16K RAM card for the Apple II

Build this Brake Lamp Flasher

Utility amplifier modules

Versatile electronic crossover

Diesel sound for model trains

Pioneer tape & reverberation amplifiers
16K RAM card for the Apple II

Add a 16K RAM card to your Apple II series computer. With the battery back-up feature it can be used to store often-used programs which don’t have to be loaded from disk every time you use them.

by DIETER KUENNE*

This static RAM card superimposes RAM over already existing ROM, namely Applesoft, integer Basic and the monitor. It may also be superimposed over a language card already in slot 0 (built into an Apple IIe). In the latter case a “supervisory program” or “memory management program” is required to control the cards. These are commercially available at low cost.

The card is designed to be compatible with Apple’s Language card in the running of Pascal, CP/M, DOS 3.3 and the current version of PRO-DOS when plugged into slot 0 of an Apple II/II+.

Placed into any other slot (seven only on certain revisions of Apple’s), it has available a 256-byte page of RAM to be used as a “peripheral driver” and thus can be used as a “DOS-less” bootstrap at start-up. It may also be used in conjunction with any of the available operating systems such as Pascal, DOS or CP/M as a peripheral such as a pseudodisk.

The optional battery back-up facility is very handy if you wish to operate without a disk in say, a control application. In this case, a program could be loaded via the cassette port or serially via a modem to start and then kept in RAM permanently. The boot page is used here as a “pretend disk”.

Readers may at first think that 16K is not very much but if you had to load it every time from cassette (or very unrealistically, key it in) then it is quite a lot. Any one with a Microbee will tell you how pleasant it is to have battery backed up RAM. Once you have it, you won’t want to be without it. Now let’s have a look at the design and operation.

Design and operation

This section should be read in conjunction with the accompanying tables and circuit diagram. The object of this card is to be able to hold its data when the Apple is turned off and be able to look like ROM. Further it should be compatible with the Apple language card to satisfy PRO-DOS and all the current operating systems available for the Apple II series of computers when plugged into slot 0. In the back-up mode, the data would be retained by a battery or large value capacitor for a period of time between uses.

Decode logic

Let us assume that we are using an Apple II+ or II, that the card is plugged into slot 0 and we have just turned on. At this point, all D flipflops of IC5 are reset because the reset line (pin 1 of IC5) is held low by C5 until after full voltage is achieved by the 5V supply. This timeconstant is set by R7 and C5. D1 serves only to discharge C5 more quickly when power is removed. The Q output of FF5a

*5 Waratah Avenue, Bayswater 3153.
16K RAM card for the Apple II

is the “read enable” and the Q output of FF5b is the “write enable” output and whilst both are low, the main 16K of RAM on this card cannot be accessed. The enable output “EN” of gate 3a is logic 1 because “REN” and “WEN” are logic 0. This inhibits address decoding by gates 1a, 2a and 2b. When the output of gate 3a is low, then gates 1a, 2a, and 2b decode address lines A15 to A12 according to Boolean equation (1) where “VMA” (valid memory address) is the output of gate 2b.

\[ VMA = \overline{(A15 \cdot A14) + (A13 + A12) + \overline{EN}} \quad \ldots \ldots (1) \]

where “EN” is derived from the read/write line and the soft-switches read enable and write enable. As can be seen from equation (1), “VMA” is only true if the card is either read or write-enabled and the address range is between $0D00$ and $0FFF$ (hex).

As the card can be read-enabled and write-protected, read and write-enabled, write-enabled and read disabled or totally disabled, “EN” is obtained from logic equation (2). Both “WEN” and “REN” refer to read-enable and write-enable card signals on the card. Their derivation will be described in the section on “soft-switch control logic” later on. In addition, “W” and “R” are “read” and “write” signals taken from the buffered CPU R/W line.

\[ EN = \overline{(WEN \cdot W) + (REN \cdot R)} \quad \ldots \ldots (2) \]

This is implemented by gates 1a, 6a and 3a. Readers should note that gates 6a and 6d are not logically needed but are included to comply with the loading rules of the Apple bus read/write and phase 0 lines. As well, they improve the timing of data transfer between CPU and memory as the timing diagrams show (see Fig. 1 and Fig. 2).

