

BFD-68 Boot Delay after Power On

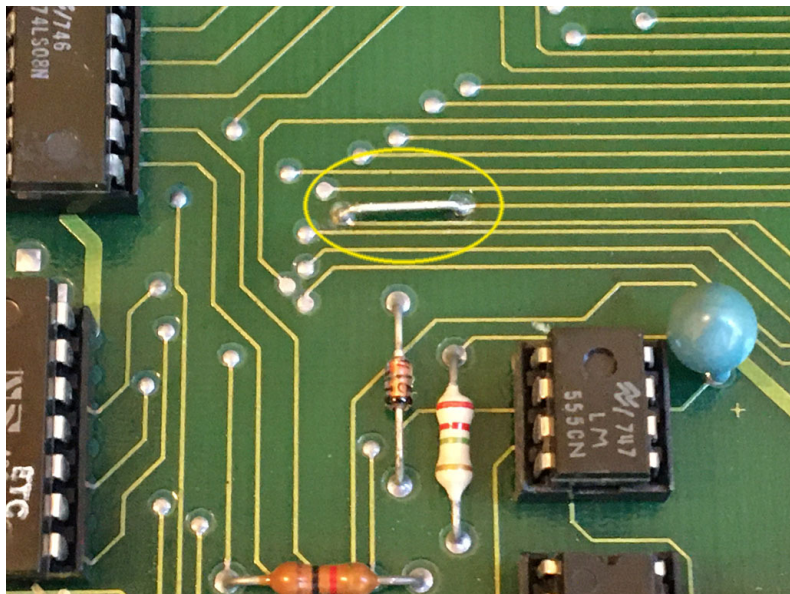
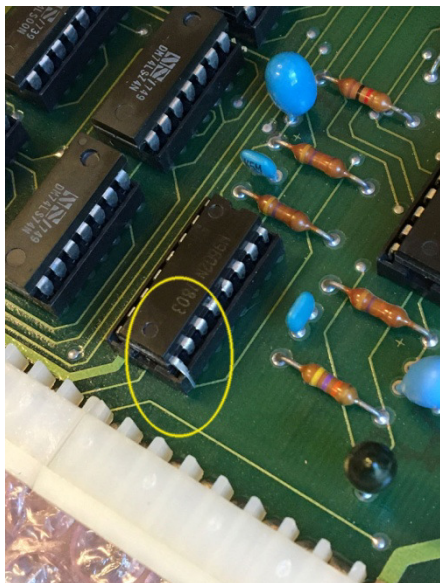
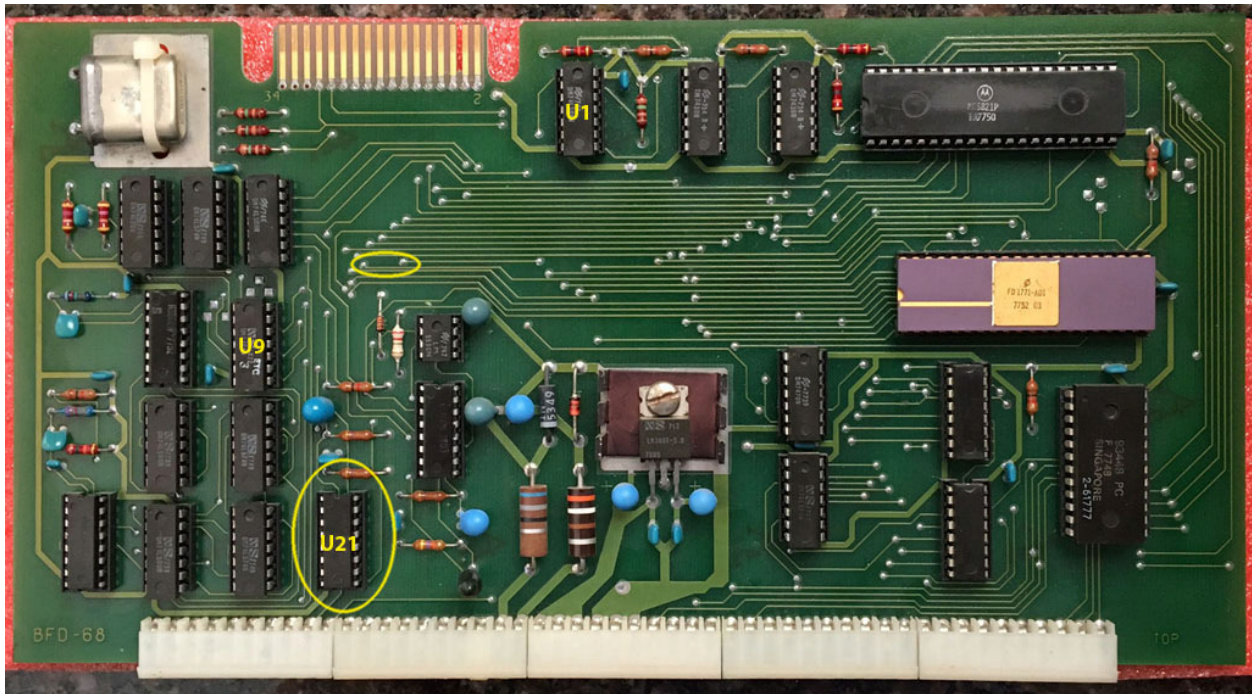
When using the BFD-68 controller, there is a 10 second delay after jumping to \$8020 (to boot the disk) until any disk activity takes place. The 68A version of the board does not have this issue. This paper describes why this problem occurs.

