CBM Expansion Memory Board

User's Guide

commodore

CBM Expansion Memory Board

User's Guide

Part Number 324008

June 1981

COMMODORE BUSINESS MACHINES, INC.
3330 Scott Blvd.
Santa Clara, Ca. 95051

WARRANTY DISCLAIMER

The computer program supplied in this manual is provided to you, their user, with no warranty of any kind. Although due care has been taken to ensure the correct operation of this program, no representation is made about its fitness for any particular use or about the accuracy of its results. Commodore Business Machines, Inc., its distributors, retailers, and agents thus assume no responsibility and accept no consequential, incidental, or other liability arising from the use of this program. Some states do not allow the exclusion or limitation or implied warranties or liability for incidental or consequential damages, so the above limitations may not apply to you.

TABLE OF CONTENTS

ILLUSTRATIONS

TABLES

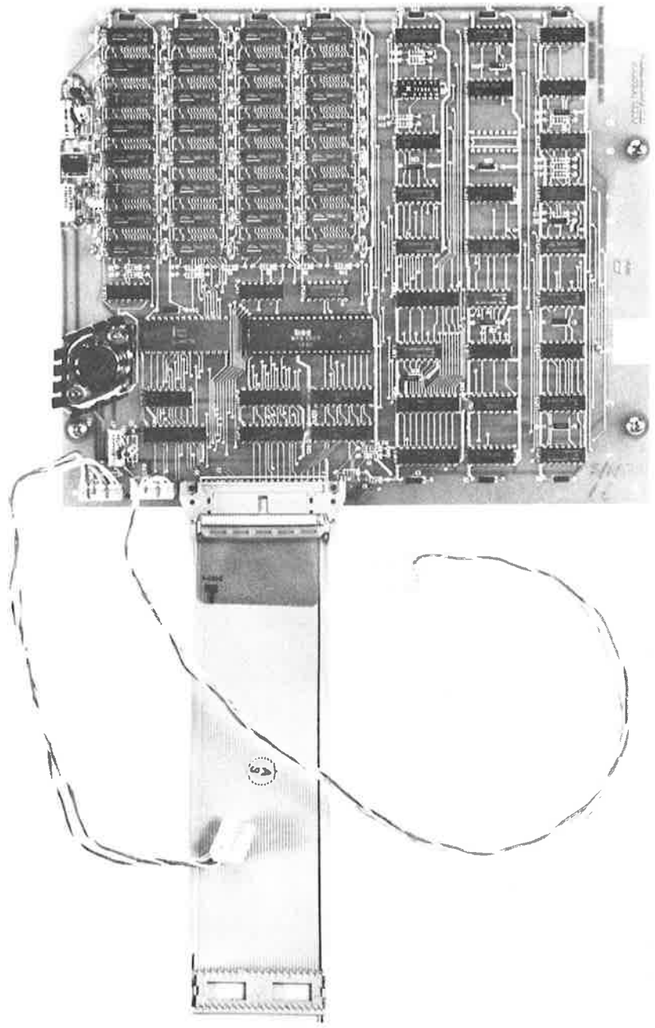


Figure 1-1. 64–k Expansion Memory Board

Chapter I

GENERAL DESCRIPTION

INTRODUCTION

The Expansion Memory Board adds 64k-bytes of RAM to the CBM 8032 providing a total of 96k-bytes of RAM. The expansion memory is mapped into main memory in 32k-byte blocks. Only one 32k-byte expansion block can reside in main memory at one time.

The Expansion Memory Board is mounted in the CBM 8032 console by four mounting brackets, and is connected to the computer by two power cables and one logic cable.

Control of the expansion memory is through a register on the Expansion Memory Board that provides selection of two 16k-byte blocks, write protection, and enabling of the expansion memory. Additional bits in the register allow the programmer to PEEK at I/O and screen memory.

The Expansion Memory Board was designed to be used by programmers that are familiar with the internal software operations of the CBM 8032.

This manual contains installation and operating instructions for the Expansion Memory Board (see Figure 1-1). The manual is organized as follows:

Chapter 1.	Contains a general description of the board, lists its contents and provides unpacking instructions.
Chapter 2.	Contains detailed installation instructions for the Expansion Memory Board.
Chapter 3.	Contains operating and programming information, and sample programs for the expansion memory.

