ProTeleVision PT8601-> PT8631 Conversion

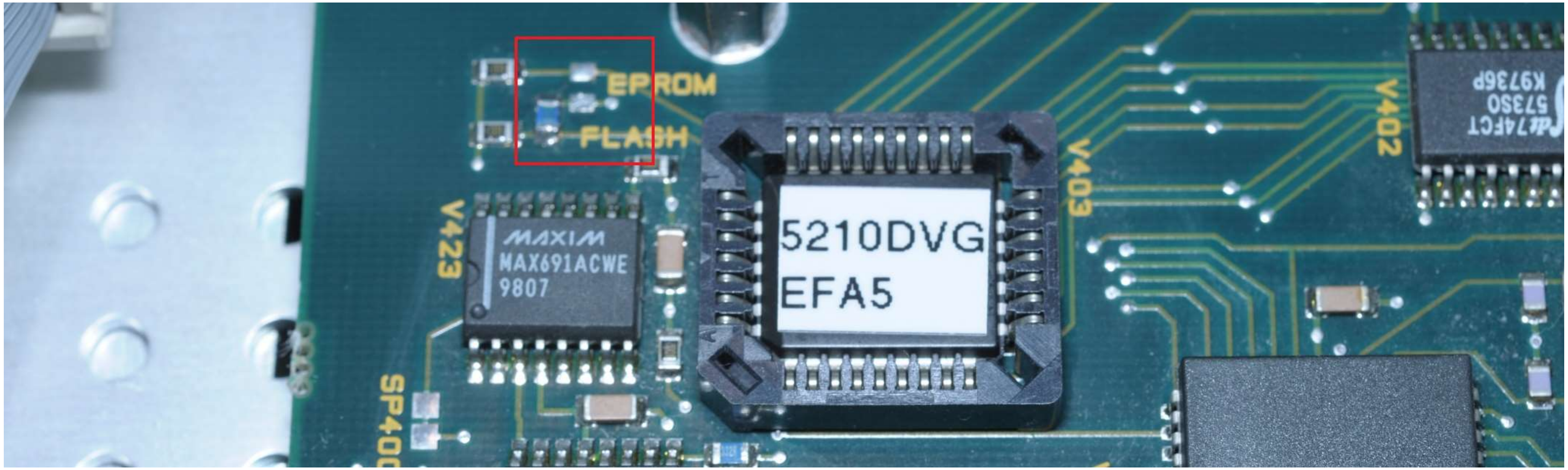
Pre-Preparation

Master PROM (PT5210 only, mainboard)

If using a PT5210, regular firmware will not accept PT863x modules. The firmware must be changed to the 5210 "DVG" version which is contained within this repository. It is functionally identical to the regular PT5210 build but supports these modules. If using a PT5230 or PT5300 this step is not required.

This operation must be performed by an EPROM programmer.

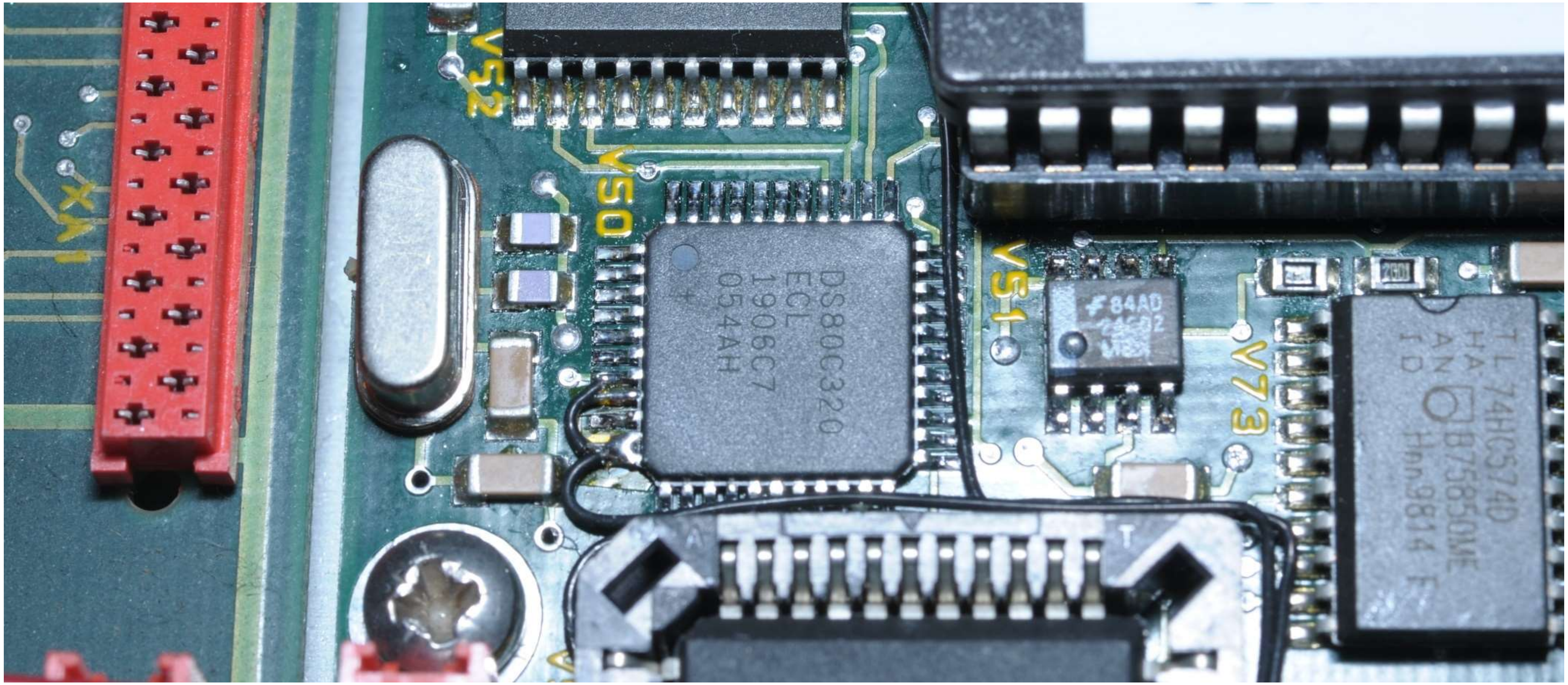
Either an OTP PLCC EPROM (M27C4001) can be used, or a flash chip like AM29F040 or SST39SF040. If using a flash chip the highlighted resistor must be moved to the appropriate position.