The card has only a valid address range of $0D00$ to $0FFF$, so we need to

---

**APPENDIX**

**Table 1: Hardware switches**

<table>
<thead>
<tr>
<th>SW1</th>
<th>SW2</th>
<th>SW3</th>
<th>SW4</th>
<th>EFFECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>Normal operation (Apple protocol).</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>Hold BANK 1.</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>Write protect (on start or after protect command).</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>Disable R/W line decoding.</td>
</tr>
</tbody>
</table>

Note: only SW2 OR SW3 should be made and not both.

**Table 2: Base addresses for control locations**

<table>
<thead>
<tr>
<th>BASE ADDR</th>
<th>R/W</th>
<th>A3</th>
<th>A2</th>
<th>A1</th>
<th>A0</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$080$</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>RAM read BANK 0, write protect.</td>
</tr>
<tr>
<td>$081$</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>ROM read BANK 0, write enable after 2nd read.</td>
</tr>
<tr>
<td>$082$</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>ROM read BANK 0, write protect.</td>
</tr>
<tr>
<td>$083$</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>RAM read BANK 0, write enable after 2nd read.</td>
</tr>
<tr>
<td>$088$</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>RAM read BANK 1, write protect.</td>
</tr>
<tr>
<td>$089$</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>ROM read BANK 1, write enable after 2nd read.</td>
</tr>
<tr>
<td>$08A$</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>ROM read BANK 1, write protect.</td>
</tr>
<tr>
<td>$08B$</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>RAM read BANK 1, write enable after 2nd read.</td>
</tr>
</tbody>
</table>

**Where to buy the PCB**

The author of this article, W.D. Kuenne, can supply the PCB. This will have plated through holes, rollsweld on the copper pattern and gold-plated edge connector. Price will be $65 including packing and postage.

The author will also provide a service to readers who cannot make their RAM card function properly. This will only be available to purchasers of the above PCB and will cost $25 plus parts and return postage.

In addition, the author will answer letters on the RAM card provided that they are accompanied by a self-addressed and stamped envelope. Address all correspondence to W.D. Kuenne, 5 Waratah Avenue, Bayswater, Victoria 3153.
Table 3: Address decoding table

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>A15</th>
<th>A14</th>
<th>A13</th>
<th>A12</th>
<th>A11</th>
<th>MOTHER</th>
<th>METHOD 1</th>
<th>METHOD 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F800</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>ROM</td>
<td>R0</td>
<td>R0</td>
</tr>
<tr>
<td>$F000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ROM</td>
<td>R1</td>
<td>R1</td>
</tr>
<tr>
<td>$E800</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>ROM</td>
<td>R2</td>
<td>R2</td>
</tr>
<tr>
<td>$E000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ROM</td>
<td>R3</td>
<td>R3</td>
</tr>
<tr>
<td>$D800</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ROM</td>
<td>R4</td>
<td>R4</td>
</tr>
<tr>
<td>$D000</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>ROM</td>
<td>R5</td>
<td>R5</td>
</tr>
<tr>
<td>$C800</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>I/O</td>
<td>ROM</td>
<td>ROM</td>
</tr>
<tr>
<td>$C000</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>I/O</td>
<td>ROM</td>
<td>ROM</td>
</tr>
</tbody>
</table>

NOTE: R0 to R7 are the RAM device number for linear decode. (R6) and (R7) are the would be locations without BANK SWITCHING.
Method 1 is using AND and OR gates to BANK select.
Method 2 is using XNOR and NOR gates to BANK select.

bank-switch another 4K over the lower range, namely $D000 to $DFFF. This is achieved by the soft-switch “Bank” (FF5d) and gates 2c and 4a. These gates form a “funny” address line given by equation (3).

\[ A'12 = (A13 + BANK) + A12 \ldots (3) \]

For now, assume that the card is enabled for reading and writing. When the CPU reads or writes from a range of memory at $D000 and above, “VMA” will then be a logic 1. At this time the motherboard ROMs are disabled by the inversion of VMA from gate 2b by gate 4b (open-collector) pulling the INH line of the Apple bus low during CPU phase 1 and staying low for all of phase 2. The Apple IIe requires this line to be pulled low during phase 1.

