

# MODIFYING THE VZ200 16K EXPANSION MODULE FOR THE VZ300

Steve Olney

This article describes a method of remapping a DSE VZ200 16K RAM expansion module preventing overlap of memory space when used on a VZ300. The cost is limited to the price of one integrated circuit chip plus a single-pole double-throw switch if dual VZ200/300 compatibility is desired. The modification is fitted inside the expansion module case.

MANY OF YOU who have updated to the new version VZ300 must be disappointed to realise that although the VZ300 comes with much more internal RAM as standard (18K as against 8K for the VZ200), use of your old VZ200 16K expansion module on the VZ300 only results in the same total memory as that which was available on the older VZ200 with the expansion module plugged in.

The reason for this becomes clear when a comparison is made between the memory maps of the VZ200 and the VZ300 as shown in Figure 1. If a VZ200 16K expansion module is plugged into a VZ300, about 10K of the expansion RAM overlaps memory space already provided to the VZ300 internally. This results in only 6144 bytes of extra memory. In order to make proper use of the expansion memory space, the start of the

VZ200 expansion module needs to be moved or remapped to the end of the VZ300 internal memory instead of somewhere in the middle. For more details on the memory map of the VZ200 and VZ300, refer to Jim Rowe's informative article on the VZ300, ETI July 1985.

The object of this article is to provide information sufficient to modify a VZ200 16K expansion module to be used on both your VZ200 as well as your new VZ300.

Before proceeding there are a few words of advice for those wishing to undertake the modification:

1. Because you are modifying an existing working unit, this project is intended for those with reasonable soldering skills and at least some experience with digital components. If you are unsure, enlist the aid of someone capable (and willing) to carry out the modification.

2. Remember, modification to your module will render the module warranty void, although I expect most modules would be out of warranty anyway.

3. The modification details provided are for printed circuit boards identified by the '700352 F' designation. If you find a different number near where the seven ICs are located, then be careful to ensure that all mechanical details supplied here agree with your board. If they don't, I advise you not to proceed unless you have sufficient knowledge to adapt the circuit for that board.

## The circuit

Modifying the address decoding logic to remap the expansion RAM only requires two extra AND gates, so half a 74LS08 IC is all that is really needed, but I used NAND gates. The reason for this is that quite often, when a design is completed, extra input sig-



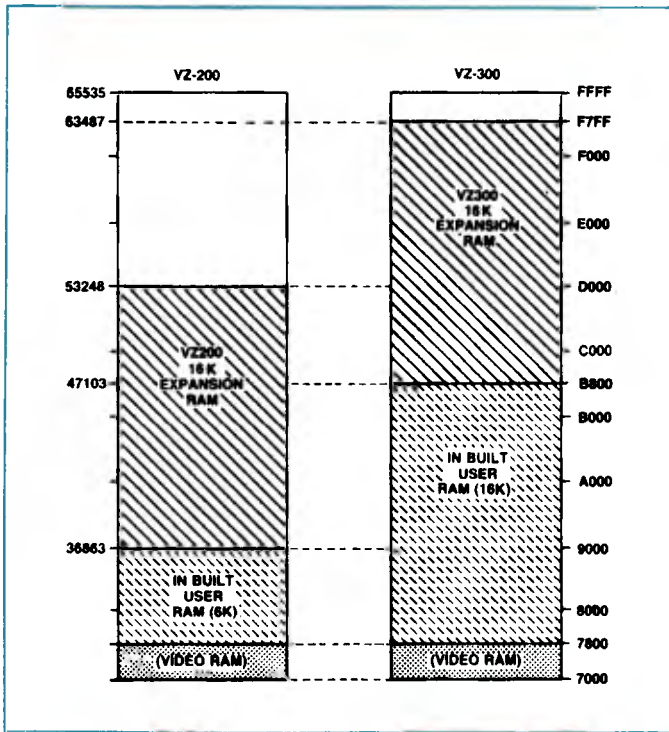


Figure 1. The memory maps for the VZ-200 and VZ-300. Note how their internal and expansion RAMs cover different address ranges.