Check for existing modifications

Depending on the revision of the PCB there may already be a wire connecting V46/pin 30 to V63/pin 24. This modification (if present) **must** be left in-place. If it is not present then it is likely the final revision of the PT8601 PCB where this has been corrected.

Modification 1: Replace V50 with DS80C320-ECL+

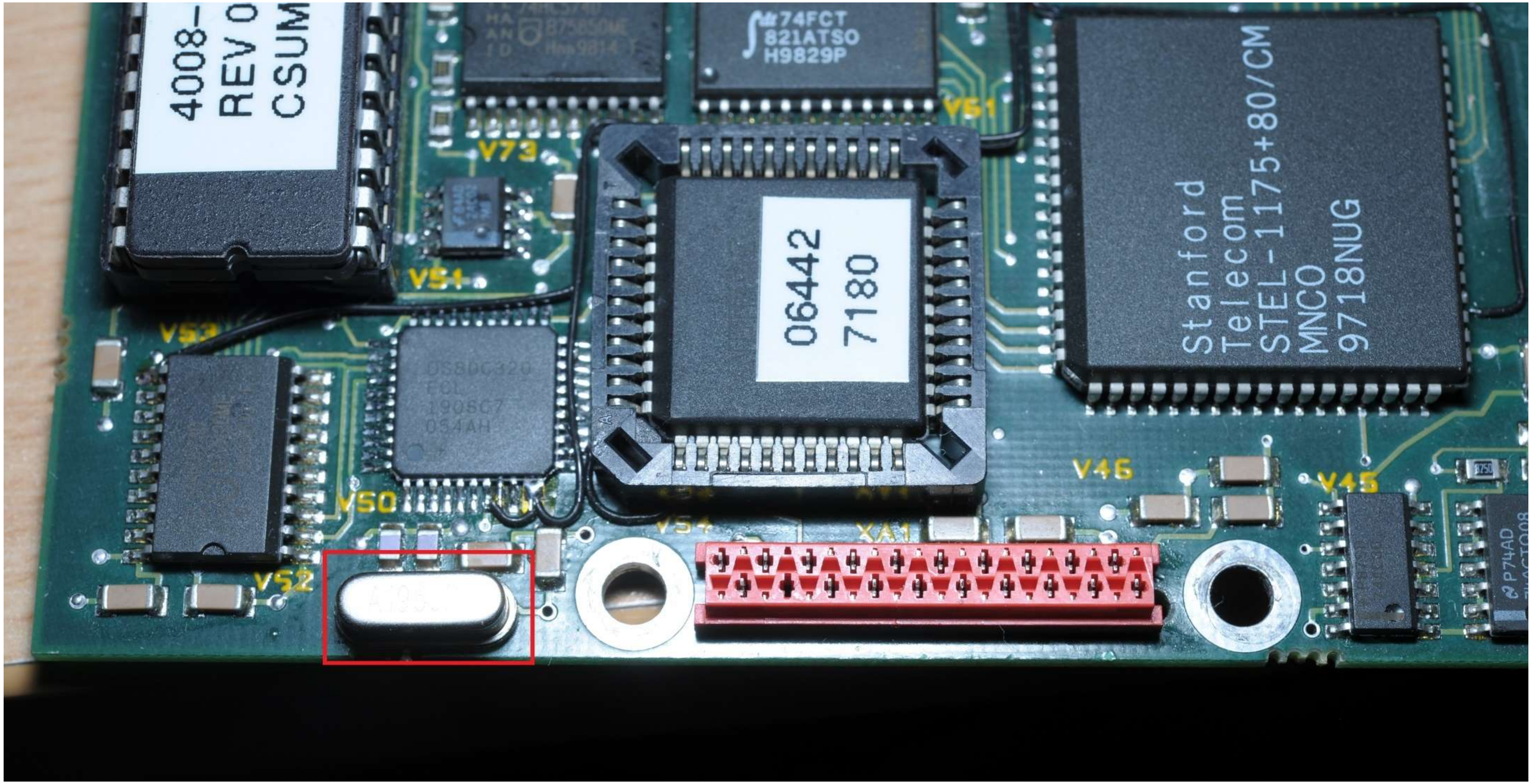


This is required because the original microcontroller is too slow for complex pattern generation.

NOTE: Pins 8, 10 and 11 must be lifted (not reconnected to the PCB). As seen here they will eventually need wires connected. I recommend placing thin strips of Kapton tape on the pads before soldering down the new part.

