7854 fixing the ROMs

If you became the proud owner of a 7854, and you cannot use it supposedly for the ROM rot problem, any fix is too slow, and so it was for me. So, I chose the solution based mostly on the possibility to set it up in a short time. I tell you what I could do in 5 working days.

The VintageTEK modification

The modification that I am going to describe is that proposed by the VintageTEK museum. I just add few tips. Its aim is to replace the four 7854 original Mostek ROMs with two 27128 (or 27256 as we will see) EPROM memories.

The reasons to chose this solution are the following:

- *there are only two types of EPROMS pin to pin compatible with the original ROMs: MCM68764 and MCM68766. They require a suitable programmer that I had not readily available.*
- *initially, I was going to use four x 2764, each with an adapter, but that is impossible: the adapters are 28 pin long, while in the 7854 ROM board there is room just for 24 pins.*

The theory

A 27128 (128k bytes) allows to store internally the equivalent of two 64k bytes Mostek ROM.

The ROM sockets on the ROM board are addressed as follows:

- *U100 and U110: selected when A14 address line is low;*
- *U200 and U210: selected when A14 address line is high.*

This is the original situation:

A14	Socket	Byte	Image file
LOW	U100	MSB	160-0408-XX
LOW	U110	LSB	160-0409-XX
HIGH	U200	MSB	160-0410-XX
HIGH	U210	LSB	160-0411-XX

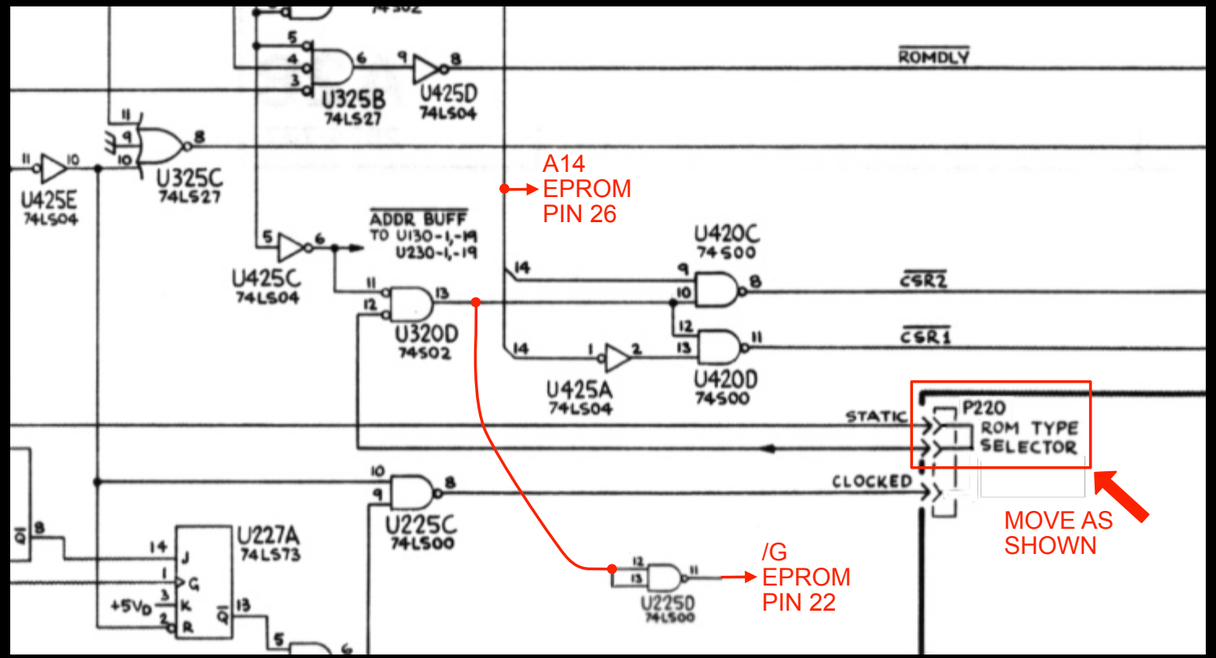
This is what we want to implement (note that using a 16-bit bus, the address line A0 is not used; every even address has a LSB and a MSB bytes - the EPROM address lines are so scaled by 1):