PACKAGE CONTENTS

The Expansion Memory Board consists of the following items:

1. An Expansion Memory Board, part number 8032099
2. User's Guide, part number 324008

UNPACKING/PACKING INSTRUCTIONS

Inspect the carton containing the Expansion Memory Board before opening it. If there is external damage, request that the carrier's agent be present when the carton is opened. Save all packing materials.

Inspect each piece of the Expansion Memory Board as it is unpacked. If it is damaged notify the carrier and the nearest Commodore sales office immediately. Retain the shipping carton for the carrier's inspection. Verify that the contents of the Expansion Memory Board match the items listed under "Package Contents."

If an item of the Expansion Memory Board is to be reshipped to Commodore, attach a tag to the item identifying the owner and the service or repair to the item. Pack the item in the original packing material if available.

RELATED DOCUMENTS

The following documents contain information relative to the Expansion Memory Board:

1. Commodore Business Computer User's Guide, Series 8000, part number 320894.

2. User's Reference Manual, Commodore BASIC Version 4.0, part number 321604.

3. MCS6500 Microcomputer Family Hardware Manual, MOS Technology, Inc., part number 6500-10A.

Chapter II

INSTALLATION

INTRODUCTION

The following paragraphs describe the tools and procedures required to install the Expansion Memory Board into a Commodore Model 8032 computer.

TOOLS REQUIRED

The following tools are required to install the Expansion Memory Board:

1. Phillips screwdriver
2. A 4/40 Allen wrench
3. IC puller

INSTALLATION PROCEDURE

Referring to Figure 2-1, use the following procedure to install the Expansion Memory Board:

WARNING
Turn power OFF at the CBM 8032 and ALL peripherals by removing the power cords from the power outlets. Failure to heed this warning may result in death or injury to personnel.

1. Use a Phillips screwdriver to remove the two mounting screws under each side of the CBM 8032 front cover.

2. Gently lift the top of the computer and support the top with the metal support rod found in the bottom front of the case.

3. Use an IC puller to remove the 6502 microprocessor from location UB14 on the main logic board. The location on the board is marked "UB14."

4. Insert the 40-pin DIP connector of the logic cable into UB14. Ensure that the connector is inserted so that the cable runs toward the front of the computer.

NOTE
Pin 8 of the logic-cable DIP connector (which connects to the UB14 6502 - microprocessor socket) is intentionally cut off.

5. Connect the short power cable to pin location J10 on the main logic board.

6. Connect the long power cable to pin location J11 on the main logic board.

7. Position the Expansion Memory Board in the CBM 8032 using the four mounting brackets on the Expansion Memory Board. See figure 2-1 for placement.

> **NOTE**
>
> The left-hand mounting brackets fit over the inside edge of the heatsink. The right-hand mounting brackets fit over the right-hand edge of the CBM 8032 case.
>
> The slot in the right-hand edge of the Expansion Memory Board must line up with the "boss" (or supporting rib) inside the top, right-hand edge of the case. Failure to properly locate the board may result in damage. Some cases may not have a boss.

Gently lower the top of the computer case, watching to make sure the boss fits in the slot on the right side of the Expansion Memory Board. When the board is properly situated, re-open the computer and support the top of the case.

8. Using a 4/40 Allen wrench, tighten the Allen screws on the mounting brackets. Be sure the Expansion Memory Board is securely mounted in the CBM 8032.

