOCK is *in* the memory map. A **disabled** or **inactive** memory BLOCK is *out* of the memory map; it does not read data from the S-100 DO bus, nor does it drive the S-100 DI bus lines.

There are three events which may change the active/inactive status of a 64KZ memory BLOCK:

- Power On Clear (P.O.C.) or RESET,
- executing an OUT 40H,A instruction which configures or reconfigures memory BANKS, or
- a DMA ←→ non-DMA transition.

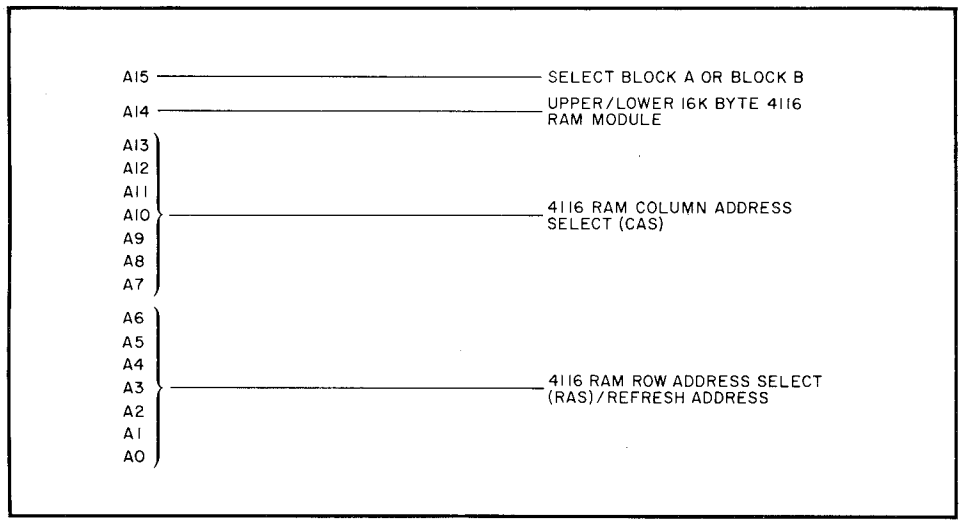
Figure 10 illustrates the relationship among these events. Subsequent manual sections will discuss Figure 10 in detail; for the present, notice the following major features. Immediately after a P.O.C. or RESET there are **no** memory BANKS. Memory BLOCK active/inactive status is determined only by the two RESET A/B switches for non-DMA access. Only during the execution of an OUT 40H,A instruction (which configures memory BANKs) are the memory BANK switch settings relevant.

Figure 10
64KZ Block Diagram
Enable/Disable



Also note that during DMA access to the 64KZ, setting an OVERRIDE A/B switch to DISABLED causes the 64KZ memory BLOCK to respond in the same way as it would to a non-DMA access; the first and second column entries in Figure 10 are identical. Finally note that during DMA access with the OVERRIDE A/B switches set to ENABLED, BLOCK A/B active/inactive is determined solely by the DMA A/B switches. Assuming now that both BLOCK A and BLOCK B are enabled for read/write access by one of the three events mentioned above, then memory BLOCK and 4116 RAM selection are decoded from the sixteen bit address A0 - A15 sent out by the CPU on the S-100 bus (see Figure 11). Since memory BLOCK A and B are 32 Kbyte modules, high order address line A15 is hardware compared to BLOCK A/B switch settings in 64KZ switch group SW1 to generate BLOCK SELECT.

Figure 11
64KZ Addressing



The 4116 RAM is a 16K X 1 bit chip with seven multiplexed address pins. Eight parallel addressed 4116s then form a 16 Kbyte module with each chip dedicated to one bit of memory data byte D0 - D7. Two sets of eight parallel addressed 4116s then form a 32 Kbyte memory BLOCK, and within the BLOCK, address line A14 is used internally to select the *upper* or *lower* 16 Kbyte module.

One of 16 Kbits on each 4116 are selected by address lines A0 - A13. Address lines A0 - A6 are applied first to the seven 4116 address lines to generate one of 128 Row Address Select (RAS) and shortly thereafter address lines A7 - A13 drive the same seven 4116 address lines to generate one of 128 Column Address Select (CAS).

Since the 4116 is a dynamic memory chip, it must be **refreshed** periodically to maintain data integrity. Refresh is accomplished by addressing each of the 128 row addresses every 2 milliseconds (max) and strobing the 4116 RAS input line; or alternately by performing a

memory read or write cycle which addresses each of the 128 row addresses at least once every 2 milliseconds.

The 64KZ board has been designed with the following division of responsibilities: the **user** switch assigns memory BANKs and memory BLOCKs, and executes software to enable or disable memory BANKs; **the 64KZ hardware** automatically generates BLOCK SELECT, 16 Kbyte module select, RAS and CAS from S-100 bus address lines A0 - A15, and also maintains 4116 refresh during all processor cycles except DMA (see Section 2.4).

2.3 64KZ Block and Bank Select

Memory BLOCK A and memory BLOCK B may be independently mapped into the upper or lower half of the CPU's 64K direct addressing range with the A15 switches in the SW1 switch group and may also be placed in any combination of memory BANKs with the BANK SELECT switch groups SW2 and SW3.

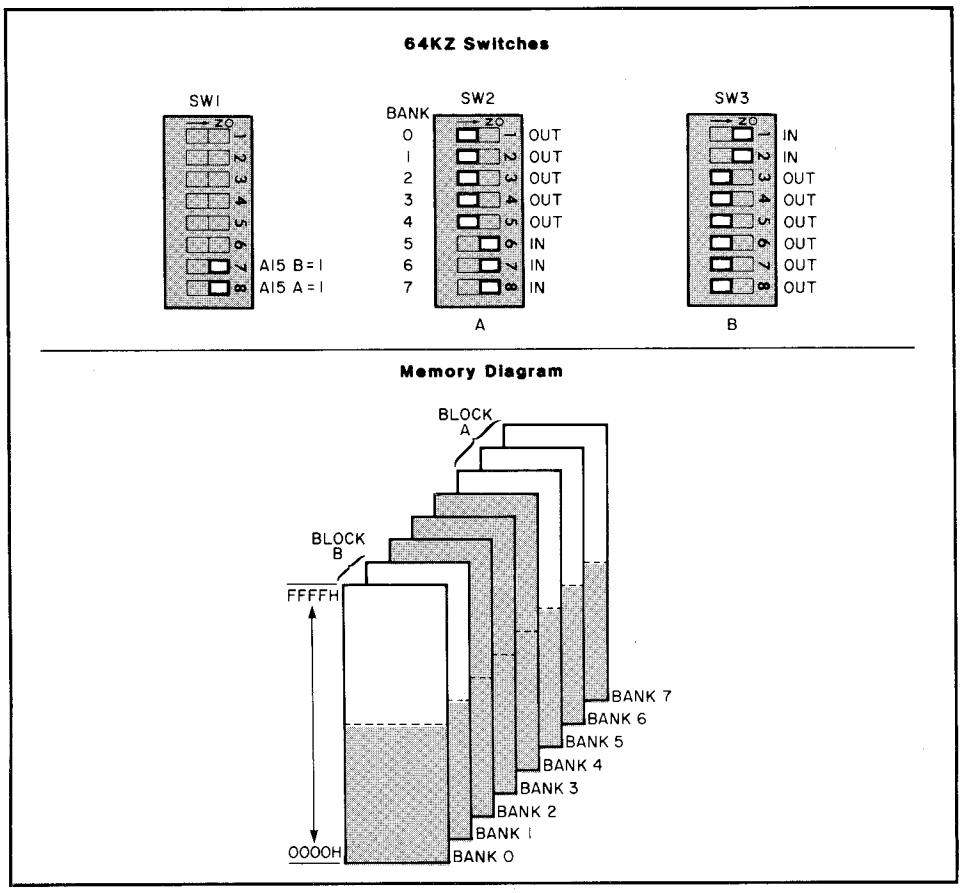
Other Cromemco memory boards which feature BANK SELECT are: the 4KZ RAM board; the 16KZ RAM board; the 8K BYTESAVER II 2708 PROM board; the 16KPR 2708 PROM board and the 32K BYTESAVER 2716 PROM board. Each of these boards may, in addition to the 64KZ, be memory mapped into a specific 16-bit address area and into any combination of memory BANKs with convenient switch settings. ***All*** boards respond to the same software generated BANK SELECT control word output by the system CPU.

