CBM Expansion Memory Board

User's Guide

commodore

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.

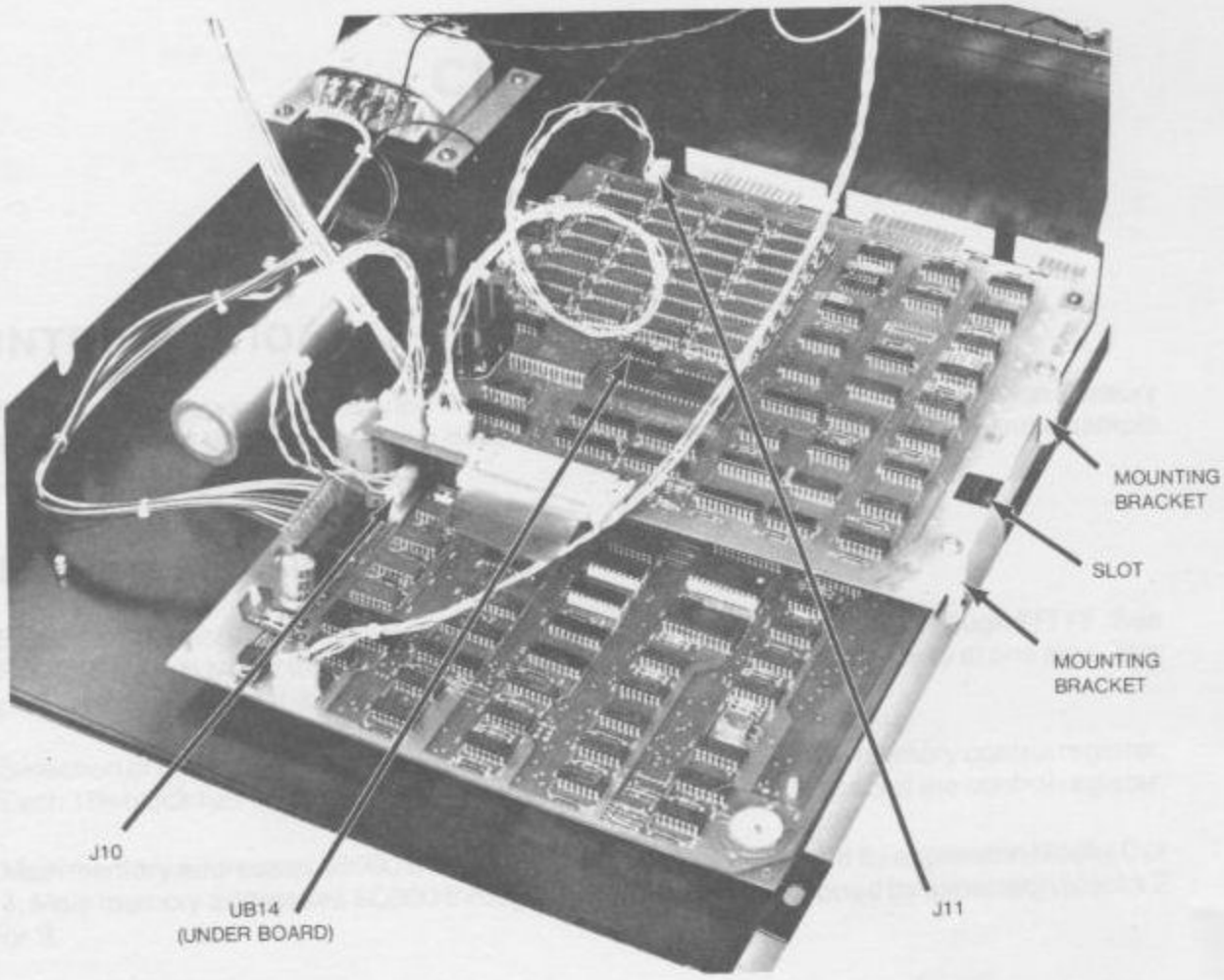


Figure 2-1. Expansion Memory Board Installation

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.