9. Close the cover of the CBM 8032 and replace the screws under the front cover.

10. Turn power on to the CBM 8032 and peripherals.

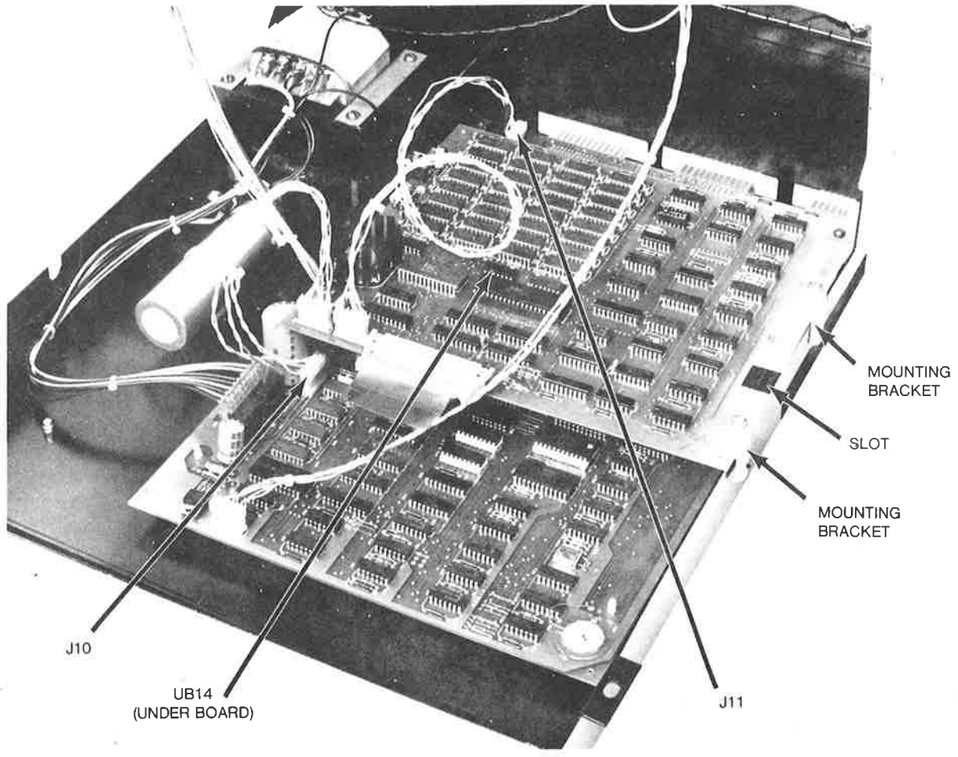


Figure 2-1. Expansion Memory Board Installation

Chapter III

OPERATION

INTRODUCTION

The following paragraphs describe the operation and programming of the Expansion Memory Board. Included in this chapter is a description of the memory control register and a sample program for handling interrupts.

Memory Mapping

Expansion memory can be mapped into main memory addresses $8000 through $FFFF. See Figure 3-1. Only two of the 16k expansion blocks can reside in main memory at *one time.* This provides an additional 32k bytes of memory to the user.

Selection of the expansion blocks is by bits 2 and 3 of the expansion memory control register. Each 16k block has a 16k alternate that can be selected by bits 2 and 3 of the control register.

Main memory addresses $8000 through $BFFF can only be mapped by expansion blocks 0 or 1. Main memory addresses $C000 through $FFFF can ony be mapped by expansion blocks 2 or 3.