EPROM	A14	Socket	Byte	Image file	Addresses seen from MPU	Addresses inside the 27128
#1	LOW	U100	MSB	160-0408-XX	0x0000-0x3FFE	0x0000-0x1FFF
#1	HIGH	U200	MSB	160-0410-XX	0x4000-0x7FFE	0x2000-0x3FFF
#2	LOW	U110	LSB	160-0409-XX	0x0000-0x3FFE	0x0000-0x1FFF
#2	HIGH	U210	LSB	160-0411-XX	0x4000-0x7FFE	0x2000-0x3FFF

The new EPROM will have:

- *pin 20 (/E) strapped to ground; pin 22 (/G) is enough;*
- *pin 22 (/G) obtained by adding an unused inverter as shown in the diagram below;*
- *pin 26 (A13) connected to the board A14, used to select the low part of the EPROM (ex U100, U109) or the high (ex U200, U219),*

Note the setting of P220.



The practice

I ordered the adapters from Retro Innovations and they very kindly shipped them immediately. But I didn't want to wait weeks for the postal service, so I designed a new adapter and ordered it by PCBway, which delivered 10 of them exactly in 5 days, with about 30€ including shipment.

My adapter is shown on the page on the right, together with its schematic, and maybe is very similar to the other one from Retro innovation, perhaps with the following differences:

- *it already leaves free the pin 20 from the ROM board;*
- *it allows to use and take advantage of 27256 (see below);*

The adapters must be inserted on the U200, U210 sockets, aligned on the bottom (opposite to pin 1).

The Gerber files for the ROM adapter are available free on the this book's web page, where you find also the binary images for the FLASHs (version -00), ready to be flashed (load them at 0x0000 if 27128 or 0x4000 if 27256) with no modification to the adapter.

Note: you cannot use the normal pin strips to insert the adapter into the socket, because the room is scarce and you could interfere with the GPIB board. You have to apply the pins so that the adapter arrives to contact with the socket. I used 8.5 mm pins, soldering them in position with the aid of a socket similar to those used on the ROM board. In next pages you find all the details.

The images

You can find all the memory images on TekWiki. Personally, I am for the policy of small steps, so I suggest not trying an update to a later version, at least at the beginning, but use exactly the same image as you already have, as you can read from the chips (the "-00", for example).