Address line A13 selects one half of the dual demultiplexer IC7 (a 74LS156). Which half is determined by whether the address is above or below $E000. During phase 0, the output of gate 8c fails to actually enable one of the demultiplexers. Address lines A11 and funny address line A12 then determine which one of the four outputs of the selected demultiplexer goes low to select a RAM chip.

Each chip-select line has a 2K range of address. Now depending on whether it was a read or write, either gate 8a or gate 8b, being the write or read control of the RAM devices, becomes active low. The read or write control lines fall at, or before, the chip-select lines so that in the event of a write, the RAM’s output buffers are Tri-state, ready for data to be written to it when the bus-transceiver is enabled. Further, if it was a read, gate 8b is low as soon as the VMA line is high and stays low until well into phase 1 again, so that DATA hold time after CPU phase 2 is achieved.

The bus-transceiver type 74ALS245 is enabled two gate delays after the 74LS156 is selected and is released the same time after the 74LS156 is deselected. These gate delays are required because of a hold-time requirement of the 6502. This is 10ns after end of the real phase 2 for reading from RAM or peripheral and the advanced phase 0 from the Apple bus. This is achieved by holding “RE” low independently by CPU address and R/W line decoding.

Writing to RAM is not as critical as the data is valid from the 6502 at least 200ns before end of phase 2. The RAM write cycle ends before the end of the real phase 2.

One further RAM device is provided and this resides in the slot ROM page. The I/O select signal goes low during phase 0 time when the CPU is addressing $CNXX space, where N is the slot of the card. On slot 0 this is not provided, and resistor R14 pulls this line high to disable the RAM chip via gate 4c (an open-collector device). R21 pulls the CS line high during power-down time so that it also is saved along with the rest of the 16K on the card. No further decoding is provided as each slot selects a different page of RAM. This allows us to write slot-independent software.

The $C800 to $CFFF space was thought of but due to the number of additional chips required to support it, it was decided against using it as well.

Soft-switch control logic

For slot 0 there are 16 addresses between $C080 to $C0FF allocated. All the soft-switches are toggled by these addresses.

In the address range $C080 to $C0FF (or $F0 for other slots where s = slot) the following results occur:
(1) If A0 is logic 0 then BANK = 0.
(2) If A0 is logic 0 then BANK = 1.
(3) If exclusive-NOR of A0 and A1 is logic 1 then the card becomes read-enabled.
(4) If the card is accessed twice with A0 at logic 1 then the card becomes write-enabled (reads).

In this address range, DEV goes low during phase 0 and rises at the end of the phase 0 cycle. To toggle a soft-switch, it is only necessary to read from a control location. The design truth tables and their Boolean equations are in the Appendix.

To read-enable the card, one simply reads or writes to a control address where A0 and A1 are the same level, or where A0 exclusive-NOR A1 is true. These are at $C080, $C083, $C088 and $C08B for slot 0. When, for example, $C080 is accessed, A0 and A1 are both logic 0 and the exclusive-NOR gate 4d outputs a logic 1. As phase 0 begins, DEV goes low and rises again to a 1 at the end of phase 0. On the rising edge of DEV, the output of gate 4d (logic 1) is clocked into FF5a. The card is now read-enabled via the REN signal (Q output of FF5a). The Q drives the LED indicator to show the user that the card is read-enabled. R2 is a pull up resistor because IC4 is an open-collector device.

\[ \text{REN (tn+1)} = (A0 + A1) \cdot (\text{tn}) \]
16K RAM card for the Apple II

SC080 to SC08F, the second of which must be read. On the first access with A0, logic 1 sets FF5c on the rising edge of DEV. FF5b stays at logic 0 because both WEN and Q of FF5c (pin 10) are logic 0 before the rising edge of DEV.

