

An Overview of the CBM 64k Memory Expansion Board

The expansion board adds 64k-bytes of RAM to the CBM 8032 or PET 4032 (with 12" screen), giving a total of 96k-bytes of RAM. The add-on memory is mapped into the 6502's main memory space in 16k-byte blocks, two at a time. A write-only control register at hex address \$fff0 provides for expansion board enable/disable, bank-selection of 16k-byte blocks, write-protect of expansion RAM and a 'peek-thru' feature for I/O and Screen memory.

A diskette containing programs for controlling the expansion memory is supplied with the board. These include: a loader which prompts the user and then loads one of three versions of CBM Basic into add-on memory, a functional equivalent of TIM which provides access to add-on memory and a program which extends Basic to provide commands for use of add-on memory from Basic programs. Also included are a set of short Basic programs which demonstrate use of the Basic command extensions.

The expansion board is disabled at power-up and the system is normal in all respects. When the Basic extension routines are loaded and started, access to the add-on RAM is enabled. The add-on RAM is configured as two 16k-byte blocks at hex \$8000-\$bfff and \$c000-\$ffff plus two alternate blocks. Either or both of the currently selected blocks can be write-protected via the control register.

Since the add-on memory is addressed contiguously from \$8000-\$ffff a means of accessing screen RAM (\$8000-\$8fff) and I/O memory (\$e800-\$efff) is required. By storing an appropriate bit pattern into the control register, memory accesses in these address ranges can be directed to either add-on memory or to the screen or I/O memory. Either of the active 16k-byte blocks may be write-protected separately and 'peek thru' for screen or I/O set separately.

The Basic command extensions reside at hex \$7800, leaving 29k-bytes of contiguous Basic program/data space in low memory. Commands are provided to allow loading additional program/data files into add-on memory and for 'down-loading' programs into low memory for execution.

Thus a Basic menu program and several frequently-used modules of an application can be loaded to add-on RAM for fast program chaining without re-loading from disk each time. Careful design of Basic applications, organizing source code into small modules can allow a combined program and data size of up to 70k-bytes.

Assembler language applications do not require the Basic Extension routines to be loaded. Thus Assembler programs may access all 96k-bytes of both low memory and add-on RAM as needed, the only problem being that the application programmer must handle bank selection as required.