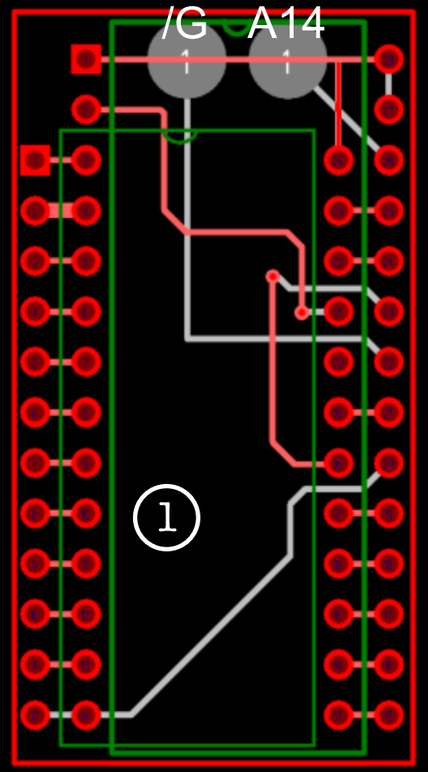
The PLA

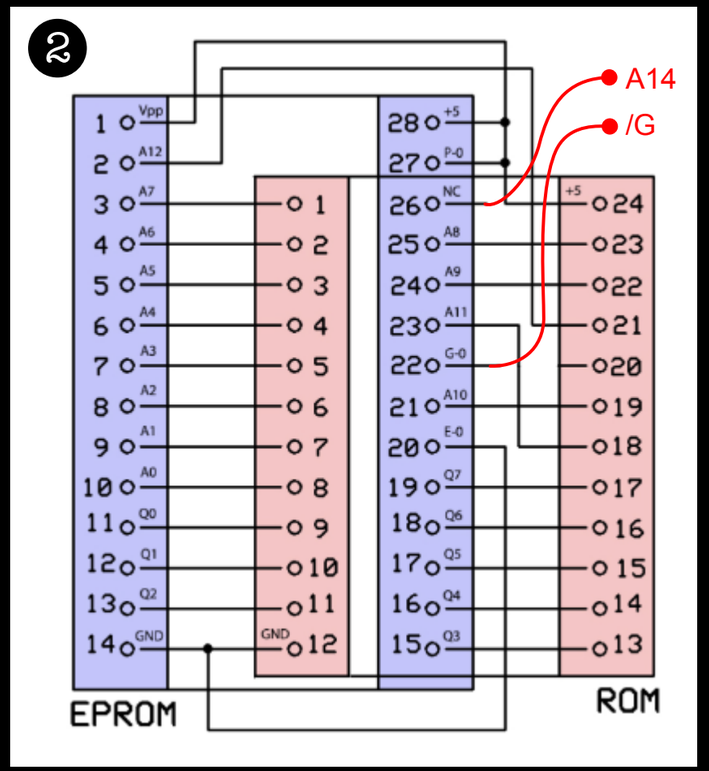
To understand the PLA, we need to go back to the first ROMs. They were very risky for the manufacturer. You had to produce a very high minimum quantity, and, if there was a mistake in the software, you had to throw all them away. To limit the damage, Tektronix invented this PLA mechanism, that I had not ever seen before. The PLA (Programmable Logic Array) is a logic circuit that, instead of being composed by discrete ICs, can be programmed via suitable development tools and programmers. The PLA monitors the address bus, and, when encounter a certain preset address, it activate two spare 2716 EPROMs U400 and U410 instead of the ROMs. In the area of the 2716 selected in this way, there was normally a JMP or little more, referring to other code in the 2716s themselves.

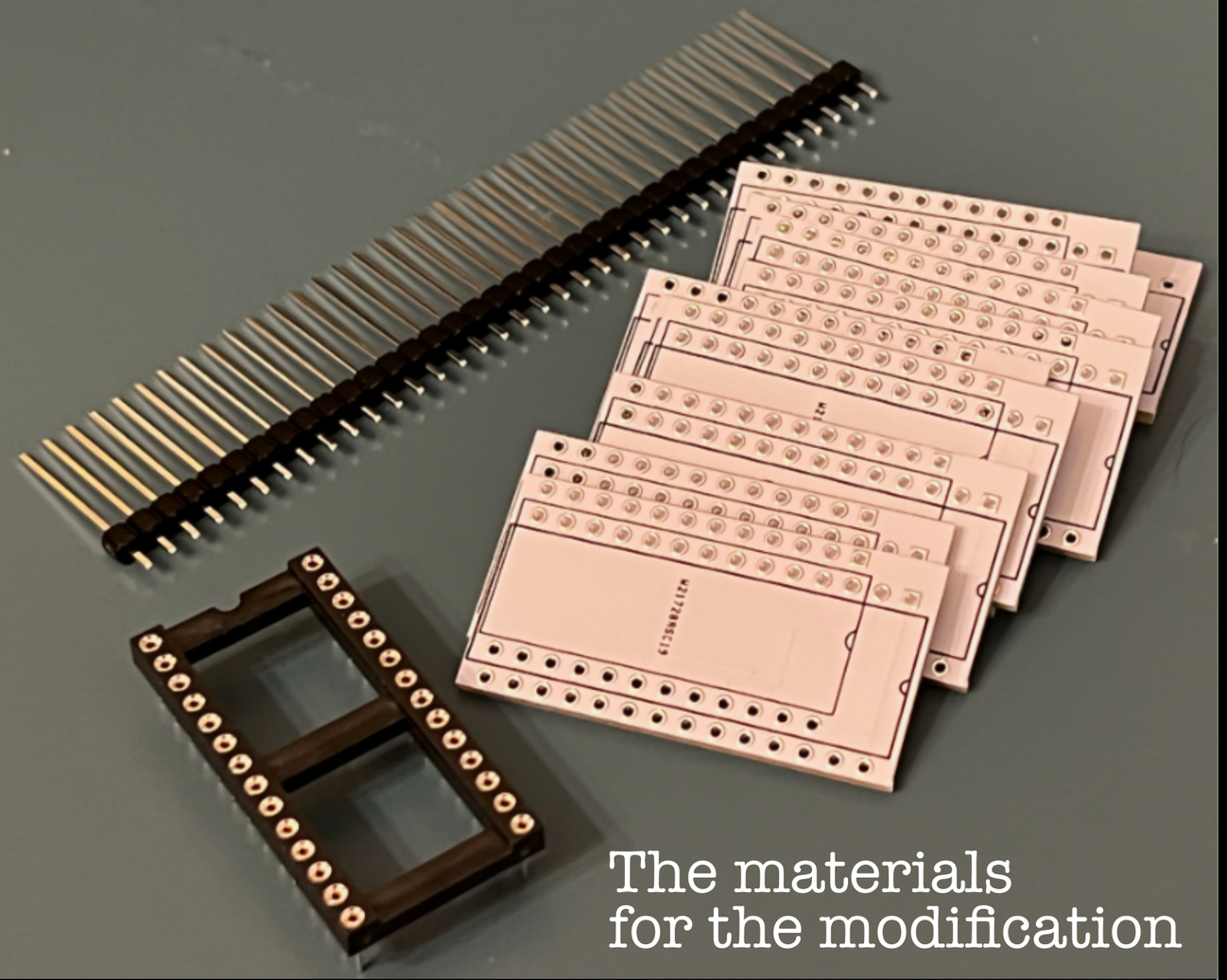
For what I could understand, the PLA allows to add a revision level. For example, in my 7854 I have the ROMs ed. -00 and the PLA -01 (but for what I read, the -01 ROM version was not different from -00, it was only a change in the supplier). On TekWiki there are three PLA version: -00. -01 and -02.