Modification 2: Replace X1 with 19.6608 MHz crystal

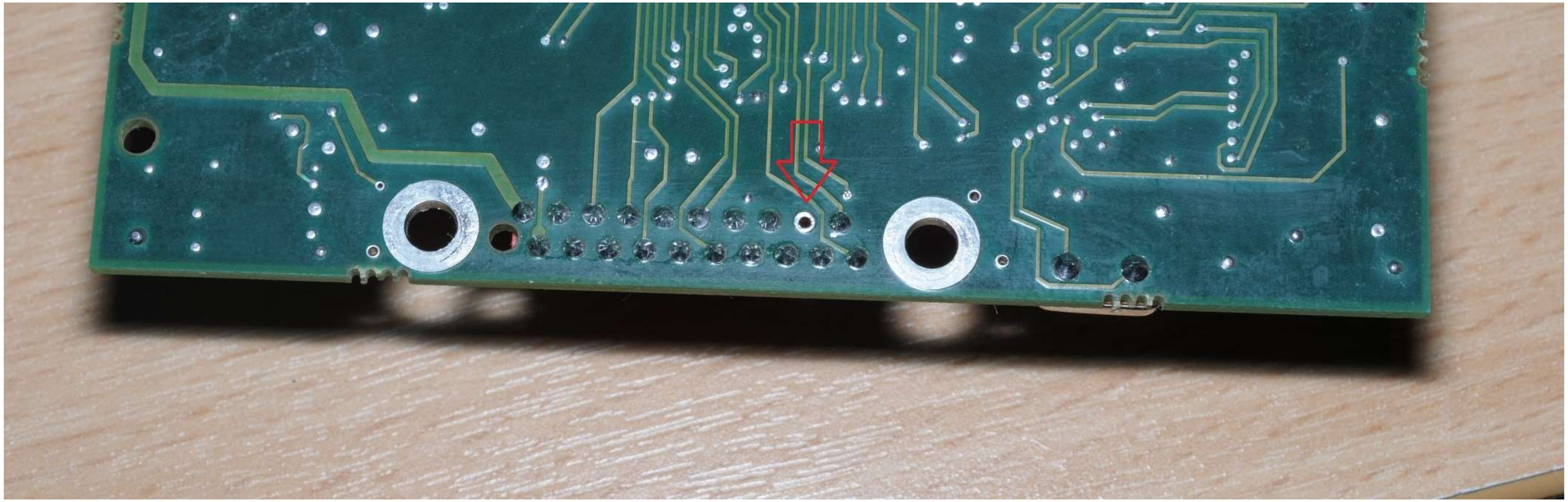
I used ABL-19.6608MHZ-B2



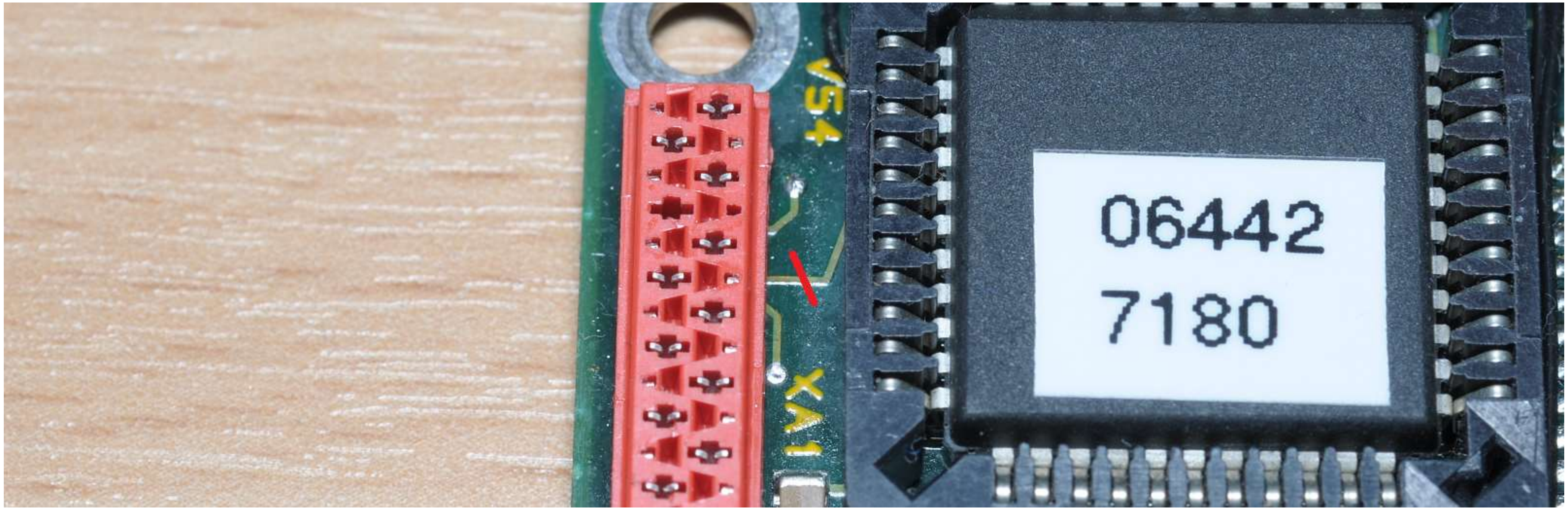
Modification 3: Disconnect V54 pin 7 from XA1 pin 17

There are two ways to achieve this.