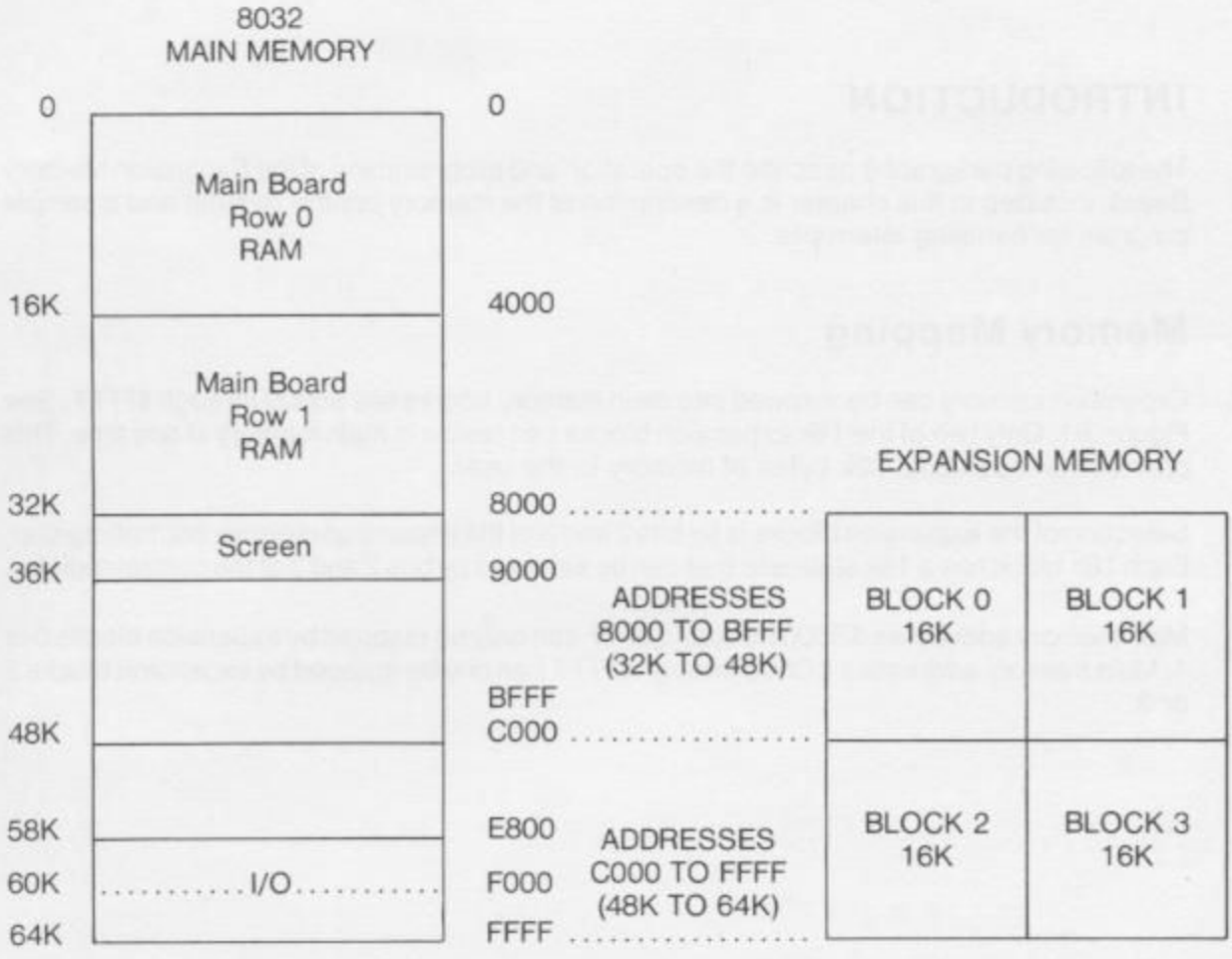


Figure 3-1. Expansion Memory Mapping

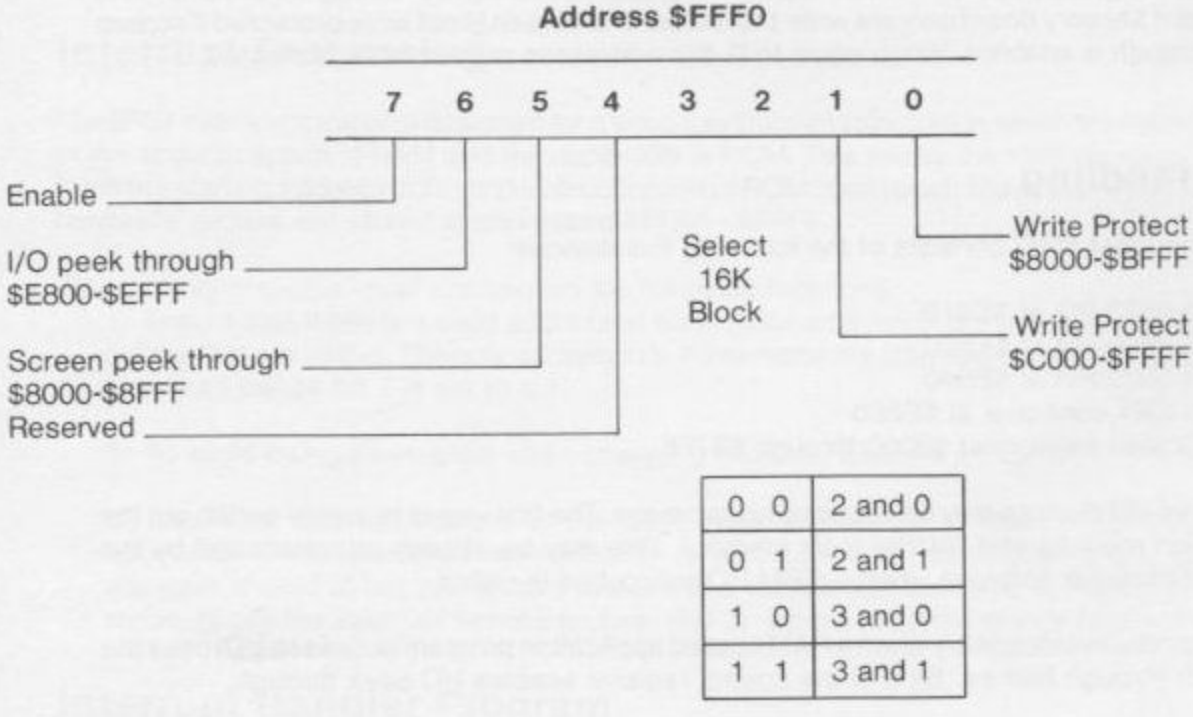
PROGRAMMING THE EXPANSION MEMORY BOARD

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.

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
```