You could think that, if you upgrade the ROMs you could stay without the 2716s, but is seems not so. Maybe that some code is always read from the FLASH, or that a checksum is performed to see if the chips are installed. I could fix the ROM, as above explained, and so I am not going for experiments.

❶ The adapter layout; ❷ The adapter schematic diagram.







The materials for the modification

Preparing the files for the programmer

The ROM image files are exactly 8,192 bytes long (0x0000 - 0x1FFF). Each 27128 is one byte wide and contains all the LSB (or MSB) bytes for total 8,192 x 2 = 16,384 memory positions. So, to prepare the images for the EPROM programmer we need to concatenate two files (both LSB or both MSB). To do that you can use the Linux or Mac terminal, as follows (this example is with -00 version files):

```
cat 160-0408-00.bin > 27128MSB.bin
cat 160-0410-00.bin >> 27128MSB.bin

cat 160-0409-00.bin > 27128LSB.bin
cat 160-0411-00.bin >> 27128LSB.bin
```

Now you have the two files for the programmer. The LSB EPROM will be put, by mean of the adapter, in the U210 socket, the MSB in the U200 socket. The two MSB and LSB files are downloadable from my web page, ready to be flashed (version -00).

Using the 27256

The adapter allows to use, instead of the 16k byte 27128, a 32k byte 27256 as well, in the following ways:

- *just load the 27128 image at 0x4000 instead of 0x000;*
- *use the 27256 double capacity to store two different images, e.g. version -00 and -02, if you like to experiment, one at 0x0000 and another 0x4000.*

In the latter case, you can cut the trace between pins 28 (+5V) an 27 (A14) of the 28 pin and connect pin 27 to GND. When pin 27 is at GND, the image at 0x0000 is active, when it is at +5V, the image at 0x4000 is active.

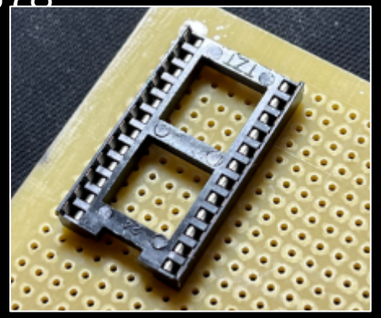
Checkpoints

The table on the right gives you some checkpoints to be sure you have the right file loaded (-00):

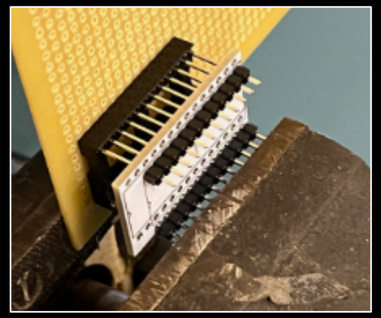
Acknowledgments

Most of the above information comes from VintageTEK museum and Modular Synthesis. I just added some explanations.