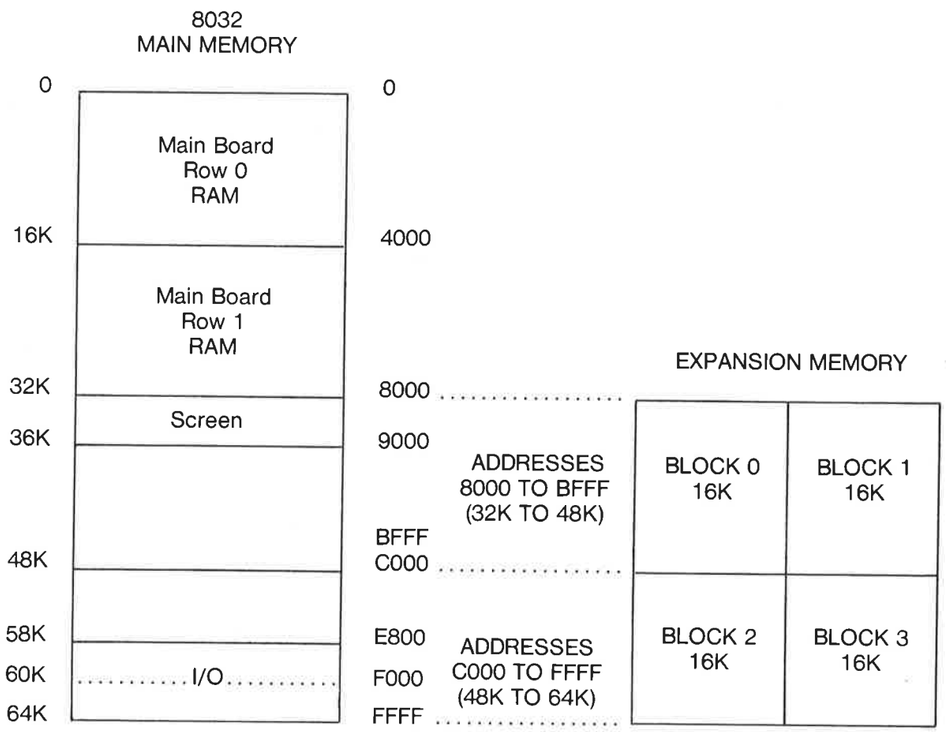


Figure 3-1. Expansion Memory Mapping

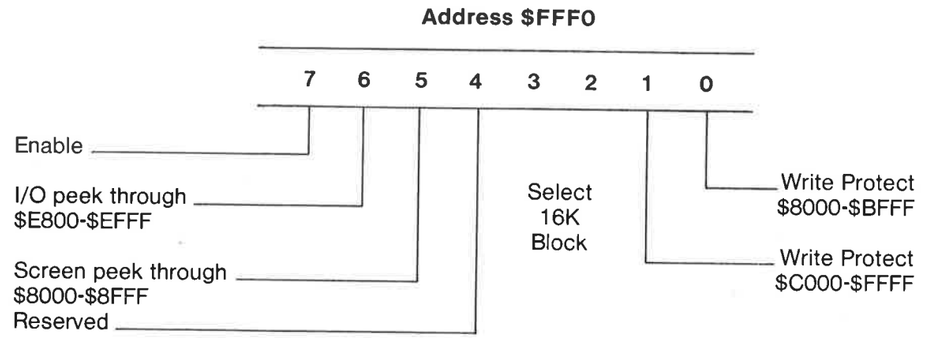
PROGRAMMING THE EXPANSION MEMORY BOARD

The user may choose to write his own programs to operate the Expansion Memory Board. A sample program is included in this chapter.

Control Register

Control of the expansion memory is through a memory control register on the Expansion Memory Board located at address $FFF0. The memory control register provides selection of 16k-byte blocks, write protection, enabling the expansion memory, I/O peek through and screen peek through. Because the memory control register is write only, a copy of the register should be kept in the lower 32k of main memory.

Figure 3-2 depicts the functions of the memory control register. The paragraphs following it describe these functions in detail.



0 0	2 and 0
0 1	2 and 1
1 0	3 and 0
1 1	3 and 1

Figure 3-2. Expansion Memory Control Register

Control Register Bit 7 - When equal to 1, enables the expansion memory. When bit 7 equal to 0, expansion memory is disabled. Bit 7 defaults to 0 on power up.

Control Register Bit 6 - When equal to 1, I/O peek through is enabled.

Control Register Bit 5 - When equal to 1, screen peek through is enabled.

Control Register Bit 4 - Reserved.

Control Register Bit 3 - When equal to 1, block 3 (16k-byte) is selected. When equal to 0, block 2 (16k-byte) is selected.

Control Register Bit 2 - When equal to 1, block 1 (16k-byte) is selected. When equal to 0, block 0 (16k-byte) is selected.

Control Register Bit 1 - When equal to 1, addresses $C000 through $FFFF on the *Expansion Memory Board only* are write protected. I/O is not write protected if I/O peek through is enabled. When equal to 0, the addresses are not write protected.

Control Register Bit 0 - When equal to 1, addresses $8000 through $BFFF on the *Expansion Memory Board only* are write protected. The screen is not write protected if screen peek through is enabled. When equal to 0, the addresses are not write protected.

I/O Handling

I/O in the CBM 8032 consists of the following five devices:

1. A 6520 PIA at $E810
2. A 6520 PIA at $E820
3. A 6522 PIA at $E840
4. A CRT controller at $E880
5. Screen memory at $8000 through $87FF

These five I/O devices may be accessed in two ways. The first way is to simply switch out the expansion memory and restore main memory. This may be already accomplished by the memory manager software when a CBM I/O subroutine is called.