PARTS LIST

1 double-sided PC board (see text)
1 4-way DIP switch

Semiconductors
9 HM6116LP-3 or HM6116P-3 16-bit RAM (Hitachi). (Do not use SRM2116200)
1 74LS00 quad 2-input NAND gate IC
1 74LS27 triple 3-input NOR gate IC
1 74LS51 dual AND-OR-INVERT gate IC
1 74LS266 quad exclusive-OR gate IC
1 74LS175 quad D-type flipflop IC
1 74LS08 quad 2-input AND gate IC
1 74LS156 dual demultiplexer IC
1 74LS12 triple 3-input NAND gate IC
1 74ALS245 or 74LS245 octal Tri-state buffer IC
2 2N3645 PNP transistors (plastic type preferred)
1 PN2222A NPN transistor (plastic type preferred)
1 1N914, 1N4148 small signal silicon diode
1 OA90, 91 germanium diode
1 1N4001 silicon rectifier diode
1 1N4729A 3.6V zener diode
3 red LEDs

Capacitors
1 470μF/16VW electrolytic (see text)
1 47μF/10VW PC electrolytic
1 4.7μF/35VW tantalum electrolytic
12 0.1μF/50VW monolithic
1 220pF/50VW ceramic

Resistor arrays
1 Resistive inline resistor array, 9 x 100kΩ (9 x 104Ω)
1 Resistive inline resistor array, 8 x 1kΩ (8 x 102Ω)
1 Resistive inline resistor array, 8 x 470Ω (1N x 471Ω)

Resistors (1/4W, 5%)
2 x 100kΩ, 2 x 1Ω, 3 x 820Ω, 2
x 680Ω, 1 x 470Ω, 4 x 390Ω, 3
x 100Ω, 1 x 39Ω.

Note: Unless specified do not substitute ALS types for the above ICs.
Table 6: Write enable truth table

<table>
<thead>
<tr>
<th>W (tn+1)</th>
<th>F (tn+1)</th>
<th>F (tn)</th>
<th>A0</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>R/W</th>
<th>W (tn)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

F (tn+1) = (tn)  
W (tn+1) = [F (tn) . R/W (tn) . A0 (tn)] + [A0 (tn) . W (tn)]  
where F is the first access latch and W is the write enable control signal.  

Note: this is different from standard card but is transparent to all operating systems to date.

This is given by equations 5 and 6 below:

\[ D5b = (R \cdot A0 \cdot Q5c) + (WEN \cdot A0) \]  
\[ Q5b (tn+1) = D5b (tn) \]  

Battery backup facility

This is accomplished by means of Q1, Q2, Q3, R13, R16 to R20, ZD1 and D2 to D4. It functions as follows:

When the Apple is turned on, D2 begins to conduct at about 0.7 volts on a totally flat battery and so tends to bring the RAM supply to within about 0.65 volts of the Apple's 5 volt supply. However, as the voltage rises above 3.8 volts, as set by the zener diode ZD1 and the divider R16 and R17, Q2 begins to conduct.

As the voltage across R19 and R20 rises, Q3 also begins to conduct and is saturated before the voltage at PU (collector of Q2) is 2.4 volts (TTL level 1). This drives Q1 into saturation so that the RAM now has a 4.8 volt supply which is within its operating range. Without this transistor, the supply to the RAM would be below the 4.5 volt minimum recommended for operation of the devices. As the voltage rises above 4.5 volts, PU reaches TTL level 1 and so turns on the gates of IC8. This then enables us to read and write to the RAM assuming of course we enable it first by accessing the control locations mentioned earlier.