Method 1 (preferred): Remove XA1 and pull pin 17 out, then re-fit. Removal will probably destroy the connector. I yanked the plastic off with side cutters and desoldered the pins one-by-one. The replacement part is 9-215079-0. Not a bad idea to replace anyway as these connectors are now very old and can be corroded.

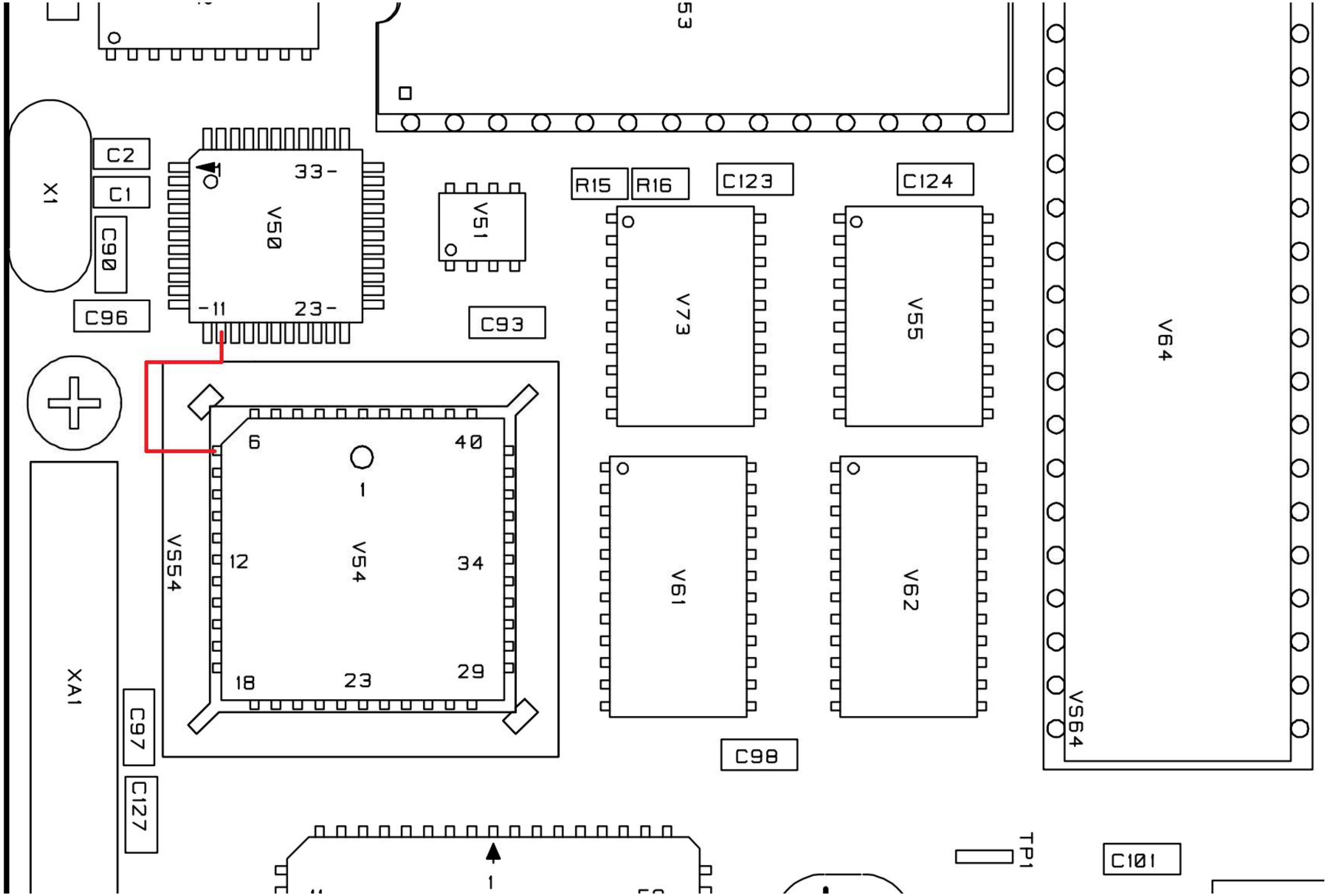


Method 2 (less ideal): Cut the trace near XA1 here:



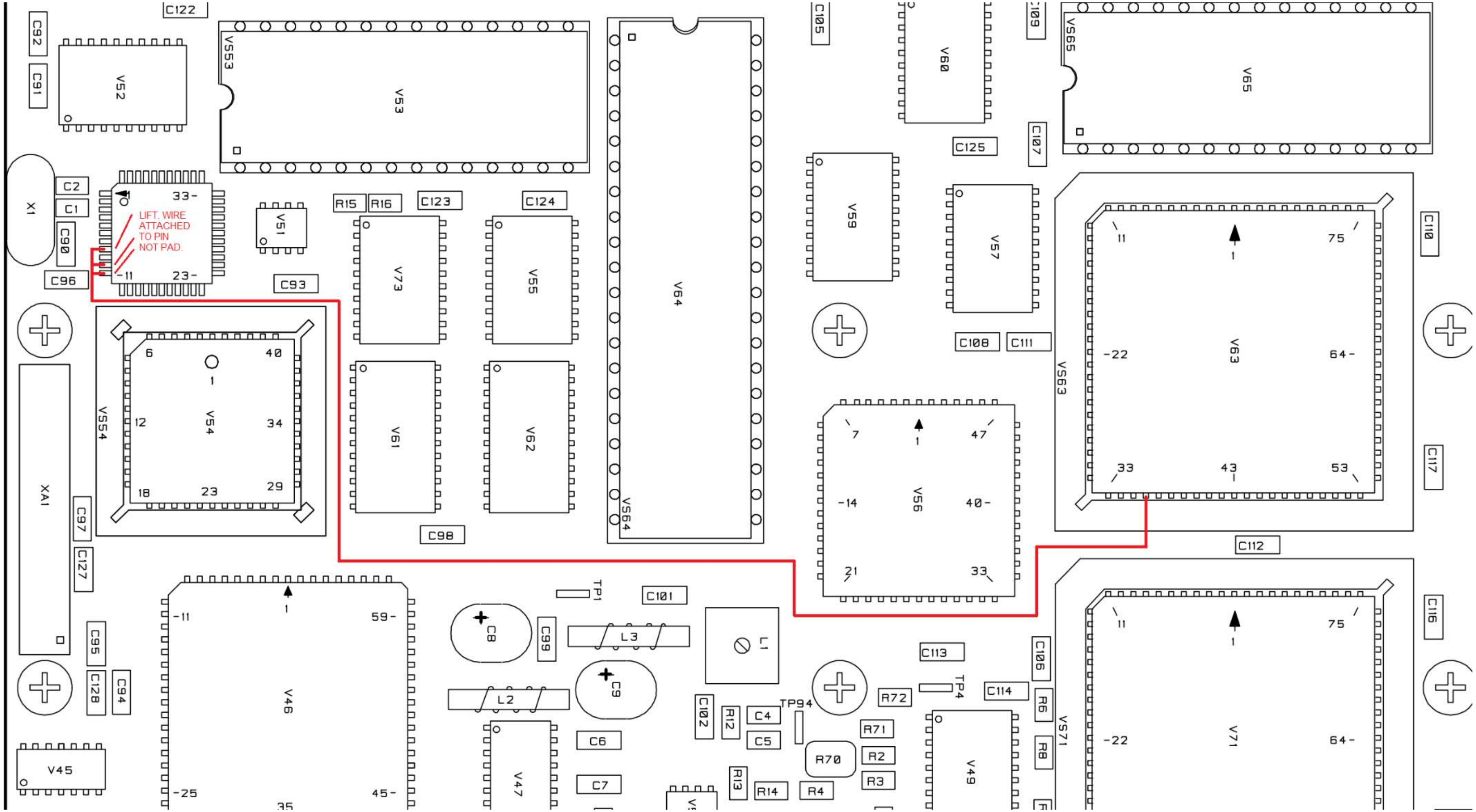
I don’t recommend this approach due to the risk of damaging nearby traces but it is theoretically easier.