Both 64KZ memory BLOCKs A and B are logically placed in memory BANK N (BANK 0 - 7) by positioning the corresponding switch BANK N = IN; a BLOCK is logically removed from memory BANK N by positioning the corresponding switch BANK N = OUT. Note that the BANK select switches are strobed **during** the execution of an OUT 40H,A instruction; re-positioning the switches between strobes has no effect on memory BANK activity. A memory BLOCK may be placed in any combination of memory BANKs (none, one, several, or all). Setting BANK 0 - BANK 7 = OUT completely removes BLOCK A/B from the memory map, except possibly after a system RESET or during DMA cycles (see Sections 2.4 and 2.5).

Example 4

Setting the 64KZ switches as shown in Figure 12 produces the memory map shown in the same figure. Note that BLOCK A and BLOCK B have both been assigned to the same 16-bit address space (8000H - FFFFH), but in disjoint memory BANKs.

Figure 12
Example 4 Configuration



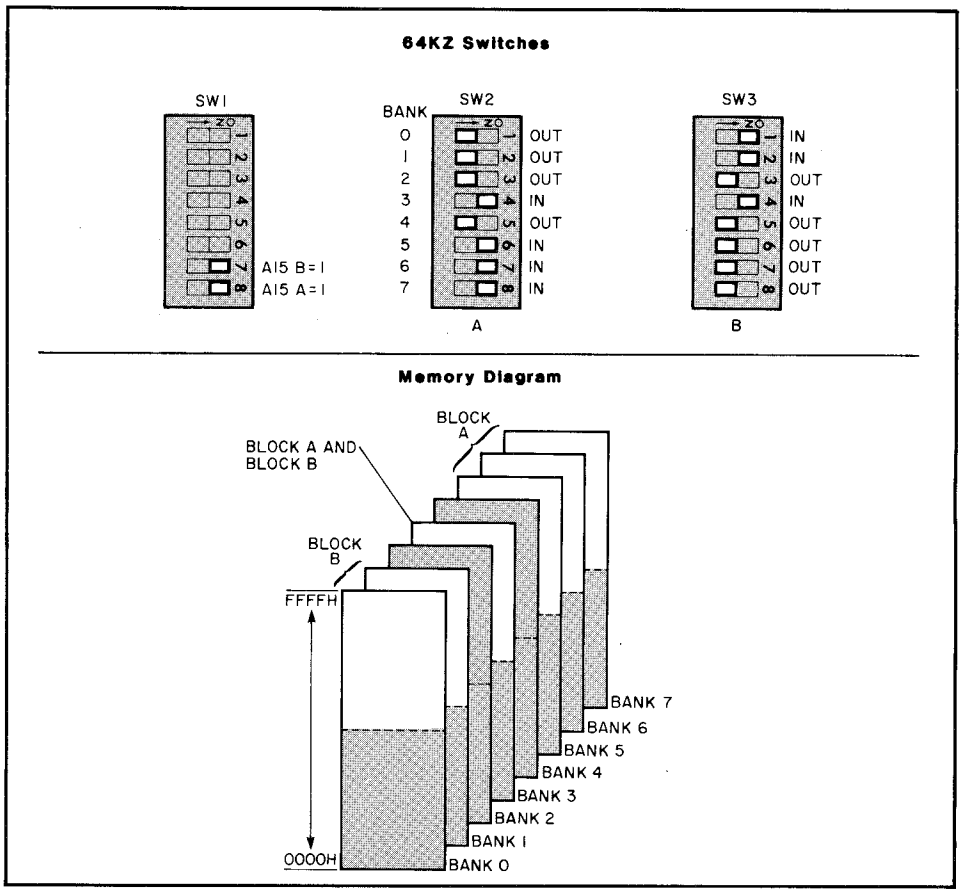
Example 5

The 64KZ switch settings shown in Figure 13 produce the memory map shown in the same figure. Notice that BLOCK A and BLOCK B conflict (both may be simultaneously enabled) in memory BANK 3. If memory BANK 3 is activated, a conflict will result when the CPU memory reads a 8000H - FFFFH address, as **both** BLOCK A and BLOCK B will be actively attempting to drive the S-100 Data In bus lines DI0 - DI7. While no physical damage to the 64KZ would result, this type of configuration is not recommended as any memory read data from the conflict memory area is unpredictable.

The two preceding examples illustrate an important point. The 64KZ, and other Cromemco memory cards with BANK SELECT, may be configured so as to place memory modules in an overlapping 16-bit address space. In such cases, BLOCK A, BLOCK B, and any other system memory module with BANK SELECT, should be placed in disjoint memory BANKs to preclude memory read conflicts. While doing this eliminates one conflict source, a software induced conflict may also result as discussed later in EXAMPLE 6.

Memory BANKs are activated and deactivated under software control. Each Cromemco memory board with BANK SELECT contains an

Figure 13
Example 5 Configuration

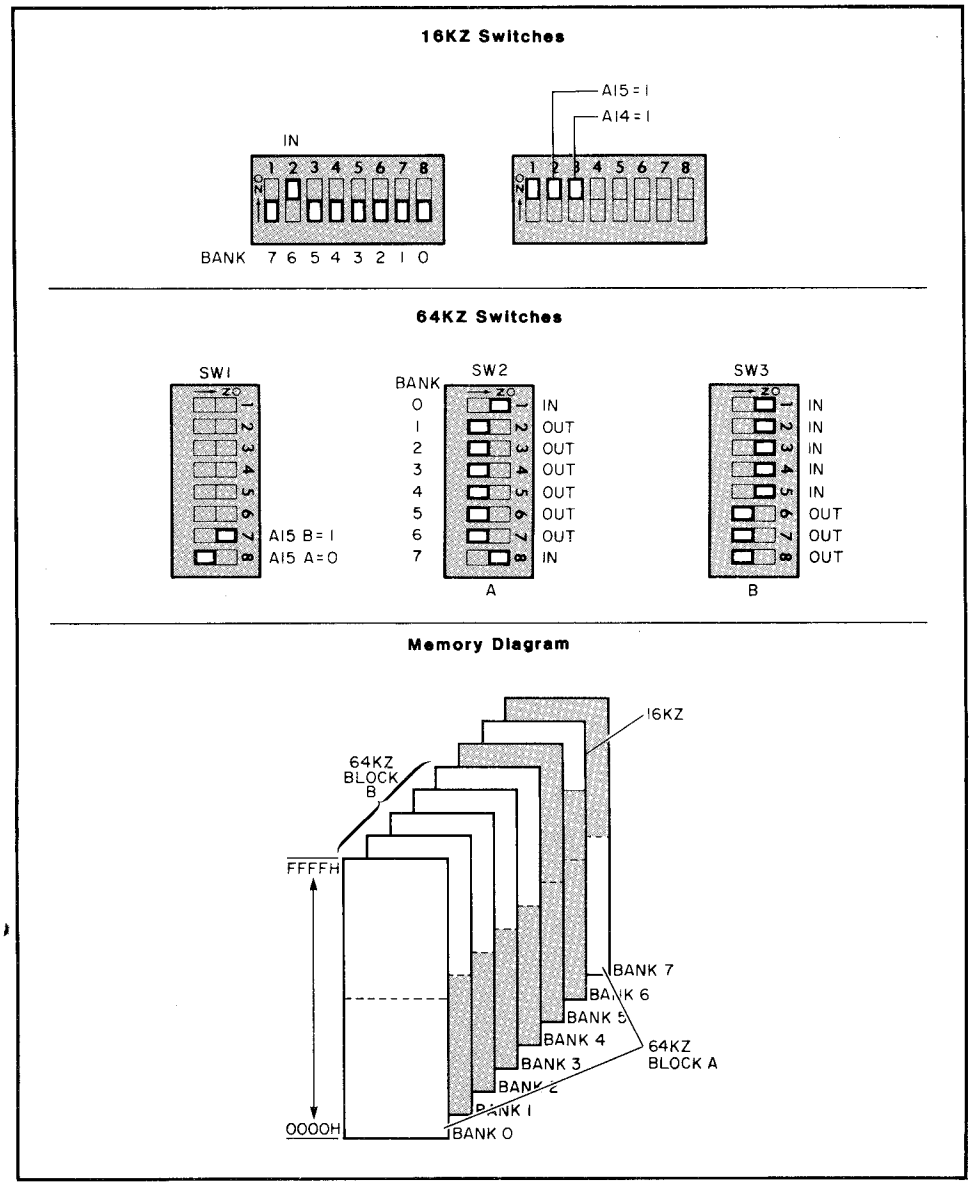


integral OUT PORT with port address = 40H. The CPU controls memory BANK activity by outputting a **BANK SELECT control byte** to all system output ports with address 40H. Each bit of the control word manages one memory BANK. BIT 0 (LSB) controls BANK 0, BIT 1 controls BANK 1, and so on. Outputting a logic 1 control word bit activates its corresponding memory BANK; outputting a logic 0 control word bit deactivates its memory BANK.