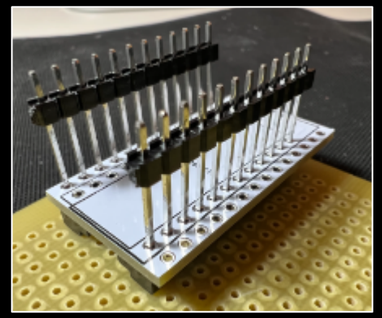
EPROM	From image file	27128	27256	Byte	Content
MSB	160-0408-XX	0x0000	0x4000	MSB	0xDB, 0x04, 0xDC...
	160-0410-XX	0x2000	0x6000	MSB	0xC0, 0xDA, 0x13...
LSB	160-0409-XX	0x0000	0x4000	LSB	0x00, 0xE2, 0x70...
	160-0411-XX	0x2000	0x6000	LSB	0x20, 0xA0, 0xA1...



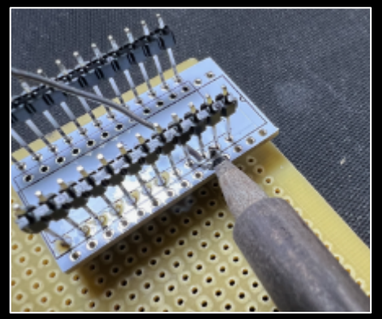
❶ Start preparing a breadboard with a socket similar to that you have on the 7854 ROM board. It will be used to help the adapter construction.



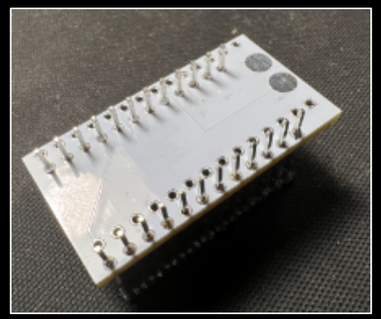
❷ Use a vice to be sure that you inserted the pins all the way in. It is very important because we have not one tenth of a millimeter to waste.



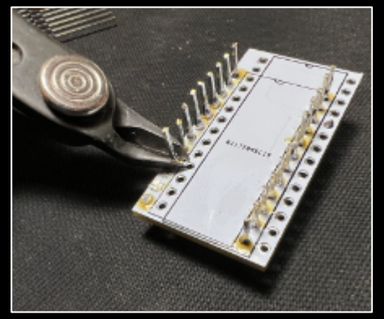
❸ Lower the small PCB to arrive in contact with the socket. Be sure it didn't remain raised.



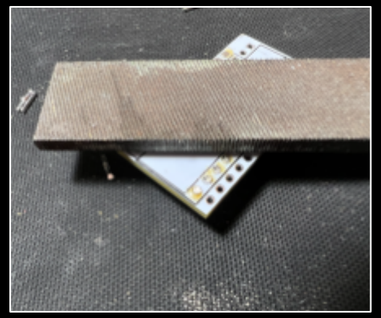
❹ Solder the pins from the upper side. Don't use too solder, or it will drip on the pins below, but be sure you soldered well all of them.



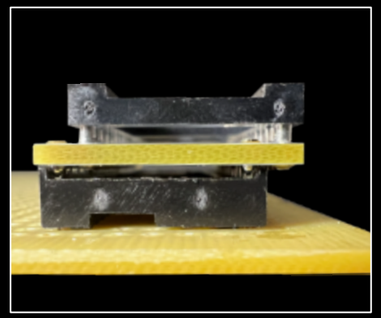
❺ If you turn the PCB, you can admire your work. The pins will be straight and well aligned.



❻ Cut the residual pins on the top side. Cut them very low to save space.



❼ As we said, we have no room to waste. So carefully file any protrusion before soldering the EPROM socket.