Change 4: Connect V50 pin 13 to V54 pin 7

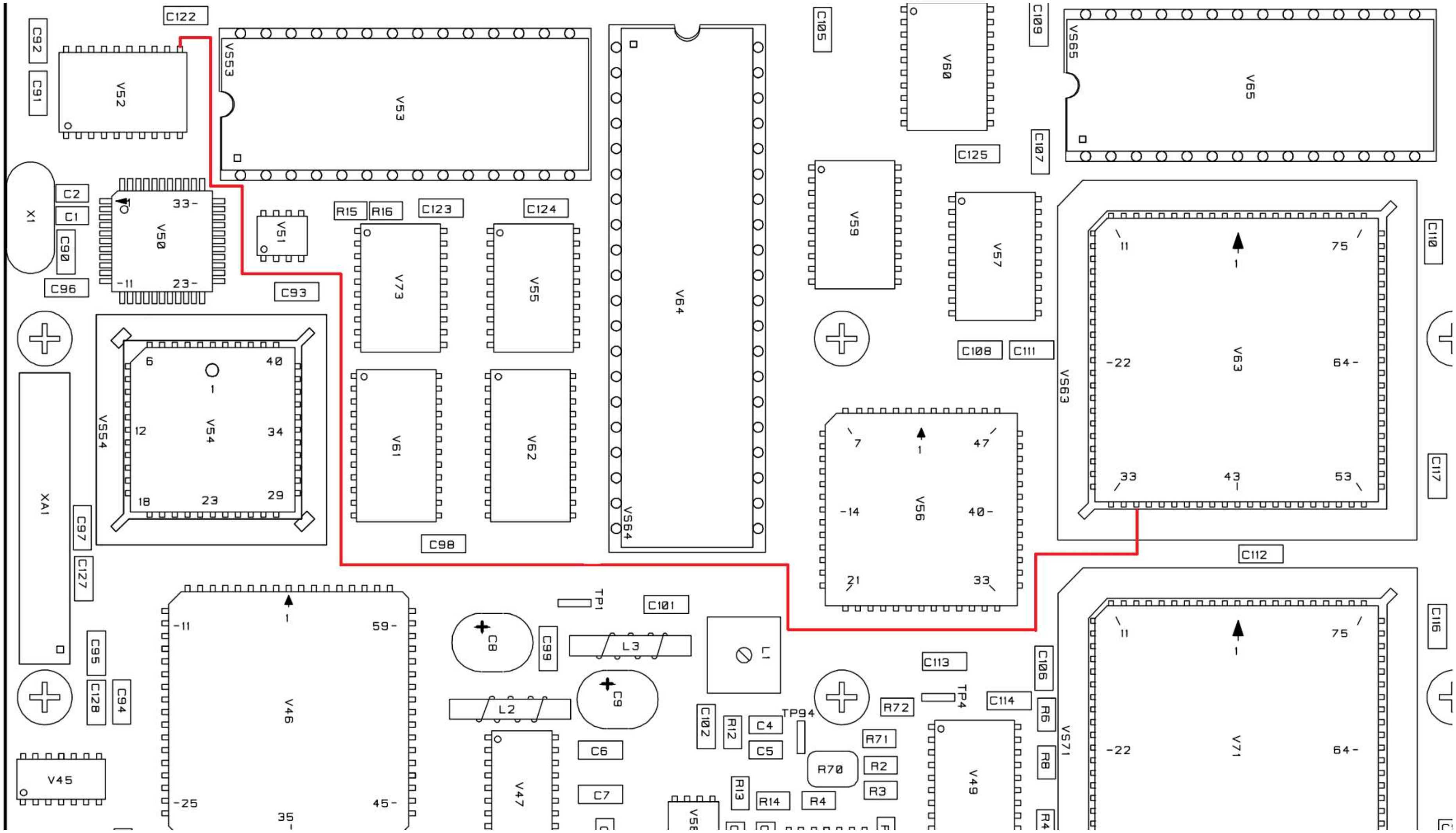


Modification 5: Connect V50 pins 8, 10 and 11 to V63 pin 36

NOTE: V50 pins 8, 10 and 11 must be lifted/disconnected from the PCB



Modification 6: Connect V52 pin 11 to V63 pin 35

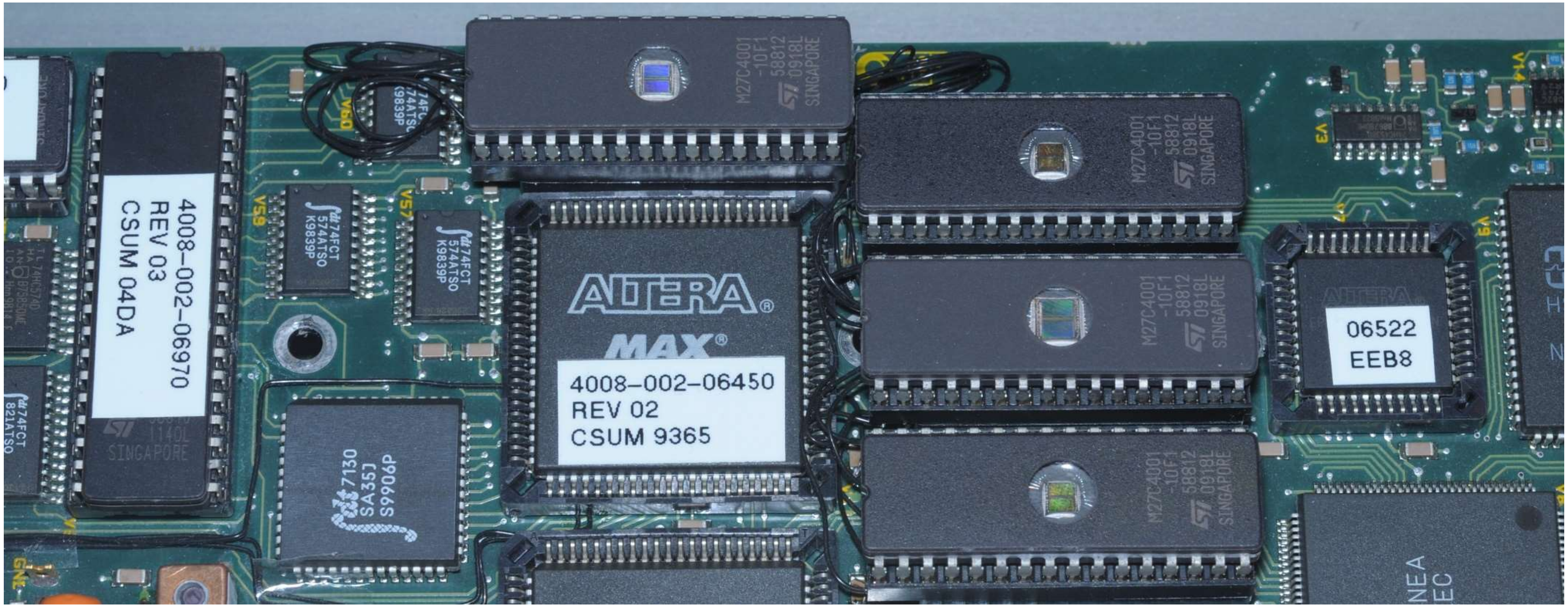


Modification 7: Change to larger ROM chips

The PT8631 needs a lot more ROM memory for all of the extra patterns. There are two ways to accomplish this:

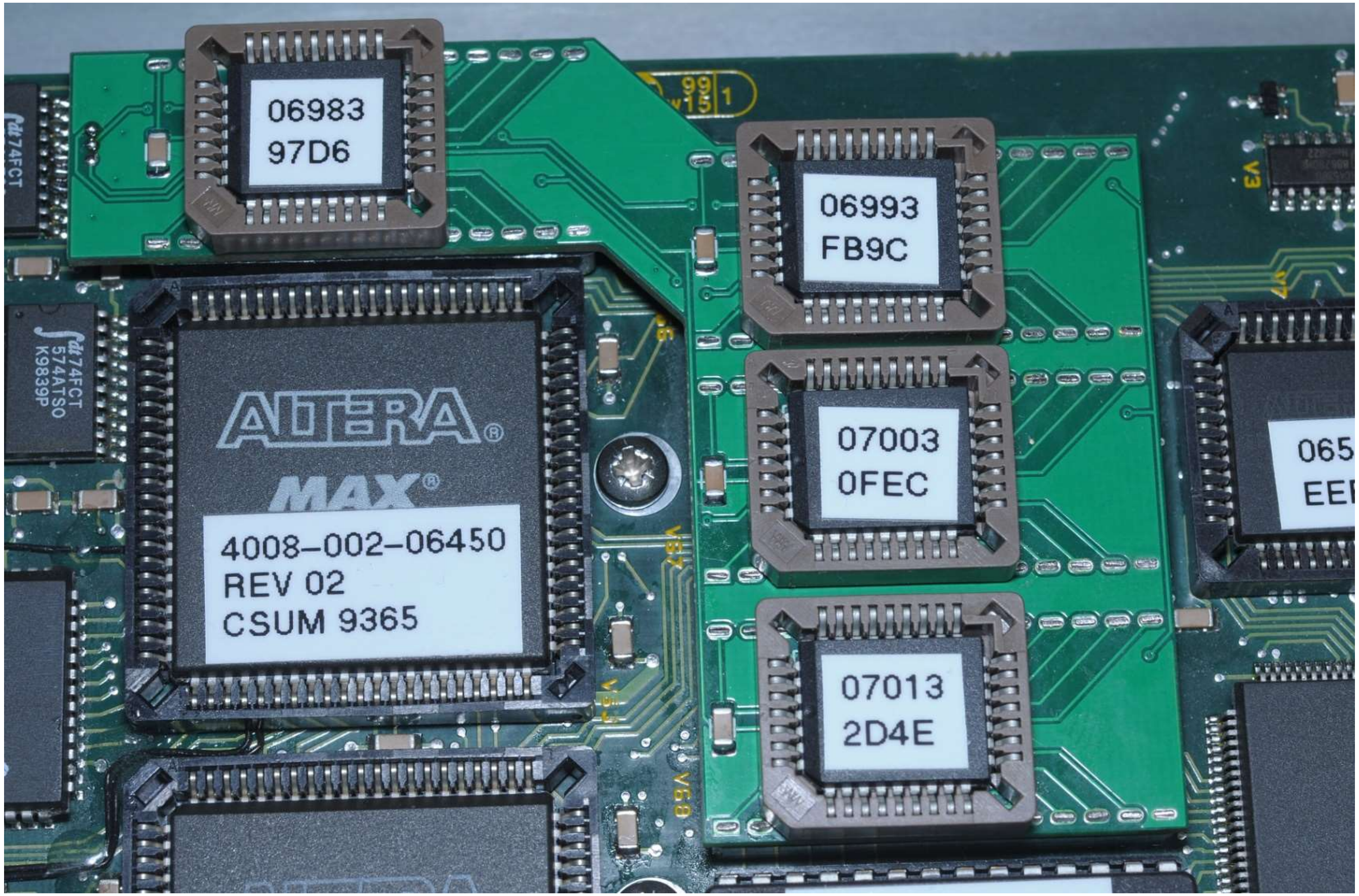
Method 1 (bodge):

Write the pattern data into larger M27C4001 EPROMs for V65, V66, V67 and V68 Insert them into the existing sockets. Overhang the top 4 pins. Do the following with the overhanging pins: Pin 1 (VPP) of each must be connected to VCC. Pin 2 (A16) of each must be connected to V60 pin 16. Pin 30 (A17) of each must be bent away from the socket and connected to V60 pin 15. Pin 31 (A18) of each must be connected to V60 pin 14. Pin 32 (VCC) of each must be connected to VCC at the decoupling caps for each.



