



48K RAM CARD

INTERFACE FOR THE 6502

P. G. J. de Beer

After tinkering for some time with 4K RAM cards and the accompanying tangle of wires and connections, my eye was caught by the 48K RAM card for the 1802 published in the March 1982 and following issues of RB. This offered the chance to solve the memory problem once and for all, so the possibility of adapting this card to the 6502 processor was studied. By leaving a number of ICs off the original board and adding a small interface in the space freed up, you get 48K of RAM for your 6502 system – a real wealth.

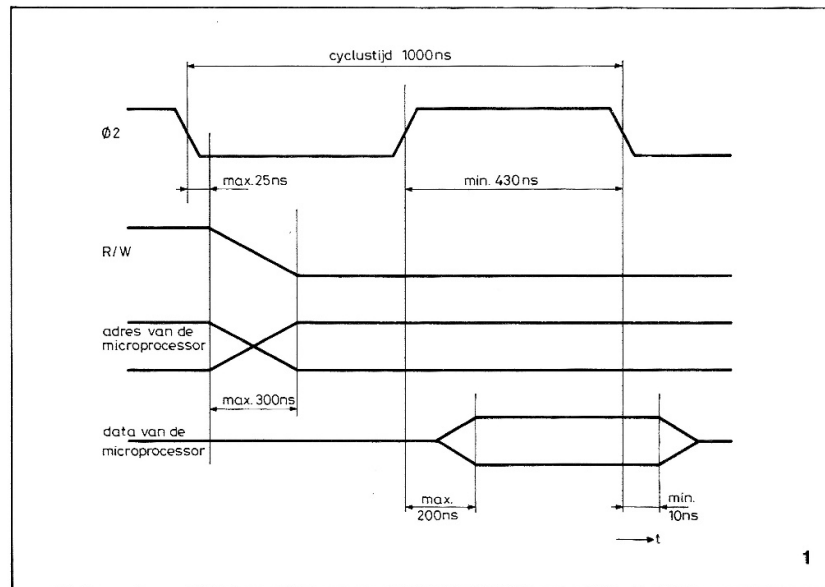


Fig. 1 Overview of the signals in a 6502 system.

Design goals

The following requirements were set for the adaptation:

1. The circuit must not slow down the 6502; in other words, the microprocessor must

notice nothing of the dynamic nature (refresh) of the memory.

2. The interface must use as few processor signals as possible.

3. The interface must be as universal as possible.

4. The 48K memory space must be placeable anywhere desired within the total memory space.

5. The rest of the computer system must remain unchanged.

This design more or less meets all these wishes.

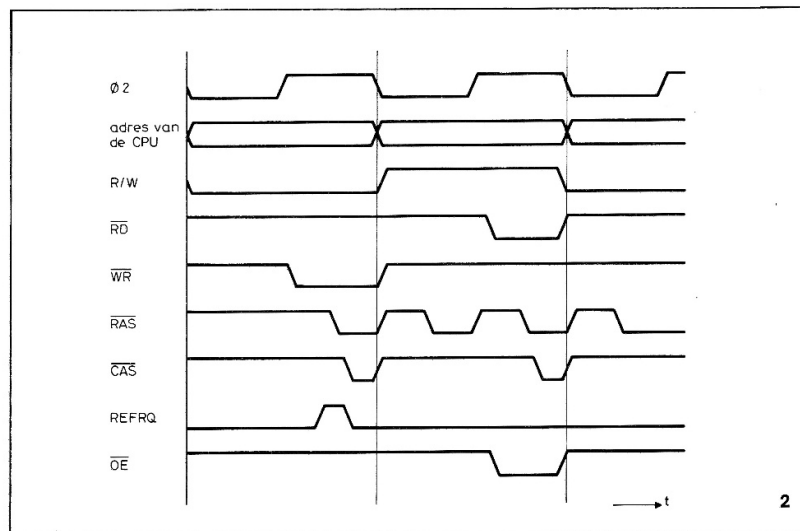


Fig. 2 Overview of the signals during a hidden refresh.