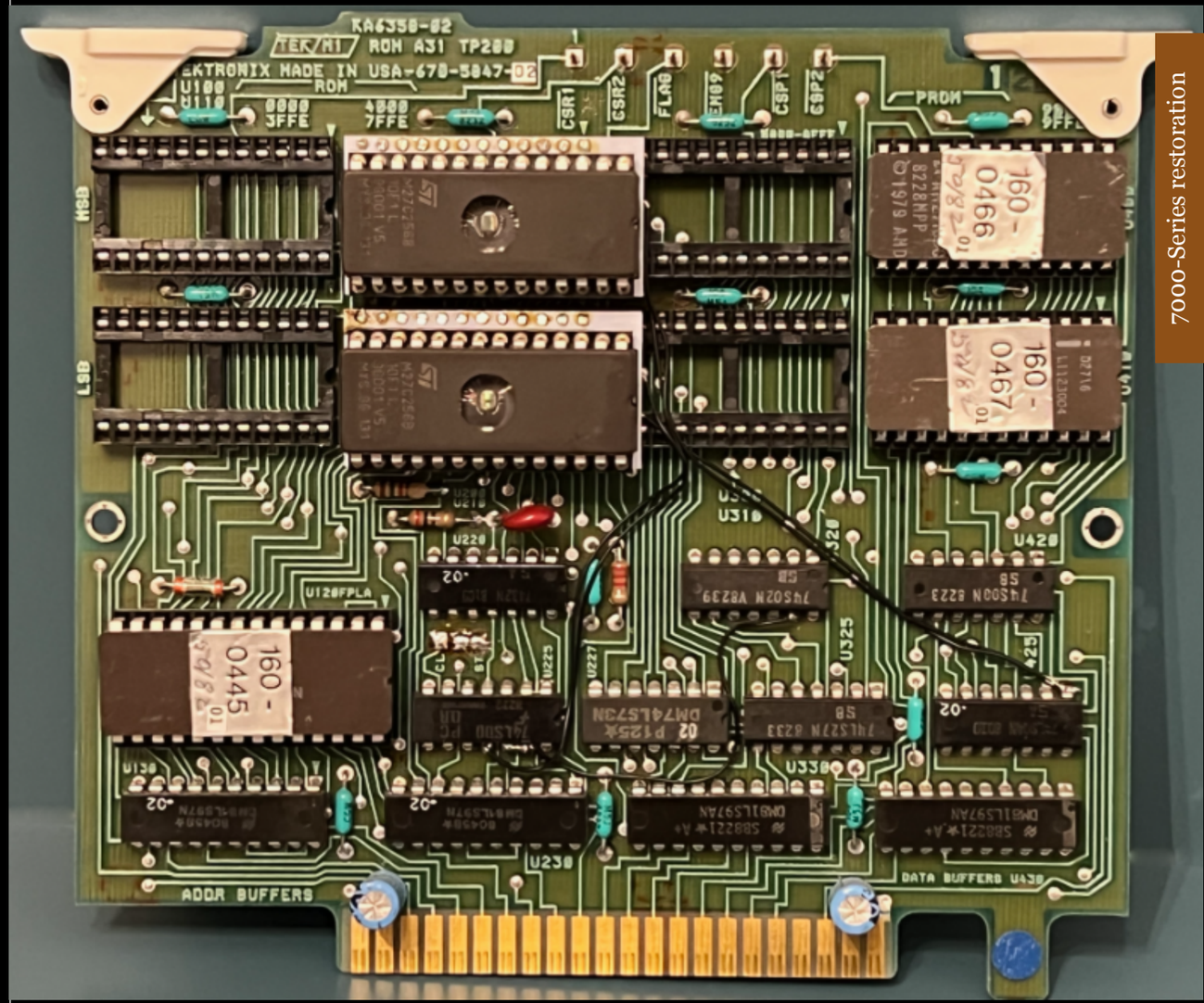


❽ Here you can see the completed adapter wafer. Note that every level is in touch with the adjacent. Wash the solders with a suitable spray product.



❾ The completed adapter with its EPROM on top. You have now only to solder the A14 and /G wires to the underlying pads and insert it in the ROM board.

The retrofit step by step



The modified ROM board. See the previous schematic diagram for wiring. There is no track to be cut, only wires to add. Don't forget to move the ST/CL jumper. The four 24-pin sockets will remain empty. Reinsert the board in the rack together with the adjacent GPIB interface. Note: for testing you don't need the GPIB interface, the scope starts even without it.