Carefully note that if a 64KZ BLOCK is currently switch mapped into **any** memory BANK activated by a control byte, the physical BLOCK is enabled for normal read/write access and the BLOCK's corresponding BANK ACTIVE LED is lit.

Example 6 Assume your system contains both Cromemco 64KZ and 16KZ memory boards configured as shown in Figure 14.

Figure 14
Example 6 Configuration



Then:

- Executing the two instructions below activates memory BANKs 3 and 5 and deactivates all others. Since BLOCK B is in BANK 3 and neither BLOCK A nor the 16KZ is in BANKs 3 or 6, then BLOCK A is inaccessible (floating) when the CPU addresses 0000H - 7FFFH, and BLOCK B enables when the CPU addresses 8000H - FFFFH for either DMA or non-DMA access since the 16KZ is in an inactive BANK.

ADDR	OBJECT	MNEMONIC	COMMENT
C000 C002 C004	3E28 D340 —	LD A,00101000B OUT 40H,A —	;0010 1000 into A register ;control word to out 40h ;next instruction

- Executing the two instructions below activates memory BANKs 3 and 6 and deactivates all other BANKs. Since the 16KZ is in BANK 6 and 64KZ BLOCK B is in BANK 3, both boards are enabled whenever the CPU addresses C000H - FFFFH. Notice that even though the 16KZ and BLOCK B are switch mapped into disjoint memory BANKs, a *software induced* memory read conflict has been created by simultaneously activating two disjoint BANKs. Thus, care must be exercised to avoid enabling two address overlapping memory modules in disjoint memory BANKs at the same time. Also note that no conflict exists when the CPU addresses 0000H - BFFFH since 64KZ BLOCK A is inaccessible at 0000H - 7FFFH and only 64KZ BLOCK B occupies 8000H - BFFFH.

ADDR	OBJECT	MNEMONIC	COMMENT
4000 4002 4004	3E48 D340 —	LD A,01001000B OUT 40H,A	;0100 1000 INTO A register ;control word to out 40H ;next instruction

- Executing the two instructions below activates BANK 0 and deactivates all other BANKs. Since both BLOCK A and BLOCK B are in BANK 0 and the 16KZ is in BANK 7, both BLOCK A and B are accessible for DMA and read/write access when the CPU addresses 0000H - FFFFH and the 16KZ is inaccessible.

ADDR	OBJECT	MNEMONIC	COMMENT
0000 0002 0004	3E01 D340 —	LD A,00000001B OUT 40H,A	;0000 0001 into A register ;control word to out 40h ;next instruction

- Executing the two instructions below activates BANK 5 and deactivates all other BANKs. Since neither BLOCK A nor BLOCK B nor the 16KZ is in BANK 5, all memory modules become inaccessible after BANK switching. Notice that after these two instructions are executed, the system CPU finds itself with no program code to execute. Therefore, these instructions would not be executed in actual practice.

ADDR	OBJECT	MNEMONIC	COMMENT
0000	3E20	LD A,00100000B	;0010 0000 into A register
0002	D340	OUT 40H,A	;control word to out 40h
0004	—	—	;next instruction

As memory BANKs are switched, **the user must be careful to maintain program continuity.** That is, the programmer must be certain that after BANKs are switched, the CPU correctly fetches the next sequential instruction for execution. Two facts bear directly on maintaining program continuity. First, memory BANKs are activated/ deactivated ***in unison*** during the last machine cycle of the OUT 40H,A instruction. Second, the CPU itself is ignorant of memory BANK boundaries. It simply proceeds to advance the Program Counter to the next sequential address after the OUT instruction and fetches the opcode for execution. Thus, the example code segment above which deactivates all BANKs leaves the CPU with no program to execute after BANKs are switched. (The CPU would likely read floating data lines as opcode 0FFH, and execute an RST 38H instruction as a result.)

A straightforward method for maintaining program continuity is to make one read/write memory module common to all activated memory BANKs, thereby allowing the stack and data to be passed freely among switched memory BANKs. EXAMPLE 3 and the example below illustrate this idea.

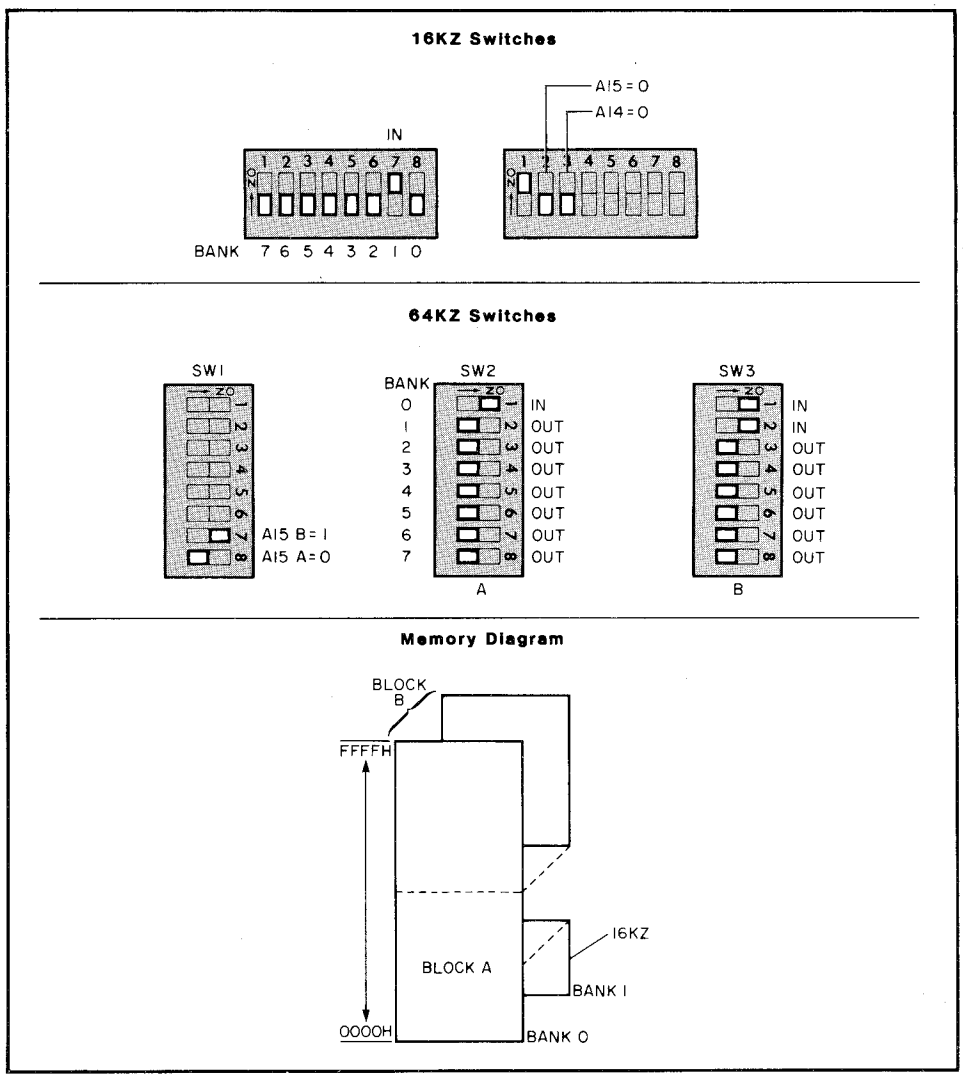
Example 7

Assume your system contains both a Cromemco 64KZ and a 16KZ configured as shown in Figure 15. Assume also that the contents of 16KZ memory location 1000H, BANK 1, is needed by a program executing in 64KZ memory, BLOCK A, BANK 0. Since the addresses of 64KZ BLOCK A and the 16KZ overlap, the safe way to transfer the data is through an I/O handler routine in 64KZ BLOCK B (notice that a program which is resident in BLOCK A cannot deactivate BANK 0 then activate BANK 1 since contact with the running program would be lost). Sample code to effect the transfer is illustrated below. Initially assume that only BANK 0 is active.