Hidden refresh

The first requirement for the RAM card is that it must not place any extra load on the processor. Yet a dynamic memory must be refreshed periodically. To avoid demanding extra time from the processor for this, a so-called "hidden refresh" is used. As fig. 1 shows, the 6502 uses the data bus only during the last half of the clock cycle. This gives the dynamic RAM controller the opportunity to refresh the memory during the first half. For the operation of the dynamic RAM controller and the RAMs themselves, see the series of articles by Mr H. B. Stuurman in RB, March, April and May 1982. To fit two memory accesses into one processor cycle, a complete access must be completed within 500 ns. The clock frequency of the RAM controller has therefore been raised to the maximum guaranteed frequency of 25 MHz. A controller memory cycle then takes at most 480 ns. This makes it possible to carry out both a normal read or write and a refresh within one processor cycle. The memory could thus also be refreshed on every read or write, but this would cause unnecessarily high dissipation in both the controller and the memory ICs. The maximum interval between two refreshes of a 4116 memory cell is 2 ms. Since refresh is done per row and there are 128 rows, the maximum

repetition time between refresh actions is 15.625 μ s. With a processor cycle time of 1 μ s, refreshing every cycle would be fifteen times more than strictly necessary. A divide-by-ten has therefore been chosen, so that a row is refreshed every 10 μ s. An overview of the waveforms is given in fig. 2.

Control signals

An intermediate solution for less frequent refreshing was to use the Sync signal from the 6502. Every time the next instruction is fetched from memory, the Sync output goes high for one full processor cycle. Between two Sync pulses there are a number of processor cycles, varying from 1 to 12. The refresh rate thus becomes lower, but not enough. Moreover, an extra signal line from the processor is needed, and the 6800 and 6809, for example, have no Sync output. The solution finally chosen needs only the processor clock and Read/Write. These signals are available on many processors.

Interface

The interface has been kept as universal as possible. No monostable multivibrators or the like are used. Synchronisation is via the control signals RD-not, WR-not, PCS-not and REFRQ. The memory access time is fixed by the controller's 25 MHz clock frequency and is suitable for 4116s of 200 ns and faster. The latches already present on the original board hold the data on the data bus until the processor has read it.

Memory allocation

Most 8-bit processors have a 64K address space. The 48K on the RAM card can be placed in it in various ways. Some systems have the ROM with the monitor program at the top of the memory area, others at the bottom. Since the RAM area must be allocatable in different ways, the start address can be set in steps of 1K [translator's note: as printed; Table 1 actually gives 4K steps] from address \$0000 up to and including \$4000. This is selected by three wire links (A, B and C) on the interface board.

System design

The interface has two signal lines that may be of importance to a microprocessor system: the "Decode Enable" line (DE) and the "Bank Select" line (BS). The DE-not signal indicates when the RAM card is actually being addressed. This line can be used to disable other memory areas such as the monitor ROM, system RAM and peripherals. The BS-not line can be used to disable the RAM card itself. It is thus possible to fit several memory banks in the same logical address range, and choose between them by means of the BS-not lines. With these two signals the RAM card can be added to any system without drastic changes to it.

