The Art of Hardware Handshaking

There's more than one data-halting method in town.

You, too, can shake hands with your Model 100—figuratively, of course. Handshaking is a term used to describe certain protocols used to control a computer's telecommunications session. An example of handshaking is the popular XON/XOFF (*Ctrl-S*, *Ctrl-Q*) method of temporarily halting data transmission.

The XON/XOFF protocol isn't the only data-halting handshaking method in town. There's an even simpler one, known as DSR/DTR. Unlike XON/XOFF, which is usually a provision provided by the communications software, DSR/DTR (which stands for data set ready/data terminal ready) is implemented in the serial RS-232-standard hardware. XON/XOFF is used mainly for communicating between computers; DSR/DTR is even more widely used between computers and peripherals such as serial printers, modems and pen plotters.

XON/XOFF is a well-publicized Model 100 family communications feature; the fifth character of the telecommunications STAT command is either *E*, for XON/XOFF enable, or *D* to disable the protocol. (Sorry, DSR/DTR isn't mentioned in the Model 100 owner's manual.)

Provision for DSR/DTR hardware handshaking is wired into the Model 100—but the machine's internal programing doesn't utilize it. I considered modifying the system ROM to implement that function, but when I examined the Model 100's schematic, I found a very simple and inexpensive hardware solution that doesn't affect the laptop's normal operation.

Let's take a closer look at DSR/DTR. In the standard DB-25 connector used for RS-232 serial communications purposes, pin six is labeled DSR, and pin 20 is labeled DTR. Depending upon whether the device hooked into the RS-232 is a "host" (DCE, or data communication equipment or a "terminal" (DTE, or data terminal equipment), the pin is either set to a positive voltage to indicate the device is ready to accept data or to a negative voltage to request a halt in the data stream.

This protocol offers an improvement over XON/XOFF, which is merely implemented through the ASCII code; using DSR/DTR, each device merely monitors a single voltage level. In XON/XOFF, each device must be constantly decoding the ASCII stream, and must process XON and XOFF characters instantly. In a slow or time-share system, the XON/XOFF codes might be temporarily waiting in a buffer; with DSR/DTR, the communications hardware

By Thomas Neidermeyer

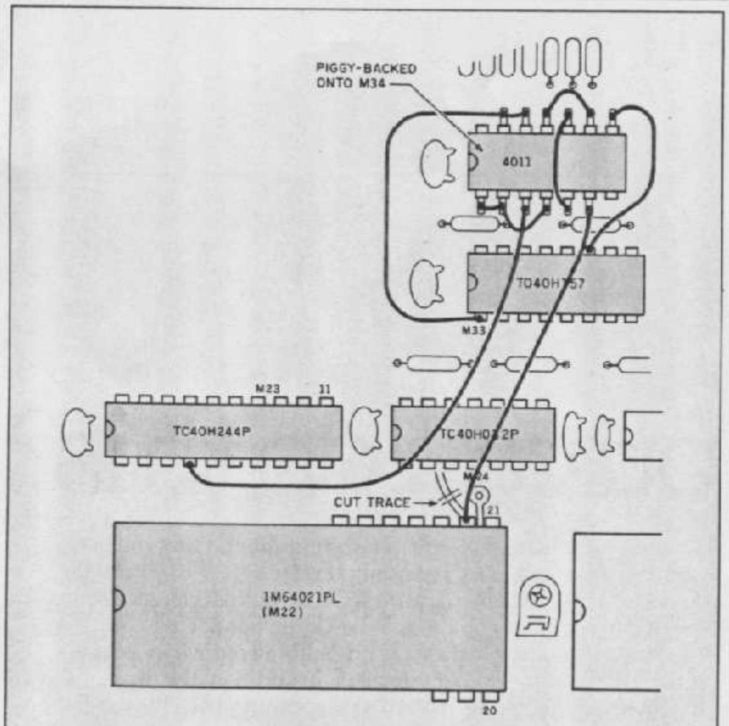

Figure 1. The 4011 logic chip is soldered piggy-back onto chip 4584 at position M34.

notices the signal instantly.

Our modification of the Model 100 is simple, requiring only a 19¢ CMOS integrated circuit. This 4011 logic chip is soldered piggy-back onto another chip, numbered 4584, at position M34 on the Model 100's main circuit board (Fig. 1). We'll also need to cut one trace and run a few short wires. The entire project should take no longer than a half-hour. Experience with soldering and wiring is recommended. Please be careful not to overheat the chips or damage them with static electricity:

1. Carefully bend all the 4011's pins (Fig. 2), except for seven and 14, so that they stick straight out from the body.
2. Open the computer's case by removing the four bottom screws and carefully prying the two sections apart.
3. Locate the 4584 chip at position M34. Place the 4011 on top of it, making sure it has the same orientation as the 4584 (the notches are on the same side). If necessary, bend seven and 14 of the 4011 so they touch the corresponding pins on 4584.
4. Use a small-tipped soldering iron to solder the 4011's pins seven and 14 to the 4584's pins seven and 14. This will provide power to the 4011. Be careful not to overheat the chips.
5. With a fine-bladed knife, such as an X-acto No. 11, cut the trace at chip M22's pin 22 on the main circuit board's component side. Chip M22 is a 40-pin IM6402 UART (universal asynchronous receiver/transmitter) between the 80C85 processor and the 81C55. Make sure you leave a gap at the point where you cut the wire.
6. Solder a short piece of thin wire, such as wire-wrap, from M22's pin 22 to the 4011's pin six.
7. Solder M23's pin four to the 4011's pin three.
8. Solder M33's pin 11 to the 4011's pin eight.
9. Solder M33's pin one to the 4011's pins 12 and 13.
10. Solder the 4011's pin one and two to pin four.
11. Solder the 4011's pin five to pin ten.
12. Solder the 4011's pin nine to pin 11. All 4011 pins should now have a connection.
13. Inspect your work and close up the computer.

Now, it's time to test your DSR/DTR modification. Since the Model 100's serial port is configured as a DTR, you'll need a null-modem cable that crosses pins six and 20, as well as two and three, to talk to most serial printers or other computers using this protocol.

The TBRE signal, generated by the UART, controls parallel data on the data bus being fed to the UART via parallel port M23. The 4011, which is a quad NAND gate, takes the DSR signal, which as been inverted and changed to CMOS voltage levels, and interrupts the TBRE signal when the DSR line is negative. Because the same UART also feeds serial data to the internal modem, this process is overridden when the modem is active.□

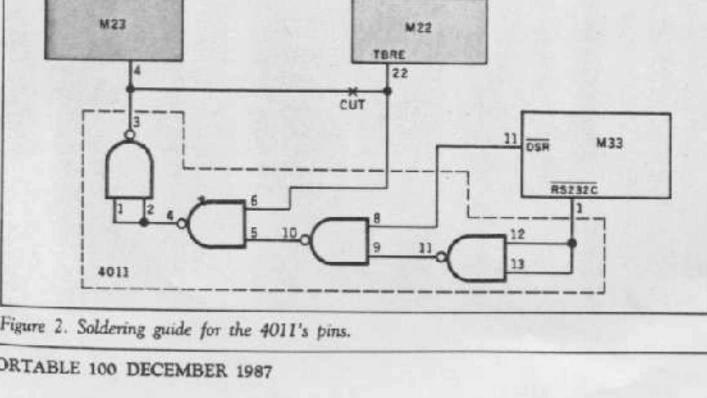

Figure 2. Soldering guide for the 4011's pins.

A

http://club100.org

Scan

By: Don Fox   sysop@atomicfrog.com

"It's full of stars....”.