Output port 40H manages eight memory BANKs on all Cromemco memory cards with BANK SELECT. Provision has been made on the 64KZ to optionally reassign the BANK SELECT port address to a value other than 40H. Each new BANK SELECT port address would then manage eight additional memory BANKS, and these additional BANKs could be used either to expand the gross amount of addressable memory (sixteen 64KZ cards would provide 1.024 Mbytes of RAM organized in sixteen memory BANKs, and still leave room for five other cards in a twenty-one slot S-100 motherboard) or to create more address overlapping memory partitions in multi-user type applications.

ADDR	OBJECT	LABEL	MNEMONIC	COMMENT
		;		
		; BLOCK A MAIN PROGRAM		
		;		
0400	21001		LD HL,1000H	;load HL with 16KZ address
0403	CD00C0		CALL RD16KZ	;call 16KZ read routine
0406	—		CONT: —	;read data in B register
		;		
		; BLOCK B 16KZ I/O ROUTINE		
		;		
C000	3E02	RD16KZ:	LD A,2	;activate bank 1 CNTRL word
C002	D340		OUT 40H,A	;now in bank 1 with 16KZ
C004	46		LD B,[HL]	;16KZ (1000H) to Reg B
C005	3D		DEC A	;activate bank 0 CNTRL word
C006	D340		OUT 40H,A	;now in bank 0 with block A
C008	C9		RET	;return to 'CONT' in block A

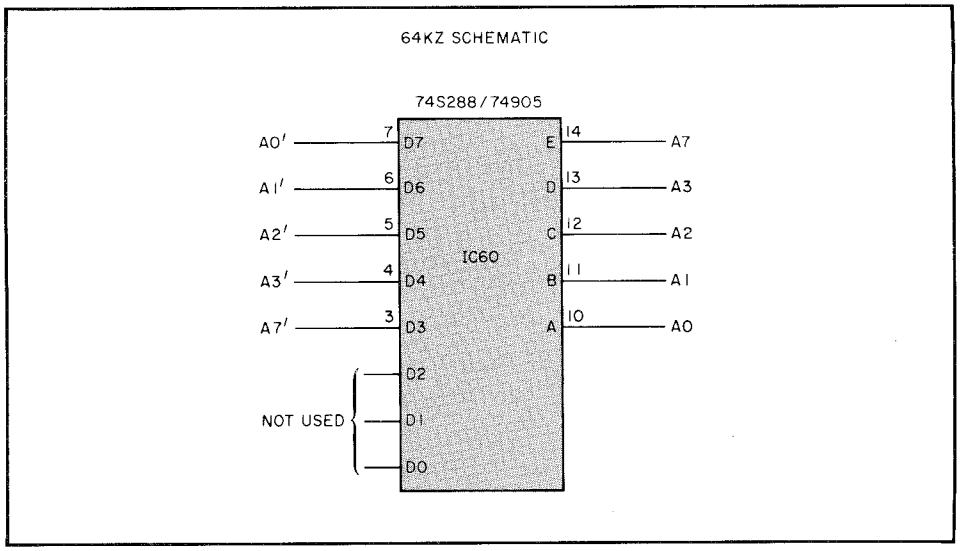
Figure 15
Example 7 Configuration



The BANK SELECT port address is changed from its factory wired 40H value by removing the jumper plug at location IC60 and installing a properly programmed 74S288 PROM in the empty IC60 socket. With the PROM inserted, the 64KZ hardware still responds to port address 40H, but the 74S288 converts the desired port address into 40H data. More specifically, S-100 bus address lines A7, A3, A2, A1 and A0 drive the 74S288 as inputs, and S-100 bus lines A6, A5, A4 bypass the PROM (see Figure 16).

Five of the eight 74S288 data out lines then serve as encoded address lines A7', A3', A2', A1' and A0'. The BANK SELECT output port is then addressed when the combined coded and uncoded lines form address 40H or when A7' A6 A5 A4 A3' A2' A1' A0' = 0100 0000B. The 74S288 should be programmed to output all zeros when the desired port address drives the five PROM address lines, and to output at least one logic 1 bit on every undesired port address. This scheme results in thirty-two permissible BANK SELECT port addresses; 40H - 4FH and C0H - CFH. Also note that the 74S288 may be programmed to supply 40H data in response to more than one port address.

Figure 16
Changing the Bank Select Port Address



2.4 Direct Memory Access

An S-100 bus resident device may gain direct access to system memory without CPU intervention by driving control line pHOLD active low. The CPU acknowledges the DMA request by asserting line pHLDA high. The DMA device then uses this level as a signal to actively take control of S-100 address, data out, and control lines. Control is passed to the DMA device when it asserts S-100 bus lines ADDR DSBL, DO DSBL, C/C DSBL and STAT DSBL active low, thereby disconnecting the CPU from system memory. At this point the DMA device must seize control of the system address, control, and data lines to transfer data to or from

memory. DMA implementations are commonly used where a fast, asynchronous memory access, characterized by a high data transfer rate, is required.

Two 64KZ aspects deserve special attention in DMA implementations:

1. In addition to being correctly 16-bit addressed with lines A0 - A15, the 64KZ must also be correctly BANK addressed.

2. The 64KZ card does not provide 4116 RAM refresh during DMA.

DMA memory BANK addressing difficulties have been minimized by the 64KZ DMA OVERRIDE feature (see Figure 10). With switch OVERRIDE A/B = ENABLED, the 64KZ entirely disregards memory BANKs during the time line pHLDA is active high (DMA memory BANK boundaries are OVERRIDDEN, or ignored); BLOCK A/B enabling then becomes only contingent upon address A0 - A15 and whether DMA A/B is IN or OUT. In essence, only two *DMA memory BANKs* exist with OVERRIDE = ENABLED: the active one containing the enabled memory module with DMA = IN, and the inactive one containing all disabled memory modules defined by switches DMA = OUT.

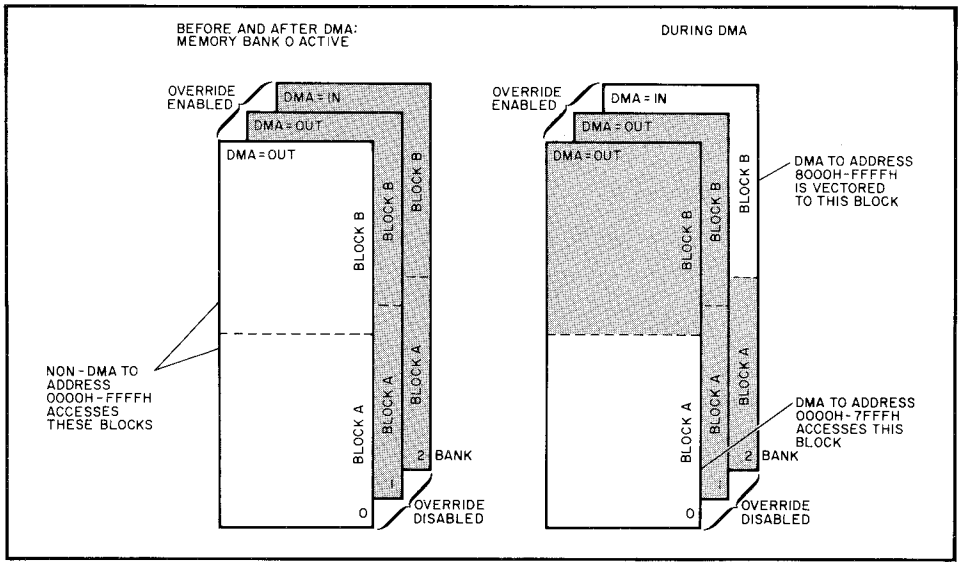
The following example illustrates how to configure the 64KZ in DMA systems with multiple active memory BANKs. The principles and DMA control switch use outlined in the example apply to other Cromemco memory boards with BANK SELECT and DMA OVERRIDE as well. These boards include the 16KZ, the BYTESAVER II, the 32K BYTESAVER, and the 16KPR.