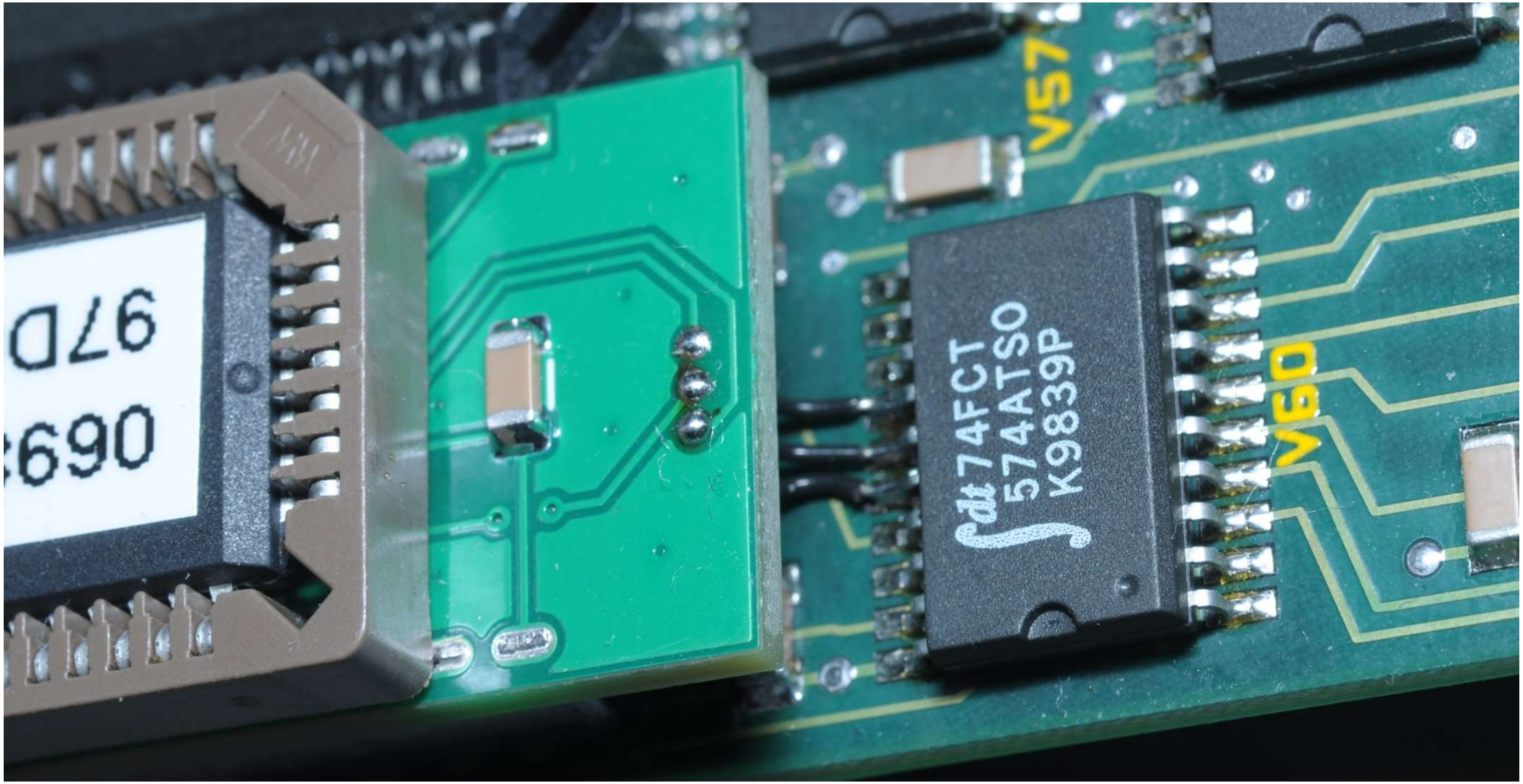
Method 2 (nicer):

PCB design files for an adapter board are supplied in this repository:



I ended up cutting pins from “donor” ICs and soldering them into the slotted holes. It is a tidy solution but it’s hard to build. The result was ideal because the chips remain socketed, but the entire assembly is slim enough that the lid can still be closed when the module is installed in the correct (top) slot.

The three extra address lines are conveniently routed (in the correct order, *do not cross the wires*) to the end of the PCB overhanging V60 where they must be connected:



Modification 8: Reprogram all other chips

All of the other chips on the board must also be re-programmed. Flash/EPROM chips are easy, but the PLDs are not.

I used a Labtool 48XP/48UXP for the task. While EPM7xxxS CPLDs are used, I am unsure if JTAG programming them is feasible as the JTAG pins have been used in the design in some cases. If using a different set of PLD chips, good samples of EPM7128SLC84 are hard to find, thus “00206532E.POF” is provided which can be programmed into a EPM7128ELC84.

A quirk of the POF files is that they all target the original EPM7xxxLC (ancient) parts for some reason or other. Probably because nobody could be bothered re-targeting them. The Labtool software offers the option to convert them to EPM7xxxS on the fly. This is probably what they did in the original factory given that EPM7xxxS parts were used, but the checksums on the stickers are what they would be if written to EPM7xxxLC parts.

Modification 9: Reset device ID

Having done all of this unfortunately there are further measures in place to prevent modifications like this. The on-board EEPROM contains a signature which allows the firmware to check that it is being run on the appropriate hardware. If the signature is not corrected the module will ignore the select pattern command from the SPG and only output a black & burst signal.

There are two ways to solve this problem. The first is to desolder the EEPROM, modify its contents, and re-fit it. The `Programming_Files` directory contains a before and after (`V51_PT8601.BIN`/`V51_PT8631.BIN`) but use your EEPROM as it contains the calibration constants which are unique to each board.

The second and cleaner option is to tell the module to re-write the correct signature. This can be done by undocumented SCPI commands via an RS-232 connection to the rear of the SPG. The commands can just be pasted into PuTTY at 9600 baud Xon/Xoff (factory default. check config menu to be sure):

```
:FACT:PASS 'Marilyn';
:FACT:V24C:ADDR BBU_1;COMM 'AZ',1;
```

The first command unlocks the "FACTory" command. The password is hard-coded and will always be as specified above. The second command will tell the analogue TPG to re-write the signature, but keeping the rest of the EEPROM data. The ",1" at the end is very important. If this were not provided, or ",0" is provided instead the entire EEPROM will be erased.

The "address" in this case is BBU_1 however it may also be BBU_2, BBU_3 or BBU_4.

If uncertain of the address issue the following command to each:

```
:FACT:PASS 'Marilyn';
:FACT:V24C:ADDR BBU_1;COMM 'AR?',0;
```

This will read out the manufacturer name (which is only ever "PTV"). Only an analogue TPG will respond to "AR" so if you get a response, you have the correct address. Changing the '0' to '6' should return the model string (it is not used by the software so its exact contents in unimportant). It may already say "PT8631". This is because they used the same factory tool to program both. It does not mean you already have a PT8631.