On power-down, the sequence is reversed such that the logic level 1 to the gates of IC8 is removed when Vcc goes below 4.5 volts so that we can't access it any more. This holds true for all locations of RAM on this card. In the back up state, supply is via diode D3 and only low current is required to keep the data in the RAMs. In this mode, Q1, D4 and D2 are all reverse-biased and prevent power from the battery going back to the Apple circuitry.

Note that there is no resistor between base and emitter of Q1. This is to avoid the possibility of discharging the battery via the collector-base junction and the base-emitter resistor.

During normal operation, the battery is charged via R13 and D4. The diodes for battery charging and supply during backup are selected for low forward voltage at low current drain (type OA90 etc). Also all the CS lines to the RAM are from open-collector gates so that their voltage rises to within at least 0.2 volt of their supply. This together with tying the address lines to ground via 100kΩ resistors minimises the total drain on the backup supply.

The backup supply would normally take the form of three or four nickel-cadmium cells connected in series and mounted on either side of the Apple main board and positioned so that any leakage will not cause damage.

As an alternative, a one Farad supercap, made by NEC and distributed by Soanar Electronics Pty Ltd (30 Lexton Road, Box Hill 3128) may be considered for the backup supply.

The prototype card is shown assembled with a 47uF capacitor as a backup supply. This gave a backup time of about four minutes. On this basis, a 1F (yes, one Farad) supercap would give a backup time of five to six days.

Construction

Before construction begins, it is advisable to read through this whole section.
section, making notes if you decide that your card should be assembled differently. The method described below is the way the original was built and it has proved to be the most reliable. It should result in a product of quality and reliability which approaches that of most reputable manufacturers.

Construction of the circuit involves the mounting of all the components onto the double-sided printed circuit board. Firstly all the sockets for the RAM chips should be soldered in place. All the remaining LS devices were directly soldered into the board as this maximises reliability. It is therefore desirable to use new ICs throughout. Next, all remaining LS devices should be tacked in place by two of their leads with solder. A final check should now be made that they are definitely in their correct place and orientated correctly. If OK, they can be fully soldered in place. The resistors and capacitors, diodes, transistors, LEDs and the dip-switch may now be soldered in place. Do not install the RAM at this stage.

Having soldered all components into place, it is a good idea to have one final check to see that all components are correctly installed.

All OK? Then the card may be checked out with the help of program listing 1. Also required is a backup copy of the DOS 3.3 master disk with the integer Basic on it and Apple Cillin II or other test program if you have one (Don't worry if you don't).

The procedure is as follows:

(1) Turn the Apple off (important) and remove its lid.
(2) Take out all plug-in cards for the first switch on.
(3) Plug the partially constructed card into slot 0 (the card should not have any RAM).

(4) Connect a video monitor as usual and switch the system on and set the card switches to normal.
(5) The system should display "Apple II" and the blinking cursor as a DOS-less system. If it hangs, there is a fault and everything should be checked and corrected.
(6) All being well, key in the short machine routine of listing 1 and RUN.
(7) The keys pressed will then exercise the soft-switches and the LED indicators should show their settings.
(8) Press <ESC> to end.
(9) Turn the Apple off and remove card.
(10) If all was well then the RAM may now be installed in the sockets and the card plugged into slot 0.

Install the disk controller card and boot up the system back up. The card should load the integer basic and when it finishes, type "INT <CR>" You should have a ">" prompt followed by a flashing cursor to show all is well.

(12) Key in a short program as in listing 2 and let it run for a while. The program should not bomb out nor should funny characters be placed onto the screen. If this program does go strange after a while, then there is a timing fault. If it bombs first go then there is probably a faulty RAM chip or other hardware fault. The use of Apple Cillin II (or other diagnostic program) to test your board fully is highly recommended.

Note: On boards that have problems reading RAM E8 (IC11), a supply lead pair should be connected between C4 (close to edge-connector) and the 74LS245 supply pins. This condition should not be confused with errors occurring all over the RAM range.