Example 8

Assume a certain system contains three 64KZ cards. On each card, BLOCK A is assigned to the lower 32K of memory and BLOCK B to the upper 32K. The first card is assigned to memory BANK 0, the second to BANK 1, and the third to BANK 2. Assume further that BANK SELECT control word 01H is output to all three boards, giving rise to the configuration shown on the left in Figure 17. Then for *non-DMA* memory read or write cycles, any address in the range 0000H - FFFFH accesses the first 64KZ card in BANK 0.

For DMA (Figure 17, right), assume all BLOCKs A have OVERRIDE = DISABLED. Then DMA to an address in the range 0000H - 7FFFH accesses BLOCK A in BANK 0 since BANK boundaries are *not* overridden in the 0000H - 7FFFH address range and BANK 0 is active (no differentiation is made between DMA and normal CPU addressing since OVERRIDE = DISABLED).

Figure 17
DMA Override



For DMA, assume all BLOCKs B have OVERRIDE = ENABLED, BANKs 0 and 1 have DMA = OUT, and BANK 2 has DMA = IN. Then any DMA to an address in the 8000H - FFFFH range automatically vectors to BLOCK B in BANK 2 (the BLOCK with DMA = IN). BLOCKs B in BANKs 0 and 1 with DMA = OUT automatically disable (float), permitting DMA vectoring to the BLOCK with DMA = IN, even though BANK 0 was active before DMA.

After the DMA transfer is completed, the system memory configuration reverts back to that shown on the left in Figure 17 and the system CPU resumes program execution from the point of interruption.

Note that it is also possible to include memory BANK switching as part of the DMA device's addressing responsibilities when OVERRIDE = DISABLED. This would not typically be done, however, as the DMA device would also then be responsible for ascertaining pre-DMA memory BANK status and BANK restoration after the DMA transfer. These additional tasks slow down DMA access and complicate the DMA controller. Except for extraordinary situations where more than one BANK is accessed during DMA, the OVERRIDE and DMA IN/OUT switches would be used to provide complete DMA memory BANK switching control.

During DMA (while pHLDA is active high), the 64KZ suspends dynamic memory refresh. It is then the responsibility of the DMA device to provide memory refresh **if necessary.**

During normal (non-DMA) operation, the 64KZ provides transparent M1-cycle refresh. That is, both the Z80 and 8080 processors require at least four clock cycles during an M1 cycle to fetch and decode the

instruction opcodes from system memory. The normal opcode fetch occurs during the first two clock cycles leaving the last two available for memory refresh. This technique has the advantage of requiring no overhead time for memory refresh since refresh occurs during CPU *dead time* (thus the term *transparent refresh).* The disadvantage of the technique is that other means must be provided to supply refresh when the CPU is not executing program code (no M1-cycles). The 64KZ does provide refresh in the first two instances (CPU HALTed, CPU WAITing), but not during DMA.

The crucial refresh specification which must be observed during DMA is that no more than 2 milliseconds may elapse between either a read or a write cycle to each row address A0 - A7, if the DMA device is providing refresh. If the 64KZ is to provide refresh during DMA, the CPU must execute at least 128 M1-cycles during every 2 millisecond interval.

There are two broadly defined DMA accessing techniques: **DMA cycle stealing** and **burst DMA.** Cycle stealing, as the term implies, involves periodically stealing from one to several clock cycles from the CPU for the purpose of DMA. For cycle stealing implementations, the 64KZ typically provides refresh and the specification then translates to a ratio of CPU to stolen DMA cycles. This ratio must be such that, for every 2 millisecond time interval, the CPU executes at least 128 M1 fetch cycles. Assuming the longest instruction for both the Z80 and the 8080 (23 T-cycles), the CPU would then conservatively require 128 X 23 X 0.25 usec = 736 usec out of any 2 msec = 2000 usec interval (assuming a 4 MHz system clock) to provide refresh, or a 37% duty cycle. Assuming a 2 MHz clock, the CPU then would require 74% of all clock cycles to assure memory refresh. This simplified analysis assumes that DMA cycles are uniformly distributed among CPU cycles.

In burst DMA implementations, the DMA device typically accesses memory more infrequently, but each access is for a longer period of time than in cycle stealing implementations. If each DMA burst access is less than 1 msec in duration, and spaced more than 1 msec apart, then the 64KZ card itself will provide adequate dynamic memory refresh. If either the DMA access time is lengthened, or the interval between successive accesses is shortened, then it becomes the responsibility of the DMA device to provide refresh. In such a case, refresh may be easily provided by reading from (or writing to) any 1/2 page (128 bytes) of contiguous 64KZ memory at least once every 2 msec.

2.5 Multi-User Basic Application

The 64KZ is the ideal memory card for use in a Cromemco Multi-User BASIC system. The Multi-User BASIC system timeshares up to seven different BASIC programs resident in the same S-100 bus system. Each BASIC user is assigned to a different memory BANK (BANKs 0 - 6), and may be allocated either 16 or 32 Kbytes of read/write memory starting at

address 0000H for program text. The BASIC operating system requires 48 Kbytes of read/write memory. Of the 48 Kbytes, 32 Kbytes spanning 8000H - FFFFH are assigned to all memory BANKs 0 - 7; this area contains the BASIC interpreter program itself and part of the operating system. The remaining 16 Kbytes, spanning 4000H - FFFFH, are assigned to memory BANK 7 only.

This memory area currently contains most of the system disk and terminal I/O routines. The memory area 0000H - 3FFFH in BANK 7 is not used at this time.