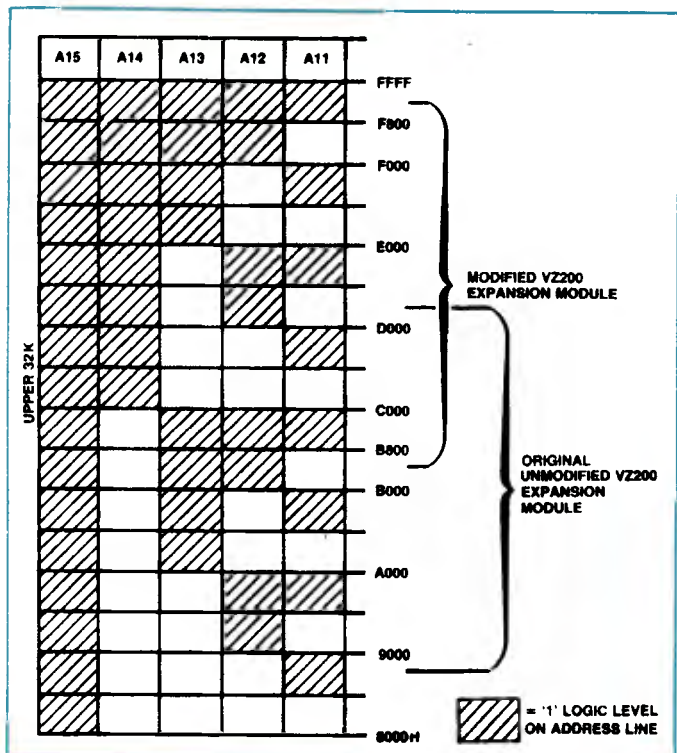


Figure 2. Address decoding map.

nals or controls are required. Because NAND or NOR gates can be configured to implement all of the basic logic functions, they are often used in at least some part of a circuit — even when that part could be more efficiently designed with other logic units. This is done with the view that if modification is required, then spare NAND or NOR gates allow some flexibility.

To further illustrate this point, it occurred to me, after working out the circuit, that it might be useful to have a block of RAM separated completely from the contiguous internal RAM for such purposes as having a reserved area of memory for running machine code programs, or implementing a printer buffer in RAM under software control. To do this the 16K RAM pack could be remapped to extend from C000H to the top of addressable memory, FFFFH. This would result in a 2K byte gap (for the VZ300 only) between the end of internal memory and the start of the expansion memory. When the BASIC interpreter seeks the top of memory, it is unable to jump this gap and so the top of memory pointers are set to the end of internal memory. This creates a reserved 16K block of RAM from C000H to FFFFH. That is, the top of memory pointers in BASIC are set to the same values as for a VZ without expansion module. This would still mean, of course, 18K for the VZ300, but only 8K for the VZ200. If the original circuit was implemented with AND gates the circuit would have to be re-designed. However, because NAND gates are being used, one of the paralleled inputs of one NAND gate can simply be switched to implement this change. This is shown in Figure 3.

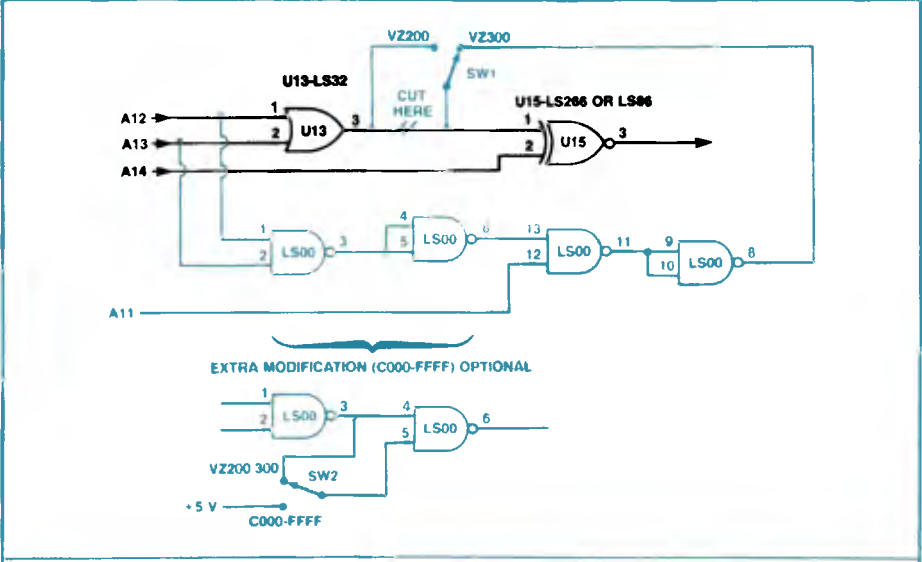


Figure 3. Modification details.