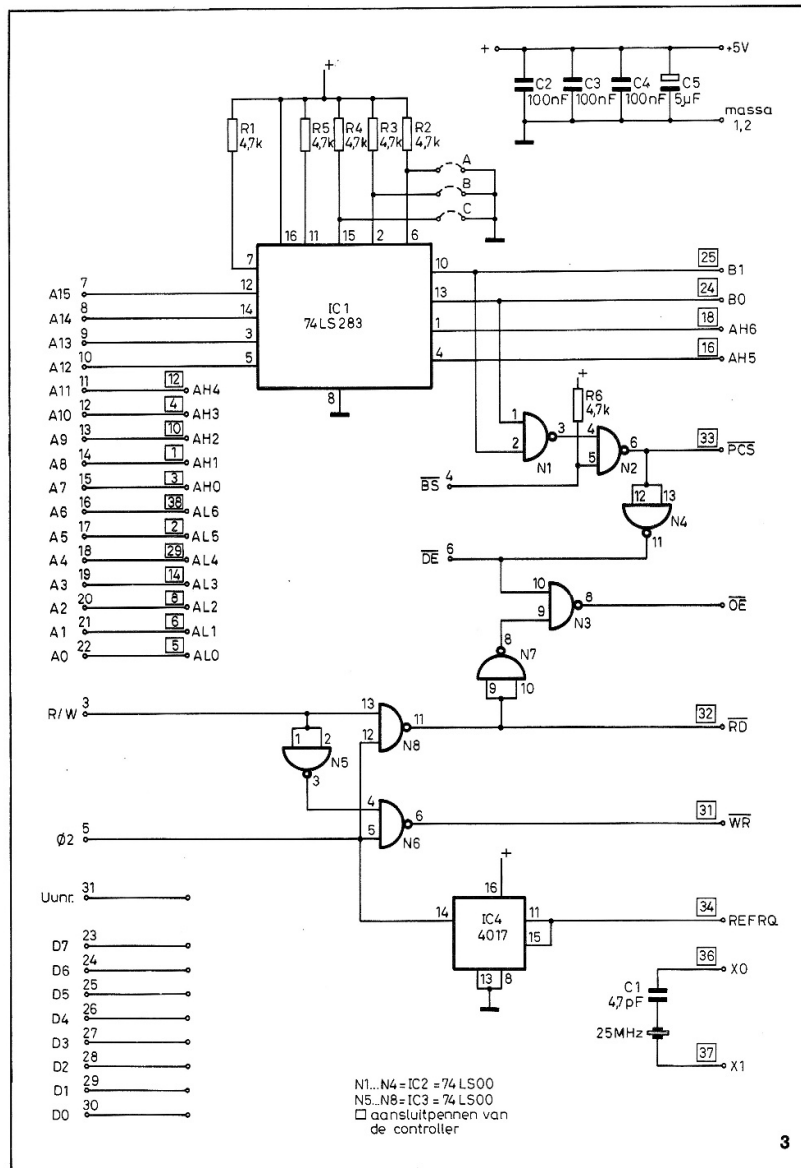


Fig. 3 Schematic of the interface for the 48K RAM card.

Circuit

Following the schematic of fig. 3, the interface consists of an address decoding and selection section (IC1 and IC2), a read/write section (IC3) and a refresh section (IC4). IC1 (74LS283) is an adder IC. Address lines A12 to A15 are connected to its A inputs, and a selectable fixed binary value to its B inputs. The carry input (pin 7) is connected to the positive supply via a resistor. This immediately suggests the logical setup for subtracting two binary numbers – and that is exactly what it is. One of the requirements was flexible placement of the 48K in the available space. The addresses within the RAM card itself are of course fixed. A certain value must be added to or subtracted from the address presented by the processor to obtain the correct logical address. Suppose you want the RAM card to start at address \$2000. When the processor presents address

\$2000, address \$0000 within the RAM card must be accessed. So a fixed value of \$2000 must be subtracted from all addresses. This can be done by adding a negative number. The two's complement of a number equals the inverse of that number plus one. The "plus one" is achieved by making the carry input high. The inverse of the number is obtained by tying inputs to ground as required, while they are normally high. The board has only three wire links instead of four, since the meaningful combinations range from 0 to 4 (see the start addresses). The outputs of IC1 form the two address lines AH5 and AH6 and the bank-select lines B0 and B1 for the controller. Although the controller can manage a 64K memory area, only 48K is provided, i.e. the lower three banks of 16K each. Only addresses in this range should select the controller. In other cases the controller must take no action, since other memories or peripherals are probably being selected. N1 and N2 produce a select signal that meets these requirements. The second input of N2 can be used to prevent selection anyway, which disables the RAM card. The inverse of PCS-not is DE-not. This output serves to disable any other decoding logic in the system when the RAM card is selected. The data bus latches are controlled by the OE-not signal, which is active only when the RAM card is selected for a read. The controller has separate Read and Write inputs, while the 6502 has a combined R/W output. Since reads and writes for the RAM card can only be handled in the second half of the processor's clock cycle, it is not difficult to generate separate Read and Write signals; N5, N6 and N8 take care of this. IC4 generates a refresh action every tenth processor cycle. A CMOS IC was chosen because it is slow. The incoming Ø2 clock pulses make each successive output go high for one full processor cycle. Output 10 is connected to the counter's reset input. When it is this output's turn to go high, the counter is immediately reset. Because of the rather large propagation delay of CMOS, a positive pulse of about 150 ns still appears on this output. Moreover, this pulse appears at the REFRQ input about 200 ns after the Ø2 transition, which prevents problems with the "arbiter".