Figure 18
Multi-User Basic

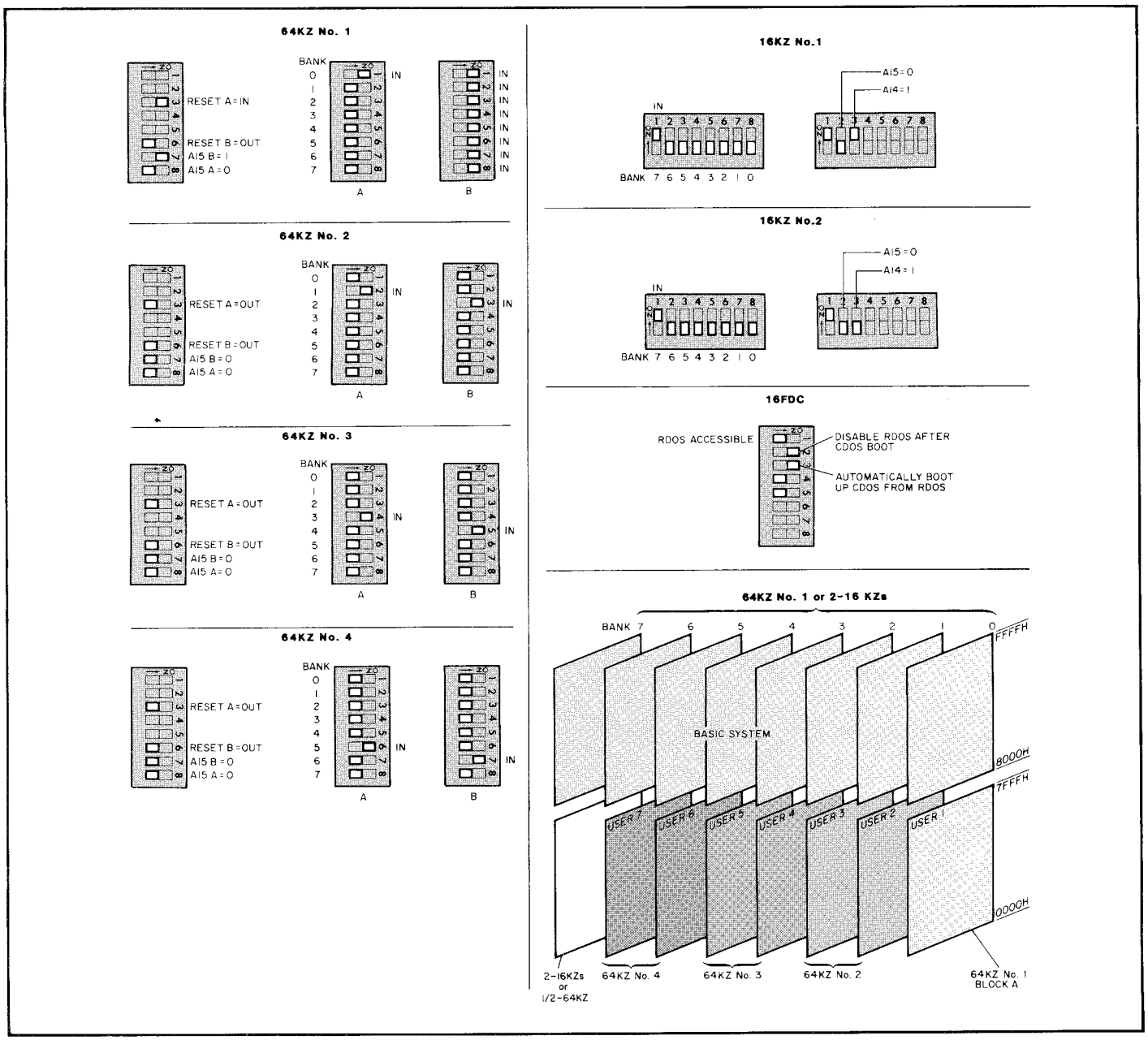
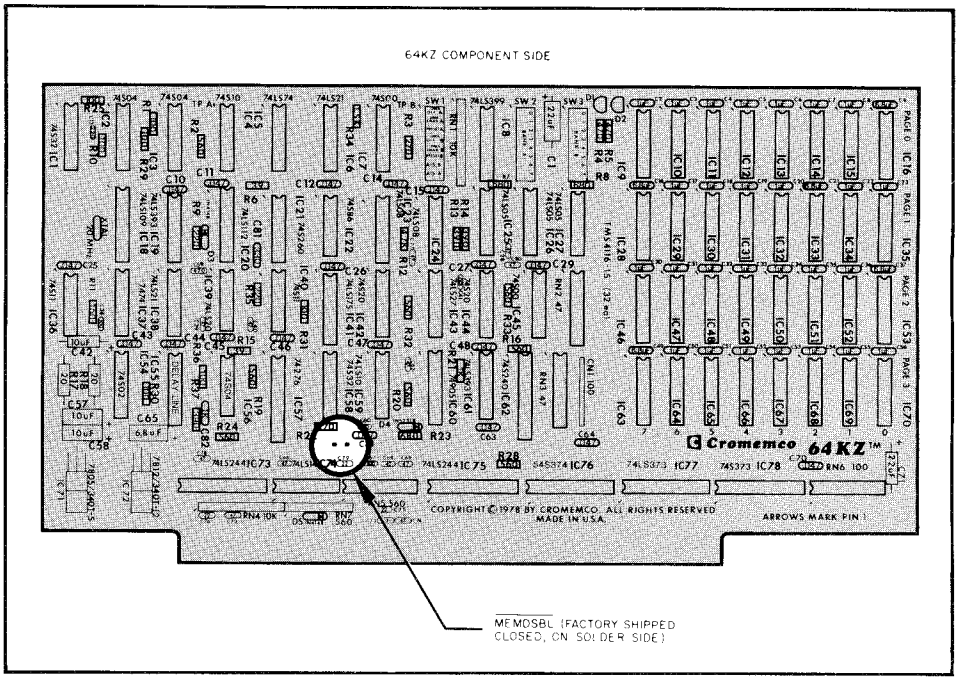


Figure 18 illustrates how four 64KZ boards and two 16KZ boards may be used to form a fully configured seven user BASIC memory system.

2.6 64KZ Jumper Options

The 64KZ provides a jumper selectable $\overline{\text{MEMDSBL}}$ option (see Figure 19).

Figure 19
64KZ Jumper Pad Locations



$\overline{\text{MEMDSBL}}$ is a function assigned to S-100 bus pin 67 which, when asserted active low, completely disables the 64KZ memory card. This hardware controlled memory disable capability may be removed by cutting the solder trace between the two pads indicated in Figure 19, on the board solder side. If your system uses pin 67 for some other function, or if the line is not used at all, Cromemco recommends that you cut the trace. The $\overline{\text{MEMDSBL}}$ function should be left in its factory wired enabled state in systems which require a phantom memory capability.

Please refer to the 64KZ Block Diagram shown in Figure 20 and to the 64KZ Schematic Diagram while reading this section.

Figure 20
64KZ Block Diagram

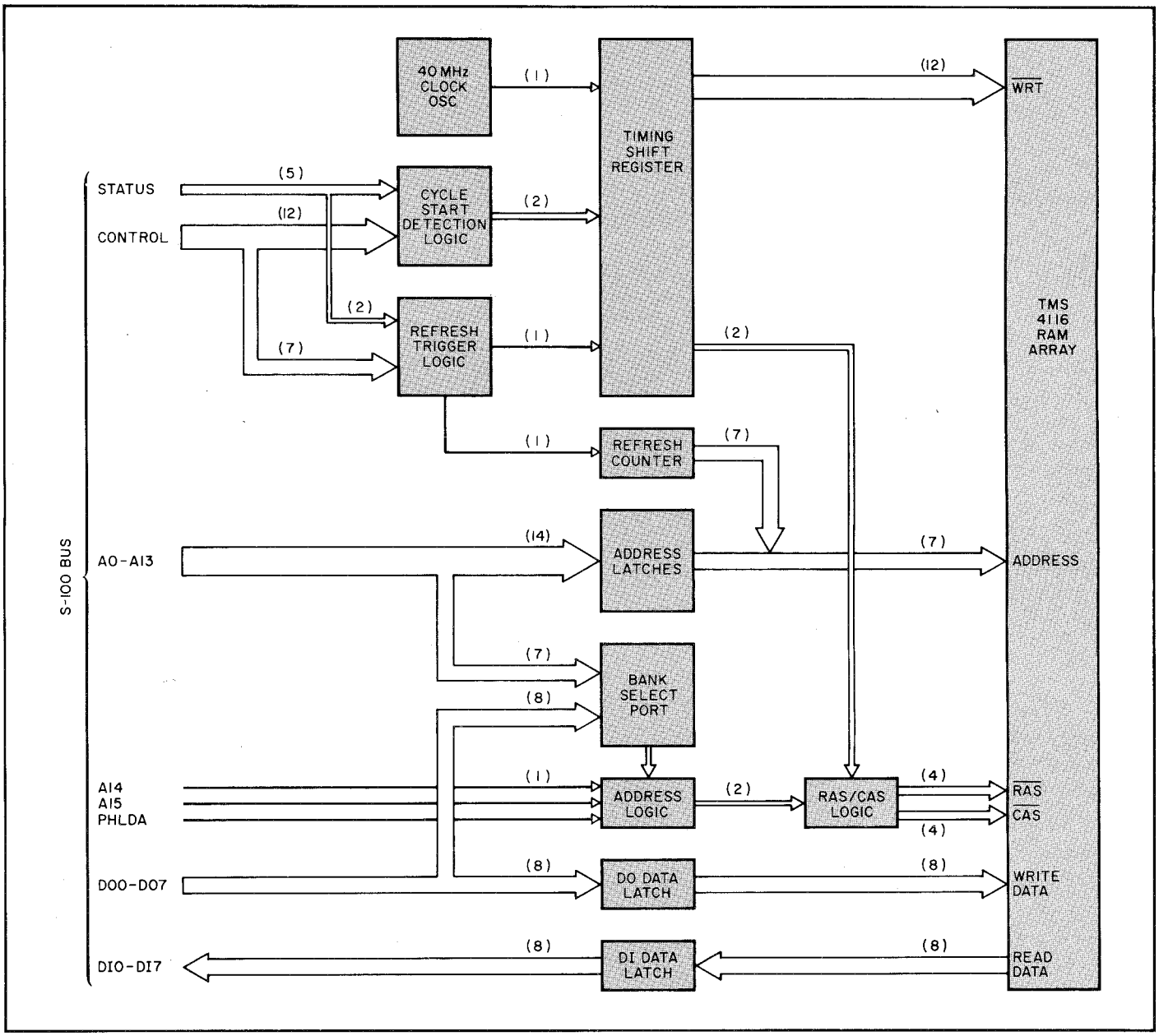


Table 1

OUTPUT	BOOLEAN EXPRESSION	CYCLE TYPE
IC57 PIN 16	$s\overline{OUT} \cdot sWO \cdot CPU \cdot \overline{MREQ} \cdot [pSYNC \uparrow\downarrow]$	Z80 WRITE
IC20 PIN 9	$[CPU \cdot s\overline{MEMR} \cdot \overline{MREQ} \cdot \overline{RFSH}] \downarrow\uparrow$	Z80 READ
IC20 PIN 5	$CPU \cdot p\overline{SYNC} \cdot [sM1 \downarrow\uparrow]$	Z80 M1 FETCH
IC57 PIN 15	$sMEMR \cdot pSYNC \cdot \overline{MREQ} \cdot CPU \cdot [02 \downarrow\uparrow] +$	8080 READ +
	$sMEMR \cdot pSYNC \cdot \overline{MREQ} \cdot DMA \; [02 \uparrow\downarrow]$	DMA READ
IC5 PIN 9	$(CPU + DMA) \cdot \overline{MREQ} \cdot [MWRITE \downarrow\uparrow]$	DMA & F.P. WRITE
	where $CPUT = pHLDA \cdot \overline{CCDSBL}$ and $DMA = pHLDA \cdot CCDSBL$	

64KZ Theory of Operation

The 64KZ is functionally built around a delay line which provides control signals to the 4116 dynamic RAM array at precisely spaced intervals during a memory cycle. These intervals are marked by four taps on the delay line which become active 20, 70, 120, and 170 nanoseconds (ns) after the start of a cycle request.

