Building the N8VEM Computer for the Total Beginner

This document is the next phase of the project, and continues from where Holmes left off on the construction phase of the computer project. In this discussion, we will work with the problem solving approach to be applied if the board fails to start up properly. We also assume that the SBC debugging involves just the SBC using the serial terminal interface with no connection to the bus or other bus boards.

We know that the N8VEM computer board is a proven design and that problems that arise will be primarily associated with construction issues and hardware-firmware compatibility. We will check a number of these issues and then retry the startup sequence. We will do the easiest or most important checks first, test a restart, and then move on to more complex issues if we are unsuccessful.

We will start with the construction issues and take the discussion through the following topics:

1. Protect your investment in the parts
2. Check for chip and part placement
3. Check jumper installation
4. Check PCB for soldering problems
5. Check Diode and Transistor orientation
6. Check for bent chip pins
7. Check power distribution and pullup resistors
8. Install chip type in conformance with base design
9. C
10.

We know that the computer design works and we know that many other folks have successfully built this processor, so we know that there are some problems that need to be fixed.

1. This board has a substantial cost investment that we do not wish to fry. The power is on or has been on and the board did not start as Holmes suggested, so let us turn the board power off and protect our parts investment.

   After turning off the power, feel all of the chips to see if any are hot. None of these chips should even be warm to the touch; and if they are, then the hot chips may well be damaged or installed backwards. Verify that the LED power indicator was or was not lit, as a short on the board would pull the power bus down such that the LED would not show green (if it was RED we have a different issue for discussion later).

The next step is to pull up the picture of the completed board from the web and compare the chips, their locations, and their orientations with your board. Make sure the proper chip (the part number designated) is in its designated socket, and that all of the indexed pin 1 ends are oriented toward the power plug. If chips other than those suggested are being used, it is important to check the data sheets to make sure the pinout and power pins are compatible with the original suggested chips and that the jumpers settings are compatible with these chip choices.

Step three is to turn the board solder side up and do a thorough inspection of the soldering. We are interested in three issues: pins not soldered to the board, cold solder joints, and solder bridges to the ground plane or other pins. It should be clear that solder bridges to adjacent pins could damage the parts or leave the chip improperly powered such as might occur if the pin were not soldered to the circuit traces at all. A good magnifying glass and a strong light are critical to this exercise.

1. Open solder joints will be the easiest problem to see and fix, as pins in open holes look very different from soldered pins. While you are working this issue, look for soldered holes where there is no pin or lead causing a pointed solder joint; as some leads may not have been in the hole when it was soldered and will be hanging loose on the top of the board.

2. Blobs of solder touching two pins causes a short, and is for the most part discouraged by the green solder mask, however, too much solder can still bridge the gap. Turn the board on its side and warm the joint with an iron and pull off the excess solder. The same thing can happen between a pin and the ground plane so keep the solder to a minimum and the joints pointed.

3. Cold solder joints are rough gray globy joints instead of shiny pointed tee-pee shaped joints. Gently reheat these joints with additional solder and pull away all excess solder when removing the soldering iron. Do the same for any pins that have round solder joints.

2. There are a large number of Jumpers on this board to allow for different chip selections, external backplane interactions, and user interface options. All of these jumpers have to be set consistent with the chips chosen and the configurations desired.

   All of the jumpers are configured to use the default pins 1&2 setting while using the base design chips and configuration. Pin 1 of the three pin set is marked with a small painted square on the silk screen layer.

   The jumper shorting plugs intended slide over and short either of the side pins to the center pin or short both pins of a 2 pin header. These plugs are available in various colors. I use Red plugs for choices that could damage components, Blue plugs for size or address selections which can hinder/affect operability, white plugs for options, and Black plugs for interface options like serial DTR/DSR which are most often not important.

   Pull up project photographs for blank and finished cards and check all of the jumpers. The blank board photo helps discern jumper location and orientation while the finished photo shows the configurations for the base design. Check off each jumper and plug so that none are missed (some option plugs may not be used).

There are solid state devices which have to be installed in the correct orientation to work properly. Diode and Transistor pins cannot be reversed. Many N8VEM designs use discrete transistors and diodes so these need to be checked for proper orientation. Completed board photos are useful in this inspection.