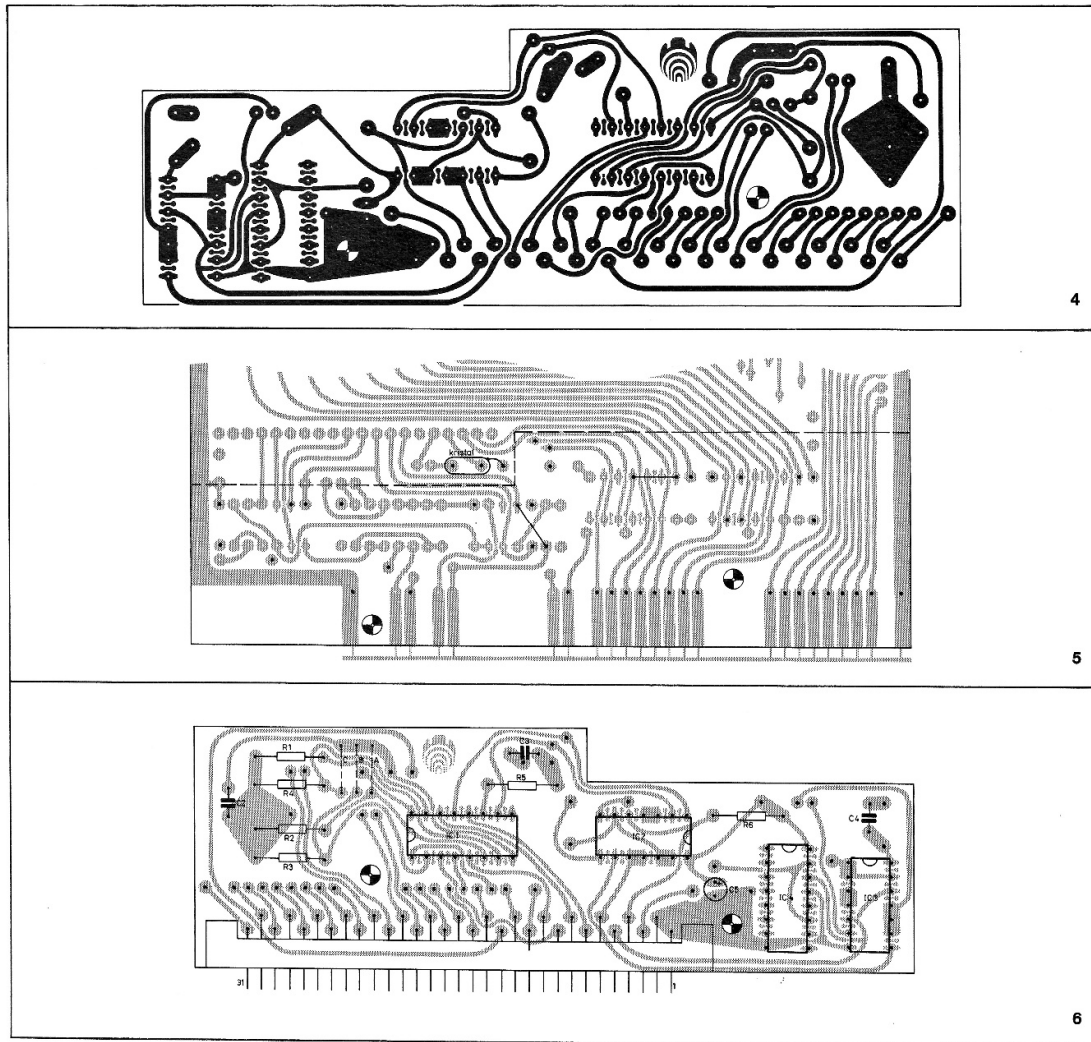
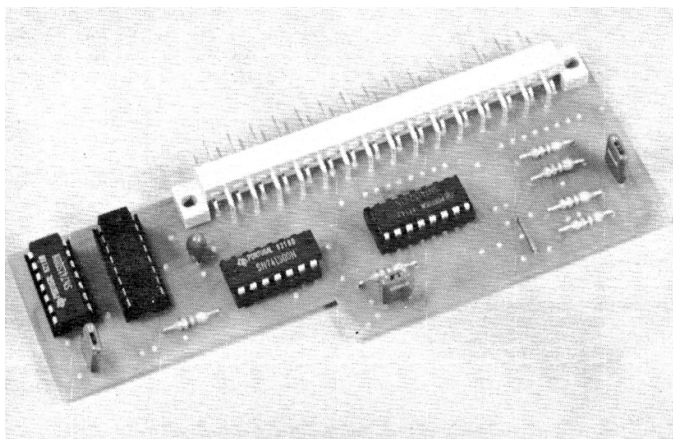


Fig. 4 PCB layout of the interface card, scale 1:1.