**The decoding logic**

Those who are not curious about the decoding logic details can skip this section and go straight on to the modifications. To work out the new decoding required, a graphical method was used. By looking at Figure 2, we can see that A14 = 1 (address line 14 = 1) covers from C000H to FFFFH (49152 to 65535 decimal). However, this is 2K bytes too high; the top 2K bytes need to be disabled, and 2K bytes added to the bottom, in order to enable a block extending from B800H to F7FFFH (47104 to 63487 decimal). That is, from the end of the

VZ300 internal memory on up. It might be noted that from F800H to FFFFH (where the memory should *not* be enabled) A13, A12 and A11 are = 1. Also from B800H to C000H (where the memory *should* be enabled) A13, A12 and A11 are again = 1. The only difference is that A14 = 1 in the first case, and = 0 in the second case. In other words, the memory should be enabled when A14 = 1 or when A13 and A12 and A11 all = 1, except when they all (A14-A11) = '1' at the same time. In logical shorthand this is written as:

$$A14 \oplus (A13 \bullet A12 \bullet A11)$$

where '⊕' is the sign for the logical XOR function, and '•' is the sign for the logical AND function.

Looking at the original circuit (Figure 3) it can be seen that the XOR function is available with A14 already connected (pin 2 U15), so if the A13•A12•A11 signal is connected to the other input (U15 pin 1) then the required memory enable signal is available on U15 pin 3. The required input is supplied to U15 pin 1 by the four NAND gates of the added 74LS00 IC.

## Modification steps

The following steps are the hardware modifications that need to be carried out to effect the change to the expansion module.

Turn over the module to find a sticker with the number 8 on it. This (apparently) indicates that the module is configured to expand on 8K VZ200.

Remove the six screws from the bottom of the case and gently separate the top of the case by means of a flat bladed screwdriver. Do this at the connector end of the module first, as there is a tendency for the cover to jam if it is pulled off at an angle. This will reveal a pc board to which a metal shield is attached by six soldered tabs.

Use solder wick or a solder sucker to remove as much solder as possible from the six tabs holding the metal shield in place, gently freeing the tabs from the board one at a time. Remove the metal shield.

At this point the component side of the board is visible with the physical layout as shown in Figure 4. Check to see that the board is marked with the 700532F designation. Hold the board with the component side towards you and the seven ICs at the top, and the discrete components (diodes, transistors etc) at the bottom (as in Figure 4). The middle IC of the seven ICs should be a 74LS86 or a 74LS266. In either case the modifications are the same.

Find the track on the component side of the board which runs from pin 3 on the 74LS232 to between pins 12 and 13 on the 74LS86/266 and cut it carefully with a sharp knife as in Figure 5.

Decide where to mount the SPDT change-over switch. I soldered a right angle pcb mounting type to the board itself (see Figure 4). You will probably need to shorten the terminal legs of the switch first and make sure the switch will not foul the metal shield when it is re-fitted. Another arrangement would be to mount the switch through a hole drilled in the top part of the plastic case. This is satisfactory providing the switch protruding out does not foul the printer or joystick interface plugged in next to it.

Using multi-strand insulated wire (wire stripped from rainbow ribbon cable is excellent) connect the centre (or common) terminal of the change-over switch to pin 1 of the