The second way (necessary when a RAM-loaded application program accesses I/O) uses the I/O peek through feature. Bit 6 of the control register enables I/O peek through.

NOTE
When using the I/O peek through, ensure that the currently running subroutine does not reside over the top of the peek through.

Accessing screen memory is accomplished in the same way as accessing the other I/O devices. When accessed, screen memory is seen as 25 lines of 80 columns. The data is stored row-wise as sequential bytes.

The CRT display circuitry cannot directly display out of the expansion RAM. A suggested memory manager function is to page whole screen-fulls of data out of the expansion RAM.

Interrupt Processing

The 6502 microprocessor is designed for a simple system architecture in which the lower 32k of the address space is RAM and the upper 32k is ROM. This allows the microprocessor to fetch the starting address of the first instruction out of ROM upon reset. The result is that three hardware vectors are stored in addresses $FFFA - $FFFF.

The memory manager must accomplish the following functions

1. Ensure that there is a valid address at each ROM address in the two 16k expansion blocks that are active. The only exception is if interrupts are disabled by a SEI instruction executed before bit 7 is set to a 1.

2. To avoid being interrupted when changing a vector, execute a SEI.

3. The ROM interrupt vectors in the CBM point to routines in ROM which are not accessible when the expansion memory is selected. For that reason, the memory manager should a) set the vectors to point at a routine that switches to main memory mode, b) call the interrupt service routine, and c) restore the expansion memory mode.

Interrupt Handler Program

The following sample program (Refer to Table 3-1) handles interrupts and passes control to the ROM routines for processing. The global variable MEMMAP contains the current contents of the control register. The user program must first call the INIT routine to load the 6502 interrupt vectors into the expansion RAM. After that, interrupts are pre-processed by the subroutines NMI and IRQ. Interrupts are exited by RTIP

Table 3-1. Interrupt Handler Program

```
; INITIALIZE INTERRUPT PROCESS
;
; ASSUME MEMMAP SETUP
;
INIT    SEI
        LDA MEMMAP
        STA $FFF0
;
; INIT USER IRQ VECTOR
;
        LDA #<IRQ
        STA $FFFE
        LDA #>IRQ
        STA $FFFF

; INIT USER NMI VECTOR
;
        LDA #<NMI
        STA $FFFA
        LDA #>NMI STA $FFFB

        CLI
;
        RTS

; PROCESS IRQ
;
IRQ     STA TMPA                        PRESERVE .A
;
        PLA
        PHA
        STA TMPPS
;
        LDA #%00000000
        STA $FFF0
;
; PUSH RETURN FROM INTERRUPT ADDRESS
;
        LDA #>RTIP
        PHA
        LDA #<RTIP
        PHA
;
        LDA TMPPS
        PHA                             ; PUSH DUMMY STATUS
;
        LDA TMPA                        ; RESTORE .A
;
; GO TO ROM IRQ SERVICE
;
        JMP ($FFFE)
```


Table 3-1. Interrupt Handler Program (Continued)

```
; PROCESS NMI
;
; PROCESS NMI
;
NMI     STA TMPA                        ; PRESERVE .A
;
        PLA
        PHA
        STA TMPPS
;
;
        LDA #%00000000
        STA $FFF0
;
; PUSH RETURN FROM INTERRUPT ADDRESS
;
        LDA #>RTIP
        PHA
        LDA #<RTIP
        PHA
;
        LDA TMPPS
        PHA                                     PUSH DUMMY STATUS
;
        LDA TMPA
;
; GO TO ROM IRQ SERVICE
;
        JMP ($FFFA)
;
; RETURN FROM INTERRUPT PROCESS
;
RTIP    PHA
        ;
; MAP BACK TO ORIGINAL RAM
;
        LDA MEMMAP
        STA $FFF0
;
; RESTORE OLD .A
;
        PLA
;
; BACK TO USER
;
        RTI
        .END
        .LIB MONITOR
```

Commodore Business Machines, Inc.
3330 Scott Boulevard
Santa Clara, California 95051