Construction

For the actual implementation of the interface a small PCB was designed, shown in fig. 4. The interface board and the original RAM card are closely linked: the size of the interface was chosen so that it fits exactly over the lower part of the RAM card. A special construction joins the interface board and the RAM card

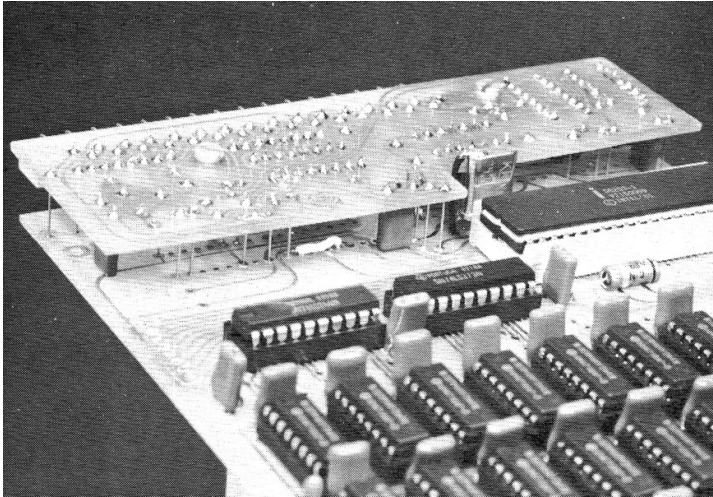
into one compact unit. First the RAM card is built according to the relevant article series. Before starting construction, two 3 mm holes are drilled in that board, as

indicated in fig. 5. IC4, IC5, IC6, IC7 and IC8, as well as C13, C15, C17 and D1, are omitted. The controller is fitted last. This may be an 8202A, but also an "8203 version 1". The latter is a newer version made suitable for 64K RAM ICs, but it can be used in place of the 8202A without extra provisions. The relevant select pin must be high to configure the 8203-1 for 16K RAM ICs such as the 4116. Next the two wire links are fitted and the 25 MHz crystal is placed. This must be a type with a fundamental frequency of 25 MHz and must not be a so-called "overtone" (third harmonic) type. The case is connected to ground with a short wire. Then work on the interface can begin. The components are mounted as shown in fig. 6: first the resistors, then the sockets and the connector, and finally the capacitors. Fitting wire links A, B and/or C selects the desired start address of the RAM card; see Table 1. After inserting the ICs (mind the orientation), the board looks as shown in fig. 7. Using two spacers about 12 mm long, the interface board is mounted component side down on the RAM card with two small bolts. If all went well, all remaining open holes in the interface board should now sit exactly above holes in the RAM card. Bare pieces of wire are now inserted into all marked holes (fig. 6) of the interface board, so that they protrude from the underside of the RAM card. Both sides are soldered. The result is shown in fig. 8.

A remark about the power supply connection: if you use an unregulated voltage for the RAM card, it can be supplied via pin 31 of the interface connector. If, however, you already have a regulated +12 V available, it can be supplied via the same pin, but then the 12 V regulator on the RAM card is replaced by a link from in to out.

Table 1 – Wire link settings for the desired start addresses.

Start address	Wire link		
	A	B	C
\$0000	no	no	no
\$1000	yes	no	no
\$2000	no	yes	no
\$3000	yes	yes	no
\$4000	no	no	yes



Connecting

The RAM card's connections now run via the 31-pin connector on the interface board. All address and data lines, as well as R/W and $\emptyset 2$, must be connected to the processor card. The RAM card is fully buffered and represents only a light load on the processor system. The supply line and a sturdy

ground wire are connected. The DE-not output of the interface board is connected to the select input of the rest of the system. The BS-not input can be connected to a PIA output, so that the RAM card can be switched on and off by software. After power-up, the LED on the RAM card should light, indicating that the 5 V supply is present. By writing to and reading from memory locations within the RAM card's range, you can check that everything works properly. Finally, a memory test lasting several hours is advisable to be sure of correct operation in the longer term as well.