The RAM array is organized into two independently addressable blocks of 32 Kbytes, with bank select determining which memory board is accessible, A15 determining which 32 Kbyte block is selected, and A14 determining which row in the blocks will be used. A separate Bank Select, A15, and DMA configuration is provided for each block defined by switch settings on SW1. A $\overline{MEMDSBL}$ line, which disables the CAS signal, is provided for use with phantom memory.

Operations with the 64KZ (read, write, op code fetch, and DMA) are detected and latched through S-100 control and status signal logic gating. When a valid memory cycle is started, the cycle request is passed on to IC36 pin 10. The output of IC36 (pin 8) goes low causing the high order address buffer IC76 to latch the current address, and IC58 to start the delay sequence. The cycle request at the input to the delay line asserts $\overline{RAS}$ through IC45 for all four memory rows. After a 20 ns hold time for the RAMs to latch row address A0-A6 (IC75), the address multiplexers are switched and enable column address A7-A13 (IC76) on to the RAM array address lines. The 70 nsec tap output causes $\overline{CAS}$ to go low through IC42 or IC44 on the selected row 50 ns after the addresses are switched. The RAM row receiving $\overline{CAS}$ is determined by the combination of A14, A15, and Bank Select. The 120 nsec tap output is gated through IC54 (pin 3) and IC4 (pin 1) to the $\overline{PRESET}$ of IC18 (pin 11). This forces $\overline{PRESET}$ to remain high, preventing a transparent refresh cycle from starting during this part of the cycle. The 170 nsec tap output is gated through IC7 (pin 1) generating a signal that clears the S-100 control and status signal flip-flops with a short pulse (about 20 nsecs).

The output of the delay line is inverted and fed back to the input of the delay line through IC58 and IC56 causing $\overline{RAS}$ to turn off. The 70 nsec tap now turns off $\overline{CAS}$ and latches the read data from the RAM array onto IC78. The read data is now ready for the S-100 bus access. The 120 nsec tap goes low next causing IC54 pin 1 to go high. The $\overline{PRESET}$ on IC18 can now be driven low initiating a transparent refresh request if the following conditions are true: 1) the current cycle is an M1 cycle (IC4 pin 13) and 2) the refresh request counter has counted 16 clock cycles (IC4 pin 2).

The above delay line cycle is common to all memory cycles. Details of specific memory cycles are described below.

READ

PHLDA, (processor hold acknowledge) inactive during a read cycle, is inverted, and ANDed with CCDSBL (control bus disable), also inactive, and delayed $\overline{CCDSBL}$. This ANDed signal, a high, goes to IC39 (pin 13), the read detect NAND gate. $\overline{RFSH}$ (refresh) is inactive and also goes to the read detect NAND gate. $\overline{MREQ}$ (memory request) becomes active, a low state, is inverted, and goes to the read detect NAND gate. At the same time that $\overline{MREQ}$ becomes active SMEMR (S100 memory read) becomes active, a high state, and goes to the read detect NAND gate. Now all inputs to the read detect NAND gate are high states, and the output of the read detect NAND gate goes low, clocking the read flip-flop (IC20 pin 13). When the read flip-flop is clocked, the J K inputs are set up to clock the Q output of IC20 (pin 9) high. The output of the read flip-flop is gated through ICs 21, 36, 58, and 56 to the delay line and initiates the memory cycling described above.

WRITE

PHLDA, inactive during a write cycle, is inverted and ANDed with $\overline{CCDSBL}$, also inactive, and delayed $\overline{CCDSBL}$. This ANDed signal, a high, goes to IC40 the write detect AND gate. $\overline{MREQ}$ becomes active, a low state, is inverted, and goes to the write detect AND gate. At the same time $\overline{MREQ}$ becomes active, $\overline{SWO}$ (S-100 write) also becomes active, a low state, is inverted, and goes to the write detect AND gate. Now all inputs to the write detect AND gate are high states, and the output of the write detect AND gate goes high, setting the J input of the write flip-flop (IC57 pin 19) to a high. The falling edge of PSYNC clocks this input through to the output Q (IC57 pin 16), starting the write cycle. The Q output goes to two NOR gates. The output of NOR gate IC21 pin 5 is latched by IC41, while the output of NOR gate IC21 pin 6 starts the sequencing cycle of addressing, RASing, clocking the latched write (from IC21 pin 5) to the RAM, CASing, and clearing the write flip-flop.

If the write is a front panel deposit, the front panel will generate an SM1. This SM1 is gated through IC5 pin 1 and clocked through IC37 (pin 5) to IC36 pin 1 to start a read cycle. The data just written is latched in the data out buffer so the front panel LEDs may display the data.

M1 PHLDA, inactive during a write cycle,is inverted, and ANDed with $\overline{\text{CCDSBL}}$, also inactive, and delayed $\overline{\text{CCDSBL}}$. This ANDed signal, a high, goes to IC23 (pin 9) the M1 (machine cycle one, or op code fetch) detect AND gate. PSYNC is inverted and goes to the M1 detect AND gate. This sets the J input on IC20 (pin 3) high at the start of the cycle. When the inverted M1 becomes active, it clocks the M1 flip-flop, setting the Q output (IC20 pin 5) high, starting the M1 cycle.

DMA PHLDA and $\overline{\text{CCDSBL}}$ are active during DMA, forcing the output of the ANDed signals to go to a low state, and disabling CPU read, write, and M1 detect gates. The disabled read NAND gate puts a high on the input of IC24 (pin 12), and the NORed $\overline{\text{CCDSBL}}$ and delayed $\overline{\text{CCDSBL}}$ put a high on the DMA write detect OR gate (IC58 pin 3), which puts a high on the other input of IC24 (pin 13) enabling the DMA read detect AND gate IC40. The delay on $\overline{\text{CCDSBL}}$ is to prevent a false cycle from starting when switching control and status signals from the permanent bus master to the temporary master. If the DMA is a read, SMEMR (S100 memory read) becomes active, a high, and goes to the DMA read detect AND gate. The output of the DMA read detect AND gate clocks the DMA read flip-flop (IC57 pin 13), and starts a DMA read cycle. If the DMA is a write, MWRITE (memory write) becomes active, a high, clocks the DMA write flip-flop (IC5 pin 11), and starts a DMA write cycle.

REFRESH A 4116 16K by 1 dynamic RAM is used on the 64KZ, so provisions must be made to periodically restore (refresh) the data logic levels. There are two means of refresh provided, transparent and autonomous refresh. Transparent refresh occurs during T3 and T4 of the M1 cycle, while autonomous refresh occurs during extended WAIT states. A request for transparent refresh is generated in the following manner: The system clock, prescaled by 16 through IC19 (pin 6), is latched and passes a refresh request through the refresh detect AND gate IC38 (pin 6), when M1 and PSYNC are true and the DMA condition is false, to IC18 (pin 2) where it is latched on the next system clock pulse. During the latter part of T2 of an M1 cycle, the 120 nsec tap goes low. This is inverted through IC54 (pin 3) causing IC4 (pin 12) to PRESET the transparent refresh request flip-flop (IC18 pin 10 goes high). This initiates the refresh cycle by driving IC54 pin 10 low.