- 1) On power up, due to the start up timing of the PIA and the 1771, the 1771 powers up thinking it has an error and asserts INTRQ. The power on reset of the 1771 (generated by a one-shot on the board) does **not** clear INTRQ.
- 2) The 1771 reset also writes a RESTORE command (seek to track zero) to the command register. However, the command will not execute until INTRQ is cleared by the host computer. This requires the host to read the status register or write another command.
- 3) At this point – just after power on, the SWTPC 6800 is at the SWTBUG prompt waiting for the operator to type J 8020 to boot the disk. The 1771 has INTRQ set and a RESTORE command is pending execution. When then the operator subsequently issues a jump to \$8020 to boot the computer, the boot ROM reads the 1771 status register which clears the INTRQ condition in the 1771. This, in turn, causes the 1771 to then execute the RESTORE command already in the command register. Note that the RESTORE command initiated by reset does not load the head, so the drive select signal – which is AND'd with the head load signal – is not asserted to the drive. This, in turn, means the drive's Track 0 output will not be seen by the 1771, so the 1771 ends up issuing 256 step pulses before aborting the command. Doing the math, you'll find that $256 * 40\text{ms}$ per step is the 10 second delay that occurs.
- 4) After the boot code reads the status register as described above, which, in turn, releases the reset RESTORE command to execute, the boot code writes its own RESTORE command to the 1771 and then waits in a loop for it to complete. This new RESTORE command does not affect the reset RESTORE that is already running, so when the boot code thinks its own RESTORE command has finished, it is actually the reset RESTORE that just finished. However, since the drive was not selected during the RESTORE, the drive did not actually step to track zero. This, in turn, means the seek and read the ROM then performs to read the boot sector will most likely fail. This failure causes the boot loop to repeat, which then finally executes properly and the system boots.

Work Around

The ten second delay can be eliminated with a minor board mod that makes the BFD-68 work the same as the "A" version of the controller in regard to this problem. See the two areas circled in yellow below.

- 1) Lift pin 9 of U21 out of its socket
- 2) Install a jumper between the pads circled in yellow closer to the top of the board



This mod connects the head load output (HLD) of the 1771 (pin 28) to U1 pin 1, replacing the FKTR00 signal previously connected to U1 pin 1. FKTR00 comes from U21 pin 9 which is disconnected by lifting the pin out of its socket. The HLD signal is picked up at a feedthru going to U9 pin 1.