With the above checks complete (and likely some problems found and corrected) we are ready to test the board again, without much risk of frying critical parts.

Power up the board and check that the green power LED lights up properly. Press the reset button and wait a few seconds for the card to startup, configure all chips, load memory, and finally present the sign-on display. Feel chips for any that may get hot.

Early EPROM chips provide a sign-on message about a Prototype Test Monitor while more recent RomWBW proms provide a lengthy display of the system configuration. All of these recent configurations use a baud rate of 38400.

There are different boot Eproms and there are different configurations of those Eproms. There are standard options of the proms which are already built, and there are unique configurations that individual users might have implemented. The following are examples of sign-on messages from the boot Eproms you might have used:

```
     Early Sign-on:  Test Prototype Monitor Ready

     Recent Sign-on :N8VEM Z80 @ 8MHz ROM=512KB RAM=512KB
                     UART0: IO=0x68 16650 BAUD=38400 FIFO
                     CVDU: IO=0xE4 VDURAM=64KB
                     MD: UNITS=2 ROMDISK=448KB RAMDISK=416KB
                     FD: IO=0x36 UNITS=2
                     PPIDE: IO=0x20 UNITS=2
                     KBD: IO=0xE0
                     TTY: RESET
                     ANSI: RESET

     Other Sign-on:  N8VEM HBIOS v2.5.1, Build 17, 25-Jun 2013

                     A=RAM B=ROM C=FD0 D=FD1 E=PPIDE0-00
                     F=PPIDE0-01 G=PPIDE0-02 H=PPIDE0-03
                     Boot: (M)onitor, (R)OM, or
                            Drive Letter ===> BOOT FROM ROM
```

If you are successful and the board is running, you can move on to working with the Test Monitor or the CP/M system itself. If not, then continue with this debug process.

There is a follow-up document on the Test Monitor which explains the commands implemented and how this monitor might be used to debug various elements of the system.

At this point it makes sense to gently pull out all of the chips on the board and check to see if there are any pins that bent at their lower ends rather than protrude down into the socket. These bent pins are hard to see until the chip is extracted from the socket, and are more prevalent on drilled sockets than other press fit sockets.

While any (or all) chips could have a pin problem, it is more likely that the larger chips might have this problem, so the large chips should be evaluated first, and keep in mind that many of the interface chips adjacent to the bus plug have little or no effect on the operability of the processor system itself.

Be very careful bending these pins straight so that they do not break. You might notice that some chips have pins that spread out beyond the width of the sockets, which can be fixed by placing the chip side-ways on the table in order to bend all of the pins closer to the proper width at once.

If all of the pins were socketed correctly, it may be that some chips are of the right type but the wrong fabrication family (that is a 74LS32 is the same logic type as an 74HCT32, but the LS and HCT are different families). Many of the family types use different voltage levels (ie TTL versus CMOS) to differentiate digital 0s and 1s so they are **not** all interchangeable. The requirement here is to look carefully at the datasheets for the parts used or use the parts selected in the base design.

While the chips are out of the sockets, it is worthwhile to power up the board and use a voltmeter or multimeter to check the voltage on each chip on the appropriate pins (74 series will be 7-14 or 8-16 while large chip pins vary). It is generally ok for the voltage to be a 0.01 of a volt lower on one side of the board than the power plug side.

Locate pull-up array resistors (sip and dip types) and check that the associated pins on logic chips are at the correct voltage. Put the voltmeter on all chip reset pins and confirm that these pins transition from voltage to zero when the reset button is pressed.

We have now just about exhausted the static testing that one might consider, so try the board again for operability. If success remains elusive, then dynamic testing is needed.

Dynamic hardware testing is going to require tools such as an oscilloscope and/or a logic analyzer to weed out a failed chip or missing signal. I bought two UART oscillators (1.8 Mhz) so that if I ran into trouble I could effectively half the processor clock frequency in the event that dynamic testing was necessary. Slower really is easier. Some legacy chips run slower than modern versions.

I might also suggest that checking in with the user group would make sense at this point. Both hardware suggestions and methods makes sense, but additionally someone might be aware of a conflict between a boot Eprom version and some other element of the board or system.