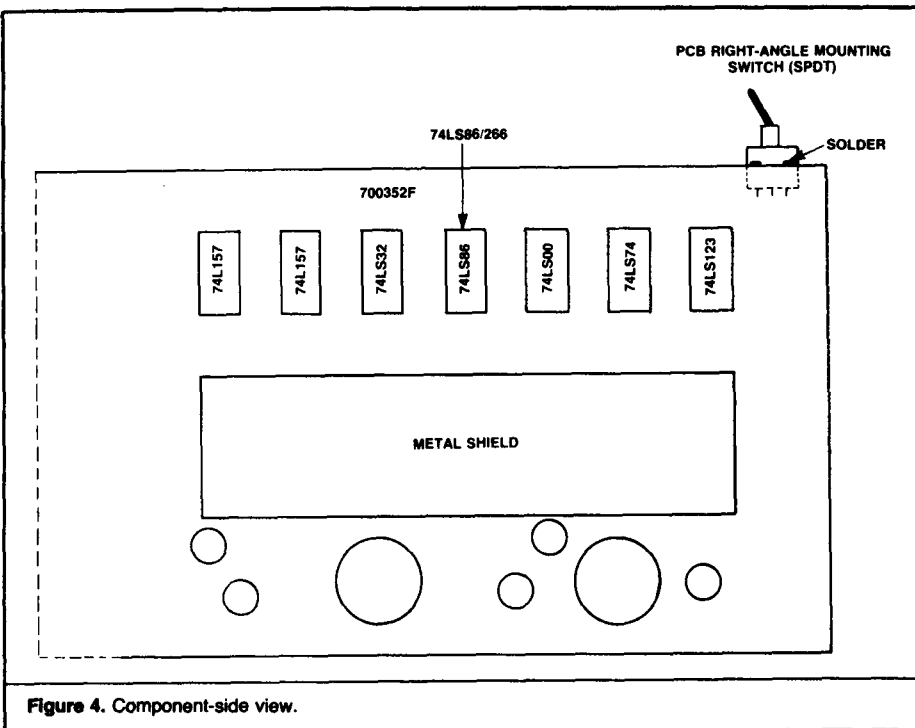


Figure 4. Component-side view.

74LS86/266, then connect pin 3 of the 74LS32 to one side of the switch (this position will select normal VZ200 operation).

Carefully bend all the pins *except* pins 1,2,7,12 and 14 on the 74LS00 at right angles to their original positions and carefully solder 'piggy-back' style pins 1,2,7,12 and 14 of the 74LS00 to pins 1,2,7,12 and 14, respectively, of the 74LS32.

Join pins 3, 4 and 5 on the 74LS00 together and solder them. Also join pins 9,10 and 11 together and solder. Join pin 6 to pin 13 using flexible multi-strand wire, then join pin 8 of the 74LS00 to the remaining side of the switch to give the VZ300 mode.

## Testing

That completes the hardware modification and the module is now ready for testing in your VZ300. Go over the modification carefully, making sure the wiring is correct and look out for solder bridges. With the power off, plug in the modified module, switch to VZ200 mode, and turn on the power to the computer. Type in the following

```
PRINT PEEK(30897) +
256*PEEK(30898) <RETURN>
```

If everything is OK, the response should be 53247

Now switch off the power to the computer, switch to the VZ300 mode and then switch the power back on. Type in the above line again. This time the response should be

```
63487
```

If any of the above two responses are not obtained, then switch off immediately, and re-check the modification looking for wiring mistakes or solder bridges.

By comparing these two responses with

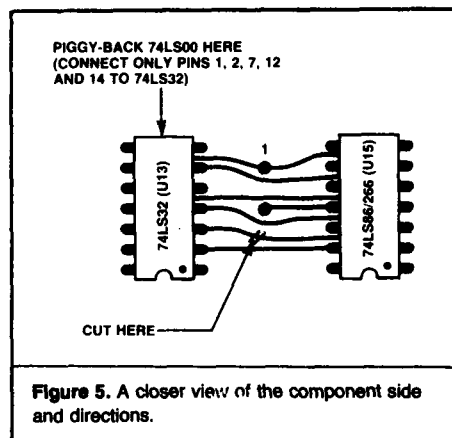


Figure 5. A closer view of the component side and directions.

the response obtained without an expansion module plugged into the VZ300, it can be seen that the modification enables all 16K (16384) bytes of the expansion memory instead of only 6K (6166) bytes of the standard VZ200 module. That is:

- top of memory VZ300 alone = 47103;
- top of memory VZ300 + unmodified module = 53247 (6144 bytes extra);
- top of memory VZ300 + modified module = 63487 (16384 bytes extra).

## Extra modifications

Before the module is re-assembled, an extra modification can be made, as mentioned earlier. This is to remap the expansion module to the top of addressable memory for reasons outlined before. This involves adding an extra change-over switch as shown in Figure 3.

Note that any of the switch connection positions can be replaced by direct wiring if operation in that mode is permanently required.

Happy Hacking!!!