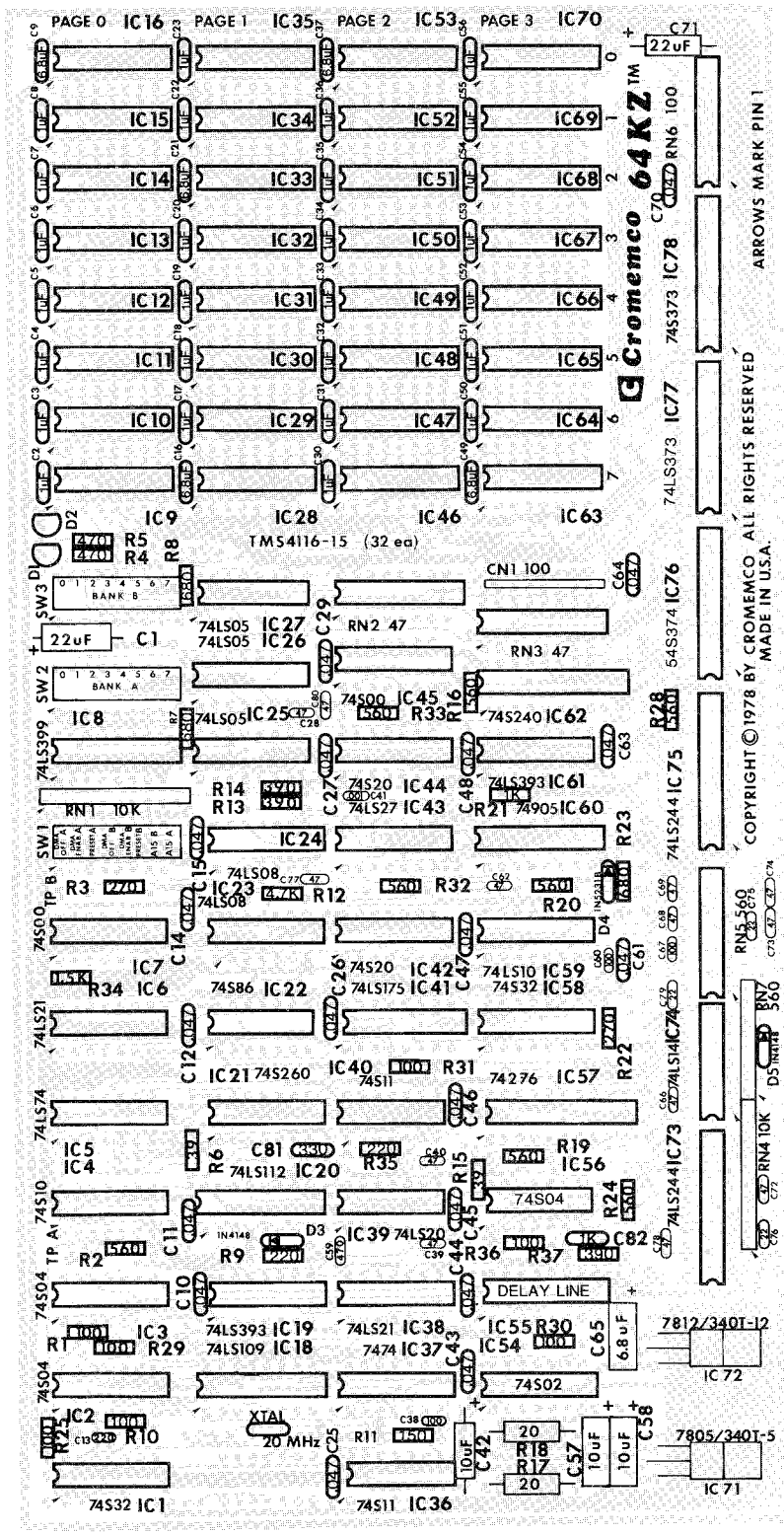
The autonomous refresh detect NAND gate IC39 (pin 8) enables a counter IC19 (pin 12) when ever XRDY, FRDY, or PRDY are pulled low. The enabled counter prescales the system clock by 16. This counter clocks IC57 (pin 8) causing the Q output (IC57 pin 6) to go high, requesting an autonomous refresh cycle. This is gated through IC24 pin 3 causing IC54 pin 10 to go low.

The refresh request, transparent or autonomous, is gated through IC54 to IC36 pin 11 which starts the delay line cycle. The output of IC54 pin 10 also clears IC37 (pin 13) which disables $\overline{CAS}$ on all rows (refresh only uses $\overline{RAS}$ to restore the logic level) and increments the refresh address counter on its $\overline{Q}$ output. IC1 enables the refresh address buffer onto the RAM array.

Appendix A 64KZ Parts List

	Part No.	Description	Qty
Resistors			
	001-0002	39Ω ¼W	2
	001-0007	100Ω ¼W	7
	001-0008	150Ω ¼W	1
	001-0010	220Ω ¼W	2
	001-0011	270Ω ¼W	2
	001-0013	390Ω ¼W	3
	001-0014	470Ω ¼W	2
	001-0015	560Ω ¼W	8
	001-0016	680Ω ¼W	2
	001-0018	1.0KΩ ¼W	1
	001-0020	1.5KΩ ¼W	1
	001-0024	4.7KΩ ¼W	1
	001-0046	20Ω 3W	2
	001-0067	680Ω ½W	1
Resistor Networks			
	003-0006	560Ω 8 pin	1
	003-0020	100Ω 16 pin	1
	003-0021	560Ω 16 pin	1
	003-0022	47Ω 16 pin	2
	003-0024	10KΩ 10 pin	2
Capacitors			
	004-0000	mono 47 pF	13
	004-0009	mono 100 pF	3
	004-0022	crdc .001	1
	004-0028	22 μF	2
	004-0032	tant 10 20V	3
	004-0034	tant 6.8	1
	004-0035	crdc 330 pF	1
	004-0041	mono 22 pF	3
	004-0043	mono .001	1
	004-0061	.047 50V	19
	004-0070	mono 1 μF	27
	004-0076	tant 6.8 15V	5
	004-0077	mono 220 pF	1
	004-0080	mono 470 pF	1
	005-0001	network 100pF 8 pins	1
Diodes			
	008-0002	IN914/IN4148	2
	008-0006	IN52318	1
	008-0020	TIL-211 green LED	2

Appendix A 64KZ Parts List

	Part No.	Description	Qty
ICs			
	010-0019	7474	1
	010-0035	74S10	1
	010-0036	74S00	2
	010-0042	74LS175	1
	010-0051	74LS109	1
	010-0055	74LS74	1
	010-0060	74LS21	2
	010-0061	74LS14	1
	010-0063	74LS10	1
	010-0064	74LS08	2
	010-0065	74LS05	3
	010-0085	74S373	1
	010-0090	74S32	2
	010-0091	74276	1
	010-0094	74S260	1
	010-0095	74LS20	1
	010-0100	74LS244	2
	010-0102	74LS373	1
	010-0111	74S11	2
	010-0112	74LS27	1
	010-0115	74LS399	1
	010-0122	74S02	1
	010-0123	74S04	3
	010-0125	74S86	1
	010-0126	74LS112	1
	010-0139	74S240	1
	010-0141	74LS393	2
	010-0155	74S20	2
	010-0318	delay line	1
	010-0319	54S374	1
	011-0019	TMS-4116-15JL 16K RAM	32
	012-0001	7805/340T-5 regulator	1
	012-0002	7812 regulator	1
Miscellaneous			
	007-0003	choke 3.9 μH	1
	013-0002	8 pos DIP switch	3
	015-0020	#6 split ring lock washer - cad	2
	015-0074	6-32 hex nut 5/16 pattern	2
	015-0083	4-40X3/16 hex threaded standoff	2
	015-0084	4-40X¼ panhead screw - black	2
	015-0085	4-40X¼ panhead screw - nylon	2
	015-0129	6-32X7/16 panhead screw - cad	2
	016-0059	64KZ heat sink	1
	017-0001	14 pin burndy socket	29
	017-0002	16 pin burndy socket	40
	017-0004	20 pin burndy socket	7
	017-0044	wire wrap post	2
	017-0069	16 pin DIP shunt	1
	021-0038	25 pin bus bar	1
	026-0002	20 MHz crystal	1

64